Invited Speakers

Ambient Mobility: Human Environment Interface and Interaction Challenges

Prof. José Luis Encarnação
INI-GraphicsNet Stiftung, Germany

ABSTRACT
Through the convergence of mobility, ubiquity and multimediality/multimodality a new information & communication technology paradigm is emerging: Ambient Intelligence (AmI). AmI basically means that computers move from the desktop into the infrastructure of our everyday life to build networks of smart items serving "smart players" (humans, machines, smart items, animals, etc.) in intelligent environments. AmI begins to influence the way we interact with our environments. An important aspect of future human interaction therefore is the way this interaction evolves from human-computer interaction (HCI) to a human environment interaction (HEI) supporting us in efficiently managing our personal environment, not only at home or in the office, but also in public and industrial environments.
This contribution addresses the fundamental components that are involved in the forthcoming human-computer-environment interaction. Focus will be specially on the challenges arising when the interaction takes place with mobile services e.g. providing information, supporting work tasks, or enriching leisure, in public and possibly outdoor environments.

AVI'08, 28-30 May , 2008, Napoli, Italy

Computational Photography and Video: Interacting and Creating with Videos and Images

Prof. Irfan Essa
Georgia Institute of Technology, USA

ABSTRACT

Digital image capture, processing, and sharing has become pervasive in our society. This has had significant impact on how we create novel scenes, how we share our experiences, and how we interact with images and videos. In this talk, I will present an overview of series of ongoing efforts in the analysis of images and videos for rendering novel scenes. First I will discuss (in brief) our work on Video Textures, where repeating information is extracted to generate extended sequences of videos. I will then describe some our extensions to this approach that allows for controlled generation of animations of video sprites. We have developed various learning and optimization techniques that allow for video-based animations of photo-realistic characters. Using these sets of approaches as a foundation, then I will show how new images and videos can be generated. I will show examples of Photorealistic and Non-photorealistic Renderings of Scenes (Videos and Images) and how these methods support the media reuse culture, so common these days with user generated content. Time permitting, I will also share some of our efforts on video annotation and how we have taken some of these new concepts of video analysis to undergraduate classrooms.

AVI'08, 28-30 May , 2008, Napoli, Italy

Principles of entertainment in inhabited television

Marco Fanciulli
FOX Channels Italy
marco.fanciulli@fox.com

ABSTRACT
Inhabited TV paradigms have been around since a while and several experimental implementations have been delivered around the world. The basic model of these experiments involves the deployment of collaborative virtual environments so that users can take part in TV shows from within these virtual, shared environments. Unfortunately, this approach although cheap and easily implementable, doesn't add up too much to engagement, pace gap between real/virtual world, camera control techniques and most important, adequate TV Formats.
The talk presents a new paradigm of basic principles for entertaining a virtually deployed audience allowing for itneraction and maintaining the entertainment sensation.

Categories and Subject Descriptors
H.1.2 [Models and Principles]: User/Machine Systems—Human factors; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented and virtual realities; H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces—Theory and models; I.3.7 [Computer Graphics]: Three-Dimesional Graphics and Realism—Virtual reality; J.5 [Computer Applications]: Arts and Humanities—Arts, fine and performing

Keywords: Inhabited TV, Virtual life, Social design

1.INTRODUCTION

Since a decade, researchers and TV authors collaborate to find new compelling ways to entertain people and overcome the unidirectional intrinsic nature of linear contents. Most of these collaborations had their foundations in existing technologies developed for different uses and bended to serve these new purposes. Leveraging on broadband connections, reducing latency for continuous streaming multimedia packets, and empowering video coding capabilities were the base for introducing effective two way communication, interactivity and realistic representation of synthetic objects within the TV environment and TV contents within virtual spaces.
Only in few cases researches aimed at developing deeper, innovative integrations between the fiction of linear TV contents and the real-time, undetermined nature of participation.

AVI'08, 28-30 May , 2008, Napoli, Italy

To the latter, belong experiments such as NOWninety6, The Mirror and Heaven & Hell-Live, all demonstrating that are technically possible to let a multitude of participants to take part as active subjects into large television or para-television events. Not only and more importantly, had those led to a classification of roles and constraints [1] that is still valuable and substantially correct.

Unfortunately, major limitations to these experiments came from the lack of compelling and attractive content formats, as recognized by Benford [1], and this is a significative issue, indeed. The saying “content is king” is not really correct. We believe that there’s much more and more subtle and important missing to this equation, making the quest for the perfectly credible inhabited TV show impossible if not discovered and explained. Psychological and perceptual subtle principles underlying every successful show, either real or virtual, have to find their way into the scenario to create breakthrough contents. Great content is king.

Differently from researches that moved from the TV and movie industry to infer general principles, this paper starts from massively multiplayer online games experiences to try to understand the basics of social interactions and the role these have on perception of freedom in virtual worlds, mixed virtual and real worlds and participation to strictly ruled, narrative universes..

2.THREE PROBLEMS WITH UNRESTRICTED CVEs

1.1. Self-awareness

Before any other aspect of projecting a human perception into a virtual environment where all cognitive functions are set to be exposed not only to external objects but to an external representation of *se*, it’s fundamental that the environment we’re designing allows for self-consciousness of the viewer. A top down approach to his has been given by Sommerhoff in his book “Life, Brain and Consciousness”. His work’s conceptual basis is the identification of a viewer internal representation which he understands as the primary higher cognitive functions of consciousness. In this paper these representations will be used to better define and qualify the ability to be, to perceive and to interact with and within a virtual world. Hence, along the entire paper the concept of consciousness will follow Sommerhoff’s systemic description of consciousness by requiring the following two catego-

ries of higher cognitive functions being stimulated by the virtual TV environment:

1. First order self-awareness, which is a comprehensive and coherent internal representation of the world and the self-in-the-world. Events are representations of this kind.
2. Second order self-awareness, which are representations of the occurrence of a representation of the first order as being part of the current state of the self.

In this definition of self-awareness, there's no space for realism or other qualitative and directly sensible qualities of the world. The ability to set ourselves or a representation of ourselves into a virtual or real world is just a capability to form internal representation of first and second order categories. Thus the quest for obtaining belief and trust in the world reconstruction operated within a virtual TV show, is not a function of the reconstruction itself but should be based on some more profound and intangible aspect of perception.

1.2. Behaviour of the others

Of all the efforts done by researchers and industry player to bring efficient, realistic and enjoyable virtual spaces to final users none compares to those aimed at moderating and driving behaviours of participants. Recently, an expert in MMORPG games didn't hesitate in describing as "frequently barbaric" his own experiences in real-time, shared, inhabited virtual worlds. Insults of all kinds, including racist and homophobic slurs, are commonplace every time users have a chance to hide themselves behind anonymous virtual identities.

As the women reading this paper can easily testimony, there's no need to do something special but classifying themselves as female to be subject to unsolicited, frequent and mostly negative attention.

It's not only about verbal and written communication, too. In virtual worlds human beings tend to reproduce and possibly intensify their worst aspects: they cheat, rob, beat other players or simply don't deliver the tasks their specific roles demand. Just imagine how to deal with people leaving a TV show before their role is ended.

1.3. (Virtual) Social dismantle

Behaviour of participant in social aggregations, either stable or occasional, directly has impacts on behaviours of the mass of participants as a whole. Just like in real life, but with amplified results, in absence of clear and accepted rules a society transforms universal behaviours in non-written rules or, seen from the other side of the wall, the mass conforms to the behaviours of the natural leaders.

A social aggregation being driven or disturbed by aggressive and insulting leaders, will grow in tension, disharmony and unpleasant evolution.

Large audience virtual environments perform and are crossed by set of activities, single and social behaviours quite different from those encountered in the cited experiments.

Rules applied in that environment are quite related to basic capabilities given to participants and visualization oriented ones. Those are *bona fide* systems, based on the fact that the experiment environment is access controlled, limited in size and partially in-synch with the real-time storytelling broadcast.

1.4. How can we deal with these problems?

Social aggregations, their culture and recurring behaviours can be designed to match the goals of the environment they are set to be staged in. Like in organized sports well designed rules enable fun game play and well designed videogames allow for immersion and belief, good social design creates a socially acceptable and enjoyable environment which deeply impacts on social behaviours of its inhabitants.

Real life observation and social psychology provide several direct proofs of the role of perceived social environment (aka situation) on behaviour of individuals. That's the basic reason why people don't sing at the cinema nor go to the local church to drink a whiskey and chat with friends.

Environmental design can vehicle and alter individual behaviours to the extent of conforming it to the needs of wider interests (namely the storyline in TV shows). As per Darley & Batson experiment in 1973 [2], the situation drives different course of action by actors disrespectfully from the gravity of what interfere with their game goals.

1. SOCIAL DESIGN IN REAL AND VIRTUAL LIFE: A MATTER OF SOCIAL RULES AND GUIDANCE

Psychological researches are interesting, but they have no value if they don't influence social behaviour in real life (or in its projection into virtual worlds), changing the rules of what is considered to be socially acceptable.

This continuous interference of social design and manipulations happens so often in our everyday life, we can't recognize anymore how subject to conformism and new inputs we are. Apart from canonical samples of commercials giving social messages about what's in and what's out for a trendy living, there are subtle and more powerful samples around.

When cell phone hit the market, users simply transferred their pavlovian behaviour (phone rings = answer it) to the new device. It didn't matter where they were; answering was the right action to perform for that stimulus.

It took quasi a decade for users to be educated not to answer in theatres, cinemas or cult temples. More, it took time to convince people to silence their phones and to leave if absolutely requested to answer. The new habit was induced by pushing messages inviting to switch off phones before a movie starts, presenting everyone with the idea about how

FIGURE 1
BATTLE FOR RANKING: TEAMMATES FIGHT ONE EACHOTHER

irritating could be listening to other's phone calls while trying to focus on something different. It all insisted on the social aspects of situations incompatible with answering a personal call on a cell-phone.

The belief that "jerks will be jerks" is neither true nor useful in the effort of building socially acceptable environments and gaining a wider audience. It isn't a responsibility for us any more than it was for the movie industry -- but the economic incentives are the same.

As we said previously, the game industry already addressed these issues. They did exactly by designing social environments that could enhance the perception of the virtual place as a socially organized and just world, permitting inhabitants to focus on the entertainment contents. Friends lists and party systems are all examples of useful social design.

Unfortunately these features have a big limitation in having your friends online to participate. They are not general enough to permit entertaining and socially stimulating participation of anonymous actors or newbie (still socially alone).

But those features are not enough; they are really only valuable if you already have friends online. If the multiplayer games are going to be welcoming to new players, we need social features that affect newbie who may not (yet) have friends.

More, very often in MMORPG users don't behave as expected, especially if they are called to team-work against a common enemy or toward a common goal. Intra-team competitive behaviours arise quite often and most of the players redirect their effort at being better positioned than their team mates. This behaviour is favoured also by disparities in underlying technologies: users with faster network connections or faster PCs have a clear advantage over other users, maybe coming from mobile connections. Social equality passes also through technological mechanisms to equate and remove these gaps. This is quite a different task than solving problems such as synchronization of media assets across massive multi-presence networks; it has to do with justice and perception of equity the same manner people calling a TV game from a cell phone trust they have the same chance to be first to call as viewers calling from fixed line phones.

Next generation inhabited shows have to create as first instance satisfying social environments for non-hardcore viewers to reap the rewards of a larger and more loyal customer base. They also don't need complex CVE environments, as belief and self-projection in the show don't depend at all on surround imagery or 360 degrees environments. It's all about building the experience and managing the complexity of user intervention in storytelling.

Much of the enviable success of WoW is attributable to Blizzard's ability to create a social environment that is friendly to newbies while catering to the hardcore. The Sims is another example of this winning approach. They've shown it is possible to do both. Massive amounts of machinime shows built from individuals organized in teams, shows how this model can be effective in realizing the dream of really participative television.

2. BORING DIGITAL ENVIRONMENTS VS PARTICIPATION AND EMOTION

Any inhabited or other form of participative TV platform strives for Lost-style and 24-style shows to be set in an open-way for viewers. The research driven iTV scene often privileges new participation mechanics or subjects for brand new show formats aiming at becoming a paradigm for future experiments or commercial implementations (not differently from authoring a TV show to be a season hit), but quite always end up into low visual quality, poorly written stories with unstable pillars. That's because it's neither about how things are presented nor entirely about how things are narrated: it's all about how things are perceived by social aggregations.

3. THE FEEL OF PARTICIPATIVE TV

What is the "feel" of a participative TV show? Every gamer knows what's the feeling of a videogame and can easily recall the sensation, the sunesthetic feeling of controlling some virtual avatar or agent. It's what causes you to lean left and right as you play while controlling your virtual projection inside the game space. You don't need a dome projection facility nor virtual glasses: it's all about giving the player the illusion of controlling the story and setting the next course of action of the entire virtual universe.

It's the feeling of controlling some entity outside your body, making it an extension of your will and instinct. This "virtual sensation" is in many ways the essence of videogames and virtual television show spaces, one of the most compelling, captivating, and interesting emergent properties of human-computer interaction.

For this sensation to arise and be sustained along the entire duration of the iTV experience, many things have to happen both technically (in the platform) and in the viewer's mind. Enhancing our understanding of this mechanism is the only way to push our capability to create virtual inhabited shows way beyond our current technical-only possibilities.

3.1. Aesthetic standards

As told in the previous paragraph, technical aspects can be really marginal at providing the virtual sensation of being part of a story. This is not something new to the animation industry as it has been coded decades ago by pioneer animation team at Disney Pictures. Their codification is known as "Principles of Animation" and is nothing more than a set of non negotiable aesthetic rules; aesthetic because they deal with the way things have to be painted to look more real (independently from the painting technique or the photo-realism of the strokes), non negotiable because they simply code the basic quantities our brain subconsciously measure to set what's believable and what's not believable. That's how we can suspend our reality frame to believe an elephant like Dumbo can fly. It's not about consciously recognizing the principles rather than feeling them [3] [4].

Despite the apparently different set of rules governing massive virtual environments and participation of multitude of users to basically non-written stories, inhabited television and participative shows set on virtual stages are governed by similar principles. Identifying these principles allows for a clearer understanding of virtual sensation and how to create it by designing social environments and trustable rules.

Either we're going on a bicycle, driving a car of flying an airplane we have a strong feeling of being in control, of mastering the machine as a direct extension of our body, reacting to our will. This is probably one of the things differentiating human beings from the vast majority of animals. So much that some of us search for new, extraordinary ways of understanding how to control objects: rafting, climbing, riding, doing base jumping.

Many people find this pleasure in video games, too. These are virtual places where virtual feelings are both distilled to their essence and free of the constraints and dangers of more physical activities. All visuals, audio and physical sensations (such as those provided by active joysticks) contribute to this sunaesthetic perception of the virtual environment as a real(istic) one.

This is the "feel" or "sensation" of the virtual stage and despite technical advances in time, its essence has been the same since the creation of oscilloscope table tennis.

Descriptions of this "feeling" lead quite often to physical attributes, mapped onto real life sensations usually brought to superlative levels. These are plainly aesthetic judgments, judgments that indicate some kind of inviolable rules are in play just like in traditional cartoon animation but with a significant difference: while in traditional animation stories flow linearly from beginning to end, in participative environments and games the virtual feeling is primarily set by user inputs. No input, no action.

The following four principles of virtual sensation are a first attempt at creating useful guidelines for developing interactive, inhabited shows that "feel" right; or at the very least, to avoid common mistakes that interfere with the viewer's experience of the show.

4. FOUR PRINCIPLES OF SENSORIAL ENGAGEMENT

The four principles of virtual sensation or sensorial engagement defined here will hopefully pave the way to virtual shows designers – or indeed, anyone concerned with human-computer interaction – to enhance HCI. They are a conscious attempt to improve the users' unconscious experience by stimulating unexpressed, socially modifiable basic events and attributes:

1. Context - setting the physical rules and spatial context the stage is set into.

2. Good & predictable feedback – Enabling mastery, control, and learning by rewarding player experimentation.

3. Novelty – There are an infinite number of results from the same input.

4. Anticipation & Timing– Defining the weight and size of objects through their interaction with each other and the environment.

These principles aim at being universally applicable to any participative TV scenario or virtual environment.

4.1. Context

Most of the current platform providing some sort of virtual space for users to live in, expose no real matching between spacing (the distance between objects in the virtual stage) and the capabilities users are provided with. In Linden Lab's Second Life there's plenty of space, a low density of people and a mismatch between how fast an avatar can walk or fly and distances between buildings. Flying was thought to be a

convenient way to cover the vast amount of space lying between buildings but it's not something we experience everyday and in order for a user to suspend disbelief, the way we fly or jump should be in context with the average distance between objects and places.

Giving a user incredible powers is totally meaningless if these powers are not in scale with the surrounding world.

Context helps the user to feel the virtual engagement needed to suspend disbelief by giving a meaning to the motion in virtual space. These results as the interaction of three different elements: spacing, perception of environment and representation.

As we said, spacing deals with distribution of objects in space. In a gaming show where users have to pass obstacles, the frequency they encounter these have significant impacts on the virtual sensation: rare objects pose no challenges nor reference points to the viewer while dense ones can overwhelm the user to frustration (because they feel not to be in control of the situation).

Perception has further impacts on the balance between boring/overwhelming ratios. It's similar to the concept of anticipation in traditional animation in that it relates to the ability to see things in advance. In a racing game, the ability to see an obstacle on the road as soon as it rises from the horizon let the user react appropriately and timely. On the contrary, if objects just appear when they are a few milliseconds from player he will be frustrated because he can't react in a useful way.

Perception is also related to feeling the speed and size of things. In virtual environments much more than in real ones, the two concepts are somehow interchangeable. Speed is perceived as a relative function of how things surrounding us are moving away. The size of these objects also determines how fast this motion (not our real speed) is perceived. Big objects remain in our visual field longer and diminish our perception of speed while small objects quickly leave our visible space and enhance the sensation of speed. The same way, low speed tends to alter the perception of how big an object is. So speed directly relates to perception and spacing.

The last element affecting the context is representation. Massive objects need to behave like we expect in real life and so should any object in the scene. Big objects have to provide a sensation of their size and (maybe) weight, small and soft objects the contrary. In a game named Shadow of the Colossus, the colossi move at a very slow pace and make the camera shake at every step, rising dust and projecting small stones all around. Everything in their representation lead the viewer in being convinced they are huge, heavy and extremely dangerous beasts. If they were more agile or quick, this sensation of realism wouldn't happen. This is because the relative scale of objects in a game creates expectations in the player about how these objects should behave.

Most of the virtual environments available today, independently from how good is the graphic engine, are places with no physics or compelling representation. Every life form in the virtual space moves and behaves as if they had no weight, no relative size: in one word, no personality.

From the most massive boulder to the tiniest kernel of dust, if it's going to move it needs to move appropriately

Shadow of the Colossus

4.2. **Good & predictable feedback**

If users are called to participate to an event, either a live TV show or a pre-ordered cinematic adventure, they need to feel as if everything is under control a reacts in a predictable, consistent way.

This is the very core of virtual engagement and immersion. If moving your head to left sometimes would rotate the camera to right and sometimes to left randomly, you would not be able to understand the world surrounding you nor have the feeling of mastering it. No control, no engagement because our lives are all about understanding and controlling.

In the excitement of development, too many developers try to map non intuitive actions to improbable reactions, missing the key point of predictability. Sometimes they try to heuristically determine every course of action in the misbelieve that the underlying engine will be able to fulfil every request in a consistent way. This complicates the user controls without adding more to his capability to interact with the world.

Predictable control is all about mapping basic inputs from the user to consistent, expected movements and reactions in the virtual space. This is not only a question of believable and consistent mapping, but also of how these are implemented. A mouse moving the arrow on screen too fast, is non predictable as we perceive the movement of our hands as continuous and perfectly linear while the higher frequency sampling of the mouse is perfectly able to subdivide this movement into smallest, ever changing events. This way we loss the mapping between the expected results and the actual ones.

Behaviour of controls can change during the experience, nut in a homogeneous way. If we walk on a solid ground, we can move easily and with minimum effort. If we're transported

onto a valley of mood or on a beach, the same inputs become slower or even non effective. We changed the effectiveness of our actions but the mapping between the basic function (walking) is preserved.

Natural mapping of controls is the key concept at allowing users to effectively be part of the universe we are calling them to participate. Clear and natural understanding of rules and direct experience of their consistency is the psychological result any control interface should aim to.

Predictability is a powerful tool for show designers as it let users to infer from the very beginning of the show a clear picture of the show itself. Every rule learnt during this quick tutorial approach will be valid for the entire persistence in the virtual space. If I put my finger in a power plug I'll be shocked by an electric current. If I survive, the rule will be consistent forever; I don't need to check every power plug to confirm the theory.

This is a very important design aspect. In quite all virtual environments, most of user time is spent failing at doing something. Especially in the first few minutes of participation, a virtual run is pure experimentation of new motion mechanisms, own relative size in the virtual world, discovering what's dangerous or forbidden. If we fail at giving them a quick path for learning and consistent, stable results for their actions we'll loose them at the first scene change.

Finally, for feedback to be useful, it needs to accurately communicate the state of the show to the viewer. If you did something wrong and have been eliminated you need to know why.

One of the most appealing things about virtual environments and social aggregations is the sense of measurable progress of the projected life, which is often formalized into points or level ups or some other numeric metric of skill progression. In real life TV shows, only games are all about these metrics but still feel like something pertaining only to the game mechanism, with no sensation engagement in the viewer. On the contrary, virtual shows have potential to introduce a welcome change from traditional TV general shows, where there are very few formal metrics of progress.

4.3. **Novelty: infinity from singularity**

Talking about novelty after an entire paragraph spent at putting predictability and consistence at the very core of the emotional engagement, could seem to be a bit at odd.

This can be true in traditional animation and videogames where recurring actions by each character are reproduced every time is needed. Once the graphic artist defined the animation for a walk cycle, the character will adopt always the same walk cycle. This is evident in linear animation and most in videogames (especially old platforms, shoot'em ups and driving games).

In real life, driving a car in a circuit implies the car to pass several times over the same point and asperities but quite never the car will behave the same way. Each movement of the car over a bump on the road will be different depending on a myriad of factors such as quantity of fuel, position of the pilot, speed, asset, conditions of the pneumatics). Thus a bit of novelty (potentially leading to loose of control of the car) enter the stage at every lap. For a virtual sensation in a virtual show to hold the viewer's interest, it needs to feel novel and interesting even after dozen of episodes. Even repetitive actions should feel fresh each time you trigger them.

Traditionally games and virtual environments such as Second Life try to address this need for novelty by entering additional contents in the scene, changing difficulty level or playable objects in order to convey attention of the user to different things at every repetition.

The most promising approach adopts a deterministic global physics system, which keeps a virtual sensation feeling by processing inputs from the user far beyond their capability to sense them. Technically speaking the same input provides always the same, deterministic result so the user will be safe with a predictable model. Novelty is conveyed by the fact that the user will quite rarely be able to provide the same, exact input given the sampling frequency is far finer than he can consciously perceive. It's more sensitive than the viewer's perception, much like the real world; we expect certain reactions to take place when we interact with objects (or objects interact one each other) and we expect that no reaction will be exactly the same two consecutive times. This is the nature of the world surrounding us: messy and imprecise.

Because our perception is keenly tuned to physical reality, we subconsciously expect certain things to happen when objects interact and move. One thing we expect is that no motion will ever be exactly the same twice. This is the nature of reality: messy and imprecise.

4.4. **Anticipation & Timing: define weight and size of objects thought their interaction with other objects and the surrounding environment**

As we described in the previous paragraphs, a well balanced mix of the principles of virtual sensation makes up for a powerful sense of mass, size and weight in the mind of the viewer. It doesn't need for deep documentation to master the environment: a few minutes spent in the virtual space provide a clear understanding of its governing rules, physical laws and behavioural mapping of everything from imaginary beasts to properties of common life things such as light and sound.

And just like in real life, the viewer doesn't need to experiment every single possibility to infer general, constant rules. It's all about subconscious learning of the global network of rules determining how objects interact and react to our inputs. It let the user understand what is socially expected from him and what is perceived as unfair or even "criminal" according to a local justice system.

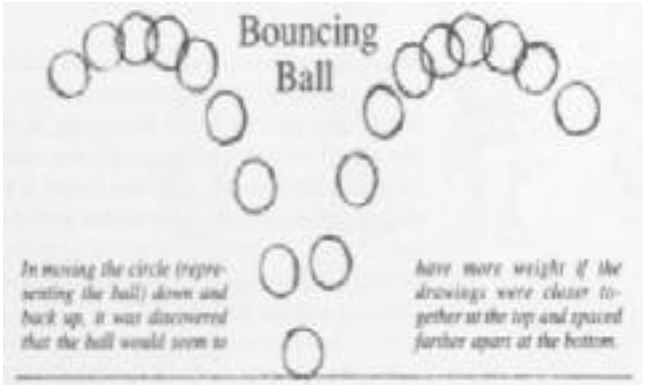
Representation is the principal weapon the authors have to make this strong participation sensation look realistic and persist in the consciousness of the viewer by supporting the intersection of subconscious details making him overcome initial disbelief. If representation is inadequate or erroneous (think of games with 3D objects intersecting other objects), the user will have a strong point of reference to reality able to disrupt the virtual sensation ("this is a badly programmed simulation"). The narrative flow is broken and the suspension of disbelief is gone.

Since much of our knowledge about the way physical reality works comes from watching objects interact, any attempt to reproduce in high fidelity the real world is doomed to fail both for the technology not being adequate yet and because immersion has nothing to do with photorealism. This is what makes satisfying resolution of virtual environment representation difficult to achieve: humans are sharply and subconsciously tuned to the way things are supposed to work at a very cognitive level. Eyes are just sensors collecting inputs but the way those are processed only partially depends on resolution and fidelity.

One way to avoid this issue is to simplify your representation. If a character looks photorealistic, it is perfectly reasonable for a player to expect that their interactions with objects in their environment will perfectly mimic reality. If a character is stylized or simplified, it will not defy the player's expectations if their interactions are also simplified.

To use object interaction to effectively convey information to the player about the relative weights and masses of objects and the nature of their interactions, remember one thing: you're faking it. The goal is only to create the perception of weight, mass, and force in the player's mind. This is different from the way things "really are" according to physics. You don't need to simulate exact physic models but only fake the sensory system of viewers to convey the sensation of a correct physical behaviour of objects.

The way to effectively fake object interactions is by looking at how people perceive things. From the principles of traditional animation, it's a well-known phenomenon that exaggeration can make it more convincing to the audience. Squashing and stretching an object in ways that when viewed as individual frames seem bizarre and unnatural makes them read much better when animated and can be used in virtual representation to add up sensation of weight, resistance, softness or hardness. It all depends on relative amount of exaggeration with respect to the surrounding objects:



As long as something happens when objects interact, and that something seems to be appropriate for the speed, mass, and weight of the objects, the feeling of impact is conveyed.

Here comes the bound with traditional filming and TV show production as the camera is a powerful actor on stage to further emphasize what the principles subtle induce. Shaking the camera, emulating lens flares or saturation help adding much physical effects to visible and invisible inputs.

4.5. **Conclusion**

The underlying goal of all the principles discussed above is to create a feeling of control and mastery so powerful that it transcends context and platform and becomes a powerful tool for self expression. This feeling creates a strong sense of ownership, which is what happens when viewers can express themselves in a meaningful way through a show or game in virtual spaces.

The goal of any participative TV show is to provide entertaining, life-enriching flow and social experiences, experiences that don't exist watching a film or reading a book. Compelling virtual sensation is a great foundation for these experiences, providing feelings of challenge, mastery, and control as well as a beautiful kinaesthetic experience unique to any medium.

Throughout the paper I referenced "inhabited TV shows", "participative TV" and "iTV" models as if those were interchangeable terms. This is an attempt to simplify and reduce *ad unicum* different streams of research wherever overlapping concept can be applied. Csikszentmihalyi's original work on flow, Beyond Boredom and Anxiety. For more information about how flow applies directly to games, reference Sweetser and Wyeth's Gameflow: A model for evaluating player enjoyment in games.

REFERENCES

[1] S.Benford, "Inhabited Television: Broadcasting Interaction from within Collaborative Virtual Environments", ACM Transactions on Computer-Human Interaction, Vol. 7, No. 4, December 2000, Pages 510 –547.

[2] J.M. Darley - C.D. Batson, "From Jerusalem to Jericho:A study of Situational and Dispositional Variables in Helping Behaviour", http://faculty.babson.edu/krollag/org_site/soc_psych/darley_samarit.html

[3] Frank Thomas & Ollie Johnston, "Illusion of life", p. 47-69

[4] Online reference for Principles of Animation: http://www.evl.uic.edu/ralph/508S99/contents.html

[5] BENFORD, S., GREENHALGH, C., AND LLOYD, D. 1997a. Crowded

[6] collaborative virtual environments. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97, Atlanta, GA, Mar. 22–27), S. Pemberton, Ed. ACM Press, New York, NY, 59 – 66.

[7] BENFORD, S. D., GREENHALGH, C. M., SNOWDON, D. N., AND BULLOCK, A. N. 1997b. Staging a public poetry performance in a collaborative virtual environment. In Proceedings of the 5th on European Conference on Computer-Supported Cooperative Work (ECSCW '97, Lancaster, UK, Sept.). Kluwer B.V., Deventer, The Netherlands.

[8] CAPIN, T. K., PANDZIC, I. S., NOSER, H., MAGNENAT THALMANN,

[9] N., AND THALMANN, D. 1997. Virtual human representation and communication in VLNet. IEEE Comput. Graph. Appl. 17, 2, 42–53. DAMER, B. 1997. Demonstration and guided tours of virtual worlds on the Internet. In Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '97, Atlanta, GA, Mar. 22–27), S. Pemberton, Ed. ACM Press, New York, NY, 10 –11.

[10] DRUCKER, S. M. AND ZELTZER, D. 1994. Intelligent camera control in a virtual environment. In Proceedings on Graphics Interface. 190 – 199.

[11] GREENHALGH, C. M., BENFORD, S. D., TAYLOR, I. M., BOWERS, J.M., WALKER, G., AND WYVER, J. 1999. Creating a live broadcast from a virtual environment. In Computer Graphics Proceedings, Annual Conference Series on SIGGRAPH '99 (Los Angeles, CA, Aug. 8 –13). 375–394.

[12] S. SWINK, "Principles of virtual sensation", Gamasutra, http://www.gamasutra.com/view/feature/1781/principles_of_virtual_sensation.php

[13] HE, L.-W., COHEN, M. F., AND SALESIN, D. H. 1996. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In Proceedings of the 23rd Annual Conference on Computer Graphics (SIGGRAPH '96, New Orleans, LA, Aug. 4 –9), J. Fujii, Chair. Annual conference series. ACM Press, New York, NY, 217–224.

[14] THE LONDON TIMES. 1998. TV from another planet: Something virtually different. The London Times (Oct. 7). Interface Section.

[15] W, G. 1997. The Mirror—Reflections on inhabited TV. Br. Tele. Eng. 16, 1, 29 –38.

[16] G. Sommerhoff, "Life, Brain and consciousness. New perspectives through targeted systems analysis", in Advances in Psychology, 63, North-Holland, 1990

Interaction Techniques

Flower Menus: A New Type of Marking Menu with Large Menu Breadth, Within groups and Efficient Expert Mode Memorization

[1,2]Gilles Bailly [2]Eric Lecolinet [1]Laurence Nigay
[1] LIG, University of Grenoble 1, Grenoble, France
[2] TELECOM ParisTech, LTCI CNRS, Paris, France

{Gilles.Bailly, Laurence.Nigay}@imag.fr {Gilles.Bailly, Eric.Lecolinet}@enst.fr

ABSTRACT
This paper presents Flower menu, a new type of Marking menu that does not only support straight, but also curved gestures for any of the 8 usual orientations. Flower menus make it possible to put many commands at each menu level and thus to create as large a hierarchy as needed for common applications. Indeed our informal analysis of menu breadth in popular applications shows that a quarter of them have more than 16 items. Flower menus can easily contain 20 items and even more (theoretical maximum of 56 items). Flower menus also support *within groups* as well as hierarchical groups. They can thus favor breadth organization (*within groups*) or depth organization (hierarchical groups): as a result, the designers can lay out items in a very flexible way in order to reveal meaningful item groupings. We also investigate the learning performance of the expert mode of Flower menus. A user experiment is presented that compares linear menus (baseline condition), Flower menus and Polygon menus, a variant of Marking menus that supports a breadth of 16 items. Our experiment shows that Flower menus are more efficient than both Polygon and Linear menus for memorizing command activation in expert mode.

Categories and Subject Descriptors
H5.2. [**User Interfaces**]: Interaction styles. I.3.6. [**Methodology and Techniques**]: Interaction techniques.

General Terms
Design, Human Factors,

Keywords
Marking menus, Polygon menus, Flower menus, *within groups*, curved gestures, novice mode, expert mode, learning performance.

1. INTRODUCTION
Marking menus [9] are a combination of pop-up radial menus and gesture recognition. Marking menus thus define an interesting alternate solution to Linear menus. However, Marking menus are not yet widely introduced in graphical interfaces. One possible reason is their limit to support an important number of commands: it has been shown that with reasonable accuracy, the limit of hierarchical Marking menus is 64 items (breadth-8, depth-2) [10]. Several variants of Marking menus [1][19][20] have been proposed to partially overcome this limitation: while Multi-Stroke menus [19] focus on the menu depth, Polygon menus [20] increase the menu breadth.

AVI'08, 28-30 May , 2008, Napoli, Italy

In this paper, we introduce Flower menu, a new type of hierarchical Marking menu, that is designed to contain an important number of commands (>1000). To do so, Flower menus (Figure 1) increase the menu breadth of Marking menus by supporting 7 different curved gestures for each 8 directions. They can then theoretically contain 56 commands at each level. In practice, Flower menus can easily support about twenty commands for a given level (for instance 17 commands in Figure 2-d), which is sufficient for many menu applications: indeed our informal analysis of menu breadth in some popular applications shows that the average number of items per level is 12.4, almost half of the considered applications contained at least 14 items and a quarter of them more than16 items.

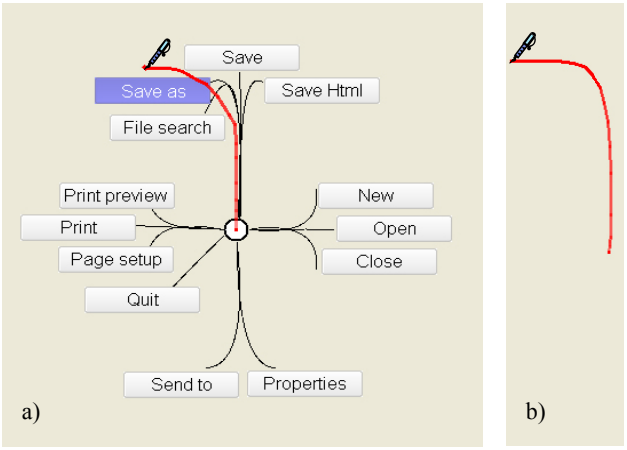
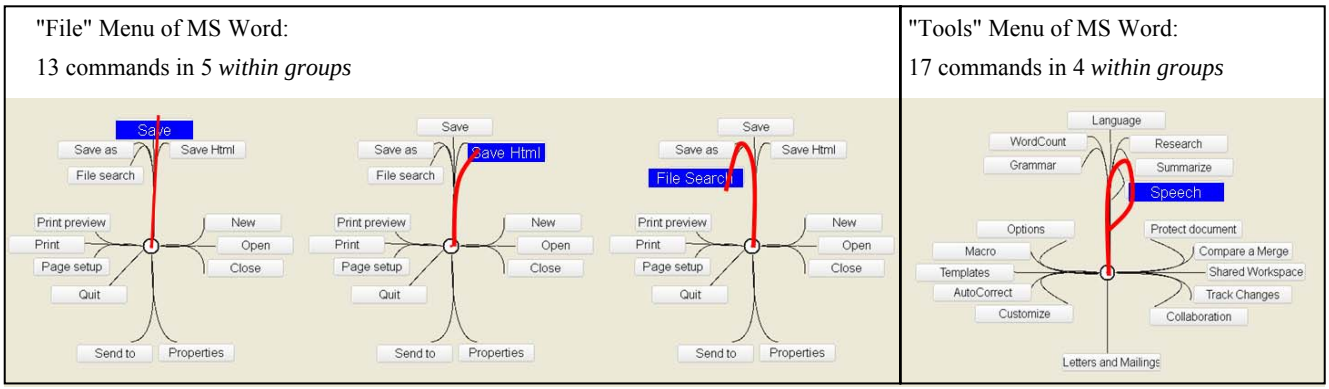


Figure 1. Flower Menus (a) Novice Mode, (b) Expert Mode.

In addition to increasing the menu breath, another key feature of Flower menus relies on their ability to support *within groups*. Two types of item groupings are commonly used in menu techniques [13]: *within groups* and hierarchical groups. *Within groups* correspond to item groups at a given level (breadth organization). They are common in Linear menus: such groups are separated by a line (for instance, "New" and "Open" are in the same *within groups* in the "File" menu)". Hierarchical groups



(a) Straight (b) Bent (c) Cusped (d) Pigtail

Figure 2. Samples of curved gestures in Flower Menus: the Straight (a), Bent (b), Cusped (c) gestures in the "File" menu and the Pigtail gesture (d) in the "Tools" menu of Microsoft Word

correspond to item groups across a menu and therefore define the depth organization of a menu. While both *within groups* and hierarchical groups are commonly used in Linear menus, it is surprising to observe that previous studies have never considered *within groups* in Marking menus. Flower menus support *within groups* as well as hierarchical groups. They can thus favor breadth organization (*within groups*) or depth organization (hierarchical groups). In this paper, we focus on 1-level Flower menu: we therefore do not consider hierarchical Flower menus. For a given level, not only Flower menus can support a large number of items, but these items can also be organized in a variety of ways in order to reveal meaningful item groups (i.e., *within groups*). The resulting flexibility in the design of Flower menus is illustrated in Figure 2. The “within group” feature is new in Marking menus and we believe that it makes the menus and items easier to remember and to learn.

The learning of the expert mode is a key point of Marking menus. As users execute the same gesture in novice and expert modes, Marking menus offer a "fluid transition" from the novice to the expert mode: Users learn the expert mode implicitly, just by using the menu repeatedly in novice mode. In contrast, hotkeys (i.e. keyboard accelerators) need to be explicitly learnt by the novice users in Linear menus, and this can slow down the learning process. There are few available experimental studies that compare the learning performance of Linear menus and variants of Marking menus [12]. In this paper, we experimentally investigate the learning performance of expert mode of Flower menus and Linear menus. In our experiment, we also consider Polygon menus since they are one the very few variant of Marking menus that supports more than 8 or 12 items at the same level. Our experimental study shows that Flower menus are more efficient than both Polygon and Linear menus for memorizing command activation in expert mode.

The paper is organized as follows: we first discuss related work. We then present the design of the Flower menus. We finally describe a formal experiment and its results that compare the learning performance of the expert mode of Flower menus with that of Linear and Polygon menus.

2. RELATED WORK

As explained in the introduction, Marking menus [9] were introduced by Kurtenbach in an attempt to facilitate the transition from the novice to the expert mode. The novice mode is triggered when the user presses down the pointing device and waits approximately 1/3 of a second. The menu then appears centered around the position of the cursor, allowing item selection by moving in the direction of the desired selection. If the user does not wait and begins dragging immediately, the menu enters into expert mode where the cursor leaves an ink trail. When the user releases the mouse, the gesture recognizer determines the selected item. As novice and expert modes use similar gestures, users should learn the expert mode implicitly, just by performing enough selections in novice mode. Another important feature of Marking menus is that they make possible "eyes free selection" thanks to the scale invariance of interpretation of marks.

The radial layout of Marking menus limits the number of items that can be selected. Performance tends to degrade as menu size increases and 12 items seem to be the maximum to ensure an acceptable error rate [11]. Hierarchical Marking menus have thus been proposed [12] to increase the total possible number of items. Commands can be selected by compound or "zigzag" marks. But this number remains limited: only breadth-8 menus with a depth of at most 2 levels can maintain a reasonable accuracy rate of more than 90%.

Multi-Stroke marking menus [19] define an alternate design that improves the expert mode of hierarchical Marking menus. This technique uses temporal instead of spatial composition: a series of simple inflection-free marks must be drawn instead of a single compound mark. However, while effective in expert mode, this design tends to decrease performance in novice mode. This problem was solved by Wave menus [1], a variant of Multi-Stroke menus that provides optimal performance in both modes.

As explained in [20], the breadth limitation of the different kinds of hierarchical Marking menus may imply awkward groupings of items as well as an increased menu depth. In expert mode, deeper menus require more complex gestures that need more time to be drawn, and are more likely to be badly recognized for traditional hierarchical Marking menus as shown in [12]. In novice mode, the user needs to navigate in a larger number of submenus that may cause disorientation [16]. For these reasons, Zone and Polygon menus [20] have been introduced as a way to extend the menu breadth up to 16 items. These two variants of Marking menus consider both the relative position and orientation of elementary strokes. In the first case, the user first taps to specify the menu

origin. This action virtually splits the screen into 4 spatial areas (up/down x left/right relatively to the tap location). Each area corresponds to a different breadth-4 marking menu that the user activates in the usual way. Polygon menus work in a similar way except that the items are the vertices of a N-sided polygon as shown in Figure 2-b. A noticeable consequence is that Polygon menus require "tangential" instead of radial gestures (relatively to the menu origin). Moreover the direction of gestures matters and triggers different commands. Hence, while Zone menus can be seen as a kind of hierarchical radial menu, Polygon menus indeed follow quite a different design. Both techniques were reported to have good performance for selecting items, although slightly slower than Multi-Stroke marking menus. But this was globally compensated, considering the fact that regular breadth-8 Multi-Stroke marking menus would require an increased depth for providing the same number of items.

A common point of all these studies is that they only evaluated the performance for selecting items in expert mode. While Marking menu techniques and their variants seem likely to favor the transition from the novice to the expert mode, we found only one study that attempted to verify this hypothesis experimentally [12]. The experiment focuses on the behavior of two users of an extended real application over a long period of time (i.e., hundreds of hours). In this setting, results demonstrate the effectiveness of Marking menus over Linear menus and show the gradual transition from the novice mode to the expert mode. Nevertheless the study was performed with two users only. Moreover alternate design of Marking menus such as Polygon menus are different enough from the original Marking menus to lead to significantly different results. This point motivated our experimental study on learning performance of the expert mode. Before describing the conducted experiment and its results, we first present the Flower menus, that we have considered in our experiment.

3. FLOWER MENU DESIGN

Flower menus[1] extends Marking menus by making it possible to draw straight or curved gestures. As with standard Marking menus, the user must press the mouse, perform a radial gesture and release the mouse. The user always starts a gesture from the same point (i.e. the menu center in novice mode) and no tap is needed to specify the menu origin as is the case for Polygon menus. This property is important as users reported that they prefer gestures "starting from the center" in Flower menus rather than "having to perform two operations" in Polygon menus.

Althought used in a different context, curved gestures have been proposed in menuing systems [6] or for entering text [8]: closed loops in the first case and bent gestures on the on-axis in the second case.

In addition to orientation, curvature provides a complementary way to encode input data. Flower menus make the most of possibilities to increase the number of available commands while making them easy to perform. In order to fullfill this criterion, we retained 4 different degrees of curvature. Considering the rotating direction (clockwise, counterclockwise), Flower menus provide 7 gestures for each 8 directions (Figure 3 shows them for the North orientation):

- S: a *straight* gesture, as in regular Marking menus,
- B-,B+: *bent* gestures, that can either be curved in the clockwise or counterclockwise direction, as in the "hybrid design",
- C-,C+: *cusped* gestures, that can also be curved in both rotational directions,
- P-,P+: *pigtail* gestures, considered in both rotational directions.

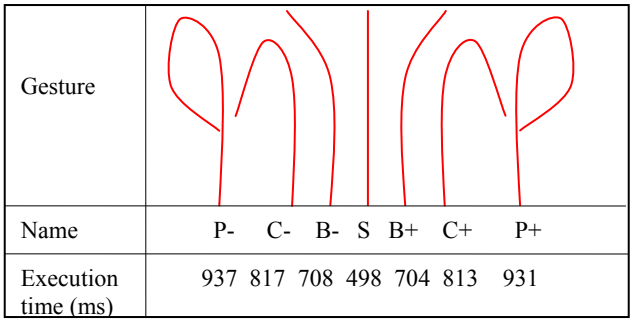


Figure 3. The 7 gestures of Flower menus for the Northern orientation and their average execution times.

A Flower menu can thus contain a theoretical maximum of 7 * 8 = 56 items for each level (practically, Flower menus can contain approximately 20 items). While most menus will obviously not contain so many items, this feature is most useful for creating *within groups*, enabling the designer to choose amongst a large variety of spatial organisations. For example, Figure 2 shows how to organize the 5 *within groups* of the Microsoft "File" menu in a Flower menu. Moreover since Flower menus support both *within groups* and hierachical groups, the designer has even more possibilities for spatial arrangments, balancing breadth organization (*within groups*) and depth organization (hierarchical groups). Meaningful groups make easier the learning and memorization of commands. Indeed commands of a particular group are semantically related and such a semantic relationship is stored in the human declarative memory. This is possible because of the spatial arrangement of commands in a Flower menu group. Commands are spatially close in a "petal": such proximity and closure are two Gestalt principles.

About memorization, it is also worth noticing that Flower menus are based on a highly symetrical design. They use 4 different types of gestures (*Straight, Bent, Cusp, Pigtail*), that can be drawn along 8 different orientations and curved in 2 different ways (except for *Straight* lines, where the curvature is null). These 4 gesture types can also be seen as a variation of the same drawing: a line that is more and more curved. As a consequence, users can consider and remember the 56 theoretical possible positions of a Flower menu as a combination of 3 variables having at most 8 possible values (i.e. 8 orientations x 4 types x 2 rotating directions). This point may be an important factor for memorization, as explained in the discussion section.

Finally, hierarchical Flower menus work in the same way as Multi-Stroke Menus [19]. Both menus support a series of overlapping marks (Figure 4) rather than the kind of single zigzag marks used in original Hierarchical Marking menus (HMM). Mutli-Stroke menus have been shown to be as fast and less error prone than HMMs [19], especially for large menu systems as HMMs tend to produce many errors for diagonal gestures.

[1] A video can be found at: www.gillesbailly.fr

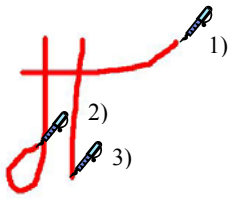


Figure 4. A selection with a 3-level Flower menu (that generalizes Hierarchical Multi-Stroke Menu) in expert mode. 1) bent, 2) pigtail and 3) straight marks.

4. PILOT STUDY

We conducted a pilot experiment to study how users perform Flower gestures. The experiment is fully described in [2]. The expected outcome of our experiment was: a) to obtain experimental data in order to develop an effective gestures recognizer; b) to verify that users could draw all these gestures precisely enough c) to find the most efficient gestures for the design of a flower menu by identifying where frequently used items should be prefrentially placed in the menu. Besides, the gesture database that was produced during this experiment was then used to train and to test the recognition algorithm. The 14 right-handed participants were asked to draw as quickly and accurately as possible 56 gestures (8 directions * 7 gestures). To illustrate the conducted experiment, Figure 5 shows all the performed "Bent" gestures drawn by all the participants for the counterclockwise direction.

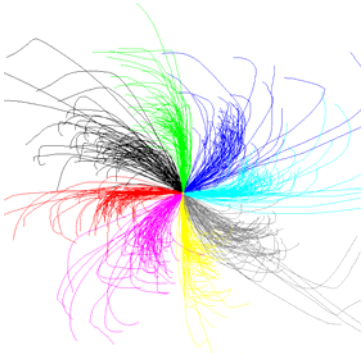


Figure 5. Bent gestures for the counterclockwise

As expected by the two-thirds power law [18], our results show that drawing time grows with curvature: straight lines (498 ms) are faster than bent (704 ms), cusp (813 ms) and pigtail (929 ms). The most frequent commands should thus preferentially be placed on straight and bent lines. These results are coherent with those of [3][20] for comparable gestures. However, it is interesting to notice that the times obtained in all these studies is higher than in [3] whose experiment favors speed because it is based on very repetitive movements. The actual speed obtained by very trained users of Flower menus may thus be shorter than in our results.

We did not consider inflexions (corner gestures) in our experiment for two reasons: this would have made it too long and inflexions have been shown to be slower than bent gestures [3]. For this reason, the 16 first commands of Flower menus (straight and bent gestures) should be faster than the 16 commands in a 2-level Marking menu as these ones require inflexion gestures (besides, 2-level Marking menus do not provide equivalent capabilities as all items can not be seen at the same time).

We also observed that the angular variability is higher on the off-axis orientation (diagonals). As a consequence, large groups should be preferentially put on the on-axis orientation of the Flower menus. However, this effect can be largely compensated by an effective recognizer taking into account the actual size and position of the angular sectors of circular menus.

So instead of considering a naïve algorithm that would not take precisely into account how users draw marks and would thus misinterpret some correct gestures, we developed a specific recognizer (based on K-nearest neighbors) which is both fast and effective. We used the samples drawn by one half of the participants for training and the other half for testing. We also removed gestures that were erroneously drawn from the database (about 2% of all gestures).

The recognizer is fast enough to provide immediate feedback. The overall recognition rate is 99% for the first 24 commands (straight + bent gestures); 96.5% for the first 40 commands (cusped gestures added) and 93% for all the commands. However, for the case of real applications, pigtail gestures corresponding to the case of *within groups* of 6-7 items will not be very frequent. The real recognition rate will thus certainly be superior to 96.5%. The tuning of the gesture recognizer was a prerequisite for the experiment presented in the following section. The samples of the testing set were merged with those of the learning set to obtain a larger learning database.

5. EXPERIMENT

The goal of this experiment was to compare the learning performance of the expert mode of Flower menus with Linear menus (baseline) and Polygon menus. In this experiment, we focus on the comparison of three significantly different menu techniques that can contain at least 16 commands at the first level as it is often the case in existing applications. We did not consider traditional 2-level Marking menus nor 2-level Multi-Stroke menus in this experiment for three reasons: a) these techniques do not allow to display many items at a single level; b) this would have introduced another variable (the menu depth) in the experiment; and c) this would have made our experiment too long. However, comparing the performance of 1- and 2-level Flower menus would be an interesting track for future work as Flower menus both generalize Multi-Stroke menus and provide more items.

5.1 Menu Configuration

We designed a “canonical” menu configuration (Figure 6) that is intended to be representative of those seen in real applications. For this purpose, we performed an informal analysis of the content of pull down menus (more precisely, the first-level pull down menu in menu bar) in popular applications for MS-Windows (Table 1).

According to this informal analysis, the average breadth is 12.4 items, 46% of the menus contain at least 14 items and 23% of them more than 16 items. As explained in [20], these results confirm the need for increasing marking menu breadth. They led us to perform our experiments with breadth-16 menus, breadth-16 being also the maximum size for Polygon menus. Furthermore, in our informal analysis, we studied the frequency of *within groups* depending on their size (Table 1). All menus have *within groups*, 58% of these groups contain 1 or 2 items and 92% of them up to less than 4 items. This led us to adopt a menu configuration with similar statistics (Figure 6): Two 1-item groups; two 2-item groups; two 3-item groups; and one 4-item group. In this design, the percentage of 1-2-3- and 4 item groups is close to the results of our analysis. Besides, this configuration (shown in Figure 6) is

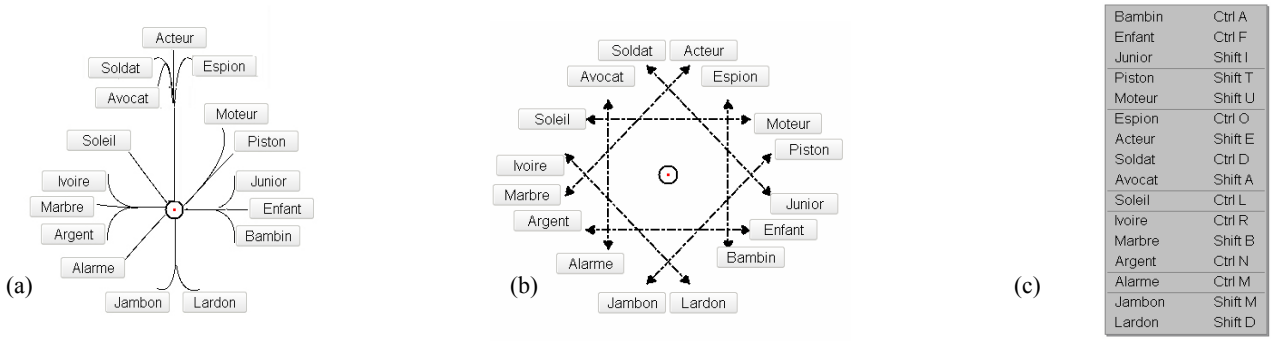


Figure 6. (a) Flower, (b) Polygon, and (c) Linear menu configurations used in the experiments.

almost identical to the File menu in word 2003 for Windows. According to the results of our pilot study (angular variability being larger on the diagonals), we placed the largest groups on the on-axis orientations of the Flower menu (Figure 6-a). We also placed same sized groups in different areas of the menu to avoid layout singularities. Groups were placed in the same order in the Polygon menu (Figure 6-b) and the Linear menu (Figure 6-c), starting from the NW position of the Flower menu and by following the counterclockwise direction.

Application	nb items	menu ≥ 3 groups (%)	groups ≤ 2 items (%)	groups ≤ 4 items (%)	groups ≤ 7 items (%)
excel 03	13.3	89	37	92	100
adobe reader 7.0	10.6	86	68	94	100
word 03	14.2	89	45	85	100
firefox 2.0	8.9	100	72	92	100
thunderbird 0.9	9.4	100	61	94	100
photoshop 7.0	18	100	66	94	99
mean	12.4	94	58.17	91.83	99.83

Table 1: Informal analysis of pull-down menus in some applications for MS-Windows.

5.2 Items and groups

The design of Flower menu makes groupings implicitly visible. We slightly changed the positions of items in the Polygon menu in order to reveal groups (so that items belonging to a same within group would be slightly closer). We used regular separators in the Linear menu. Each group contains items corresponding to a given category such as colors, animals, music, transportation means, etc. These categories and the item names were carefully chosen to avoid possible ambiguities (so that an item could not belong to multiple categories). All item names are 6 letters long and do not contain rare French letters such as Q, Z, Y, W, H.

5.3 Linear menu hotkeys

Keyboard hotkeys were assigned to items in a way that attempted to be as realistic as possible while avoiding undesirable singularities that could bias the results. For this purpose, we realized an informal analysis of hotkeys in Microsoft Word and FireFox. This study showed that there is a great variety of hotkeys and that the hotkey letters is not always part of the item name. For example, Ctrl+D activates the "Font" command in Word and Ctrl+F2 the "Print Preview" commands in Firefox. However, we decided to make the task simpler for our participants because many of them complained in a preliminary experiment where hotkeys were not always contained in the corresponding item name. Hotkey letters are thus part of the name in our experiment with the exception of the first and last letter to avoid making certain items easier to remember.

We also discarded C, V, X, and Z because some users developed specific strategies to remember the mapping between items and hotkeys with these specific letters. This effect, probably caused by the high familiarity of users with these keys, would have introduced undesirable variability. We only used Ctrl and Shift as modifier key although other modifiers and combinations of them are common in real applications (especially on the Macintosh where commands such as Shift+CMD+DEL, Alt+CMD+M, Alt+Shift+CMD+C,… are widely available). A consequence of this design is that we do not only use keys located on the left side of the keyboard (as done in some previous studies [5]) because they are not enough of them to match 16 items without breaking the previous constraints.

As a conclusion, Linear menus were tested in rather favorable case in our experiment. A real life application that would attempt to associate as many possible hotkeys to commands: a) could not use the first letter or even simply a letter of the word for most commands because of name collisions; b) could not use well known hotkeys because they are already used for standard operations; and c) would thus have to use all possible letters, symbols and function keys and a variety of modifier key combinations.

5.4 Stimulus

The stimulus was the name of the item that the user had to select. We used a textual stimulus, rather than an iconic one, in order to avoid possible confusion since the items are grouped according to semantic relationships.

We did not use a Zipfian distribution [5] but a uniform target frequency. This is because the memorization of items may depend on ordering (for Linear menus), on orientation (for the two marking menus), and on type (for Flower menus). A Zipfian distribution would thus make results dependent on where the most frequent items are placed in the 3 types of menus. A uniform distribution avoids this problem and makes results comparable with the 3 menu techniques.

5.5 Hypothesis

H1: Markings menus (i.e. Flower and Polygon) favor expert mode memorization because the same actions are performed in novice and expert mode.

H2: Expert mode memorization is better with Flower than Polygon menus. Flower menus with explicit *within groups* make the mapping between gestures and orientations very straightforward, a feature that may help memorization.

H3: Linear menus are faster than Flower menus that are faster than Polygon menus in expert mode. Linear menus should

outperform marking menus on this criterion because hotkey activation should require less time than drawing a gesture [17]. The average performance of Flower menu gestures should be higher than Polygon gestures for a well-balanced 16 item menu (that is to say a menu where items are not arbitrarily put on the slowest locations).

5.6 Procedure

In this controlled experiment, we intend to evaluate the learning of expert mode, by comparing how many items the users are able to select in expert mode. More precisely, the purpose of our study was to evaluate the *intentional* learning of the expert mode as opposed to *implicit* learning since users were explicitly asked to learn the expert mode. Nevertheless a design that makes it easier to remember the expert mode also favors its implicit learning.

We chose to evaluate *intentional* rather than *implicit* learning because this latter condition is, by nature, imprecisely defined. It is in fact quite difficult to evaluate implicit learning in a controlled experiment because these conditions are likely to influence how the users learn the expert mode. A longitudinal user experiment within the context of a real-world application as described in [12] would be necessary for studying implicit learning. Indeed, based on our previous observations and those from previous studies [5], users adopt different strategies for learning the expert mode, especially in the context of a real task.

Our experiment roughly follows the design of the *memory recall task* in [4] and comprises three different phases.

Familiarization. The familiarization process consists of explaining how the tested technique works in novice and expert mode and allows for user practice in order to be sure s/he knows how to operate. This phase took about 2 mn.

Training. Participants where instructed that the goal was to learn how to select as many items as possible in expert mode. They were told not to "rush" in selecting items because time was unimportant in this phase and excessive speed would degrade their performance in the testing phase. They were then asked to select items during 5 mn, first in novice mode to learn them, then in expert node when they felt able to do so. The same item was presented again in case of a wrong selection. Otherwise, the stimulus was chosen according to a random distribution (except that an item could only appear once in a 16 stimuli sequence).

Testing. Participants were asked to correctly select items in expert mode as fast as possible, the novice mode being disabled. The stimulus was the same and the 16 possible items were presented in random order. This phase was repeated twice in order to get more experimental data in order to evaluate the time performance. During this phase, no feedback was provided to indicate if the selection was correct. Nevertheless we gave participants a second chance to learn the menu: between two blocks, the menu was displayed again for 15 seconds.

5.7 Design

The ordering of the three techniques was counterbalanced across subjects using a Latin square design. Three equivalent sets of item names were used to avoid transfer effects between the first, second and third tested technique. As these three sets were chosen to be semantically equivalent, this should not have a noticeable effect. However, we also counterbalanced sets with techniques and orderings so that all the techniques would be tested with the same conditions. Each participant performed the experiment in one session which was about 40 mn long. In summary, the design was as follows:

18 participants **x**
3 menu techniques **x**
16 gestures **x**
2 blocks
= 1728 selections.

5.8 Participants and Apparatus

18 participants (3 female) ranging in age from 22 to 35 years (mean 26) were recruited from within the university community and received a handful of candies for their participation. They were all right-handed and familiar with computers. The experiment was conducted on a Dell Latitude D800. The experimental software was implemented in C++/Qt. Participants used a 3 button Logitech mouse. A mouse was used, rather than a tablet's stylus, for two reasons: the mouse is still by far the most commonly used input device and previous studies showed that equivalent or better results are obtained by using a stylus [10]. By performing our experiment in the "worst case" we wanted to demonstrate the robustness of Flower menus as well as their efficient usage with common input devices.

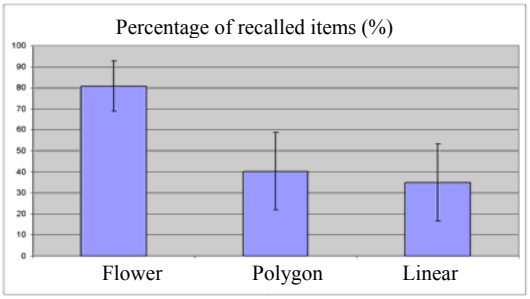


Figure 7. Percentage of recalled items for the 3 menu techniques.

5.9 Results

As expected, a 4-way analysis of variance shows that the item sets have no significant effect on memorization or selection time.

5.9.1 Expert mode memorization

Analysis of variance reveals a significant main effect for techniques on the **number of recalled items** ($F_{2,34} = 70.34$, $p < 0.0001$). A post hoc Tukey test with 5% alpha level shows (Figure 7) that Flower menus, with 81% of recalled items (12.9/16), are better than Polygon menus (40%; 6.4/16) which are better than Linear menus (35%; 5.5/16). Hypotheses H1 and H2 are thus verified, but we expected a smaller difference between Polygon and Flower menus (as they are both marking menus) and a larger difference between Polygon and Linear menus.

Analysis of variance also shows an effect for testing order ($F_{2,34} = 5.69$, $p< 0.01$). The number of recalled items is globally higher for techniques tested in second rank (9.5/16) than in first (7.3) and third (8.0) ranks, but there is no [technique x order] interaction.

5.9.2 Activation performance

The time required to activate commands comprises two components: the reaction time (interval between the appearance of the stimulus and the mouse down) and the execution time

(drawing time). ANOVA indicates a significant effect for technique on **execution time** ($F_{2,34} = 21.58$, $p < 0.0001$). A post hoc Tukey test with 5% alpha level shows that Linear Menus (0.6 seconds) are faster than Flower Menus (0.8 s) that are faster than Polygon menus (1.7 s).

While the results for the execution time correspond to our hypothesis (H3), this is not the case for the reaction time. ANOVA shows that the **reaction time** ($F_{2,34} = 9.07$, $p<0.001$) is significantly longer for Linear Menus (2.9 s) than for Polygon Menus (2.1 s), and is longer for Polygon Menus than Flower Menus (1.6 s). These results suggest that the mapping between commands and hotkeys were less well learned than between commands and gestures (Flower gestures being especially efficient).

ANOVA reveals a significant effect for technique on total time ($F_{2,34} = 7.34$, $p < 0.01$) that indicates that Flower Menus (2.4 s) are faster than Linear Menus (3.5 s) and Polygon menus (3.8 s). Hypothesis H3 is thus not completely verified as Linear menus are slower than Flower menus.

Finally, ANOVA also indicates a significant effect for block on reaction time ($F_{1,17} = 3.76$, $p< 0.001$) and total time ($F_{1,17} = 14.14$, $p < 0.001$). Block 2 is faster than block 1 both for reaction (2.0 vs. 2.4 s) and total time (3.0 vs. 3.4 s).

5.9.3 Subjective preference

In a post-experiment questionnaire, participants ranked the three menu techniques as follows: Flower, Linear and Polygon (in preference decreasing order). 17/18 subjects chose Flower Menus as their favorite technique. We also asked their opinions about the following criteria: familiarization, simplicity, learning, speed, accuracy and fun according to a 5 pt Likert scale. Flower obtained the highest value for all criteria except accuracy. ANOVA followed by a pairwise comparison reveals that: Flower and Linear menus are significantly faster than Polygon menus for familiarization (F:4.6; P:3; L:3.9), speed (F:4.3; P:3.1; L:3.9) and simplicity (F:4.5; P:3.1; L:4.4). Logically, Linear menus (4.8) are significantly more accurate than Polygon (3.9) and Flower (3.7) menus. The "recall" criterion was significantly higher for Flower menus (4.4) than for Polygon (2.2) and Linear menus (2.3). Finally, Flower menus (4.7) are more fun than Polygon menus (3.3) that are considered more "fun" than Linear menus (2.1).

Finally, most users said they preferred gestures "starting from the center" with Flower menus rather than "having to perform two operations" (hence referring to the initial tap of Polygon menus). Most of them found it easy to learn and perform Flower gestures. One user summarized up a general feeling as follows: "I make the general orientation, then I adjust". Some users found it difficult to "learn two things" in Polygon menus, "the position of the item and the gesture". Others noticed that they "knew the position of the command but could not recall the gesture" in Polygon menus.

6. Discussion

Activation performance. Our results on Total time indicate that Flower menus (2.4 s) are faster than Linear menus (3.5 s) and Polygon menus (3.8 s). The difference between Flower and Linear menus performance is caused by a much longer reaction time in the case of Linear menus. This point suggests that hotkeys were well less learned than gestures in our experiment. However, it is interesting to remark that the reaction time is overestimated and the execution time is underestimated for Linear menus. This is because, the amount of time needed for moving the hands to press the hotkeys should theoretically be counted in the execution time, but this was not technically feasible in our experiment.

Another important remark is that our experiment was not conceived to evaluate activation performance but user capability to learn the expert mode of these menus. The activation times we obtained give interesting indications for comparing the relative performance of these three kinds of menus but they should not be interpreted as the actual times that would be obtained for trained users in expert mode. Both reaction and execution time would be shorter. For instance, execution times are about 20% faster in our pilot study where the task was closer to expert usage.

Memorization performance. Our study clearly shows that Flower menus are more effective than both Polygon and Linear menus for memorizing command activation in expert mode: Flower menus are twice more efficient than Polygon menus (12.9 vs. 6.4 items) which are themselves better than Linear menus (5.6 items). As explained in section 5.6, it is important to recall that our experiment evaluates the intentional learning of the expert mode as opposed to implicit learning since users were explicitly asked to learn the expert mode. However, the fact that Flower menus make it possible to remember the expert mode in a short amount of time suggests that users will be very likely to learn it implicitly. This contrasts with Linear menus where many users never learn the expert mode (or only very few hotkeys) because it differs from the novice mode.

While these results validate our hypotheses for learning efficiency (H1, H2) they do not exactly correspond to what we initially expected. In fact, as Flower and Polygon are both marking menus, we expected a smaller difference in performance between them, and a larger difference between Polygon and Linear menus. The following paragraphs provide some possible explanations.

First, the better memorization performance of Flower menus as compared to Polygon menus may result from a simpler mapping between gestures and orientations. As for the original marking menu design, Flower gestures are radial and thus start from the menu center so that users only have to recall the orientation of gesture endings. In contrast, Polygon menus use "tangential" gestures that involve a spatial mapping that is more complex (noticeably, Polygon menus also require the users to remember from which direction the gesture must start).

This point suggests that the directness of the mapping between gestures and spatial orientation is a major factor for the efficiency of marking menus. The argument that is usually put forward to explain why marking menus are better than Linear menus is that users learn the expert mode implicitly by repeating the same gestures in novice mode. This effect may be overestimated, and the main reason why people can easily learn the expert mode of radial marking menus may be just that their expert mode is just very easy to learn. This is in fact what our results suggest. Both Flower and Polygon menus are based on this idea of learning by repeating gestures, but only the radial design (i.e. Flower menus), that provides an easy-to-learn straightforward spatial mapping, gave much better results than Linear menus.

However, Flower menu do not only require users to recall orientations but also the curvature and the rotational direction of gestures. Our experiment showed that users had no difficulty in remembering this combination of 3 different attributes (at least for activating a set of 16 different commands). This result may be

explained by the item grouping feature of Flower menus and the "Magical number seven" of the theory of Miller [14] that states that: a) there are approximately only 7 different values that can be distinguished by users for performing a one-dimensional judgment, and; b) this number can be greatly increased by considering a set of independent variable attributes. In other words: "we can make relatively crude judgments of several things simultaneously" [14]. The design of Flower menus fits very well with this principle as it makes use of 3 different attributes having few possible different values (8 orientations x 4 curvatures x 2 rotating directions).

Combining hotkeys and marks. Finally it is important to notice that hotkeys and marks are not incompatible. Although our experiment compares marks with hotkeys, it is possible to combine these two functional expert modes: hotkeys and marks will then be redundant, defining two different ways to activate a command. By doing so, introducing Flower menus in an application will not conflict with previous habits. Moreover as for hotkeys across different applications, some flower gestures should remain the same in different applications, resulting in a common gesture vocabulary with straight gestures for frequent commands.

7. CONCLUSION

We have presented Flower menus a new type of hierarchical Marking menus that does not only support straight, but also curved gestures for any of the 8 usual orientations. Flower menus make it possible to put many commands at each menu level (they can easily support about 20 commands and even more) and thus to create as large a hierarchy as needed for common applications. Flower menus also support *within groups* as well as hierarchical groups. They can thus favor breadth organization (within groups) or depth organization (hierarchical groups): as a result, designers can lay out items in a very flexible way in order to reveal meaningful item groupings. Flower menus also conserve the advantages of classical Marking menus like "scale independence" and "eyes free selection".

Focusing on the learning performance of the expert mode, we have presented a comparative study of Flower, Linear and Polygon menus. The conducted experiment showed that Polygon and Flower menus offer better performance for learning the expert mode as compared to Linear menus. Moreover the Flower menus resulted in better performance for activation and more importantly for learning the expert mode than Polygon menus. Flower menus are thus a very efficient technique for large breadth menus. They now make possible the use of Marking menus in a wide range of conditions and are well suited for applications that require menus with many items and *within groups*.

There are several directions for future work. In addition to the study of implicit learning of the expert mode in a longitudinal experiment, we plan to compare 1- and 2-level Flower menus to study the design tradeoff between breadth organization (*within groups*) and depth organization (hierarchical groups) and its impact on learning performance.

8. ACKNOWLEDGMENTS

Many thanks to Y. Guiard, M. Tahir, A. Roudaut, and S. Malacria for numerous discussions about the Flower menus, and G. Serghiou for his help with the English

9. REFERENCES

[1] Bailly, B., Lecolinet, E., Nigay, L. (2007). Wave Menus: Improving the Novice Mode of Hierarchical Marking Menus, INTERACT'07. Springer. P. 475-488.

[2] Bailly, G., Lecolinet, E., Nigay, L. (2007). Analysis of curved gestures. Technical Report GET/ENST.

[3] Cao, X. and Zhai, S. (2007). Modeling human performance of pen stroke gestures. In ACM CHI '07. p. 1495-1504.

[4] Cockburn, A, Kristensson, P, Alexander, J and Zhai, S (2007). Hard lessons: effort-inducing interfaces benefit spatial learning. ACM CHI'07. p. 1571-1580.

[5] Grossman T., Dragicevic, P., Balakrishnan, R. (2007). Strategies for accelerating on-line learning of hotkeys. ACM CHI'07. p. 1591- 1600.

[6] Guimbretière, F., Winograd, T. (2000). FlowMenus: combining command, text and data entry. ACM UIST'00. p. 213-16.

[7] Helson, H. (1933). The fundamental propositions of gestalt psychology. Psychology Review 40, p. 13-31.

[8] Isokoski, P. Käki, M. (2002). Comparison of two touchpad-based methods for numeric entry. ACM CHI'02, pp. 25-32.

[9] Kurtenbach, G., Buxton, W. (1991). Issues in Combining Marking and Direct Manipulation Techniques, ACM UIST'91. pp. 137-144.

[10] Kurtenbach G., Buxton, W. (1993). The limits of expert performance using hierarchical marking menus. ACM CHI'93. pp. 35-42.

[11] Kurtenbach, G., Sellen, A., Buxton, W. (1993). An empirical evaluation of some articulatory and cognitive aspects of marking menus. Journal of Human Computer Interaction, 8(1), p. 1-23.

[12] Kurtenbach, G., Buxton, W. (1994). User learning and performance with marking menus. ACM CHI'94. p. 258-64.

[13] Lee, E.S., Raymond, D. R. (1993). Menu-Driven Systems. Encyclopedia of Microcomputers, Vol. 11, p. 101-127.

[14] Miller G.A., (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. The Psychological Review, 63, p. 81-97

[15] Moyle, M., Cockburn, A. (2002). Analysing Mouse and Pen Flick Gestures. CHI'02, p. 19-24.

[16] Norman, K. (1991). The Psychology of Menu selection: Designing Cognitive Control at the Human/Computer Interface. Ablex Publishing Corporation.

[17] Odell, D. L., Davis, R. C., Smith, A., and Wright, P. K. (2004). Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. GI'04, p. 17-24.

[18] Viviani, P., Terzuolo, C. (1982). Trajectory determines movement dynamics. in Neuroscience, 7(2). 431-437.

[19] Zhao, S., Balakrishnan, R. (2004). Simple vs. compound mark hierarchical marking menus. ACM UIST'04. pp. 33-44.

[20] Zhao, S., Agrawala, M., Hinckley, K. (2006). Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. ACM CHI'06. p. 1077-1087

Efficient Web Browsing on Small Screens

Hamed Ahmadi
North Dakota State University
Fargo, ND 58105, USA
hamed.ahmadi@ndsu.edu

Jun Kong
North Dakota State University
Fargo, ND 58105, USA
jun.kong@ndsu.edu

ABSTRACT
A global increase in PDA and cell phone ownership and a rise in the use of wireless services have caused mobile browsing to become an important means of Internet access. However, the small screen of such mobile devices limits the usability of information browsing and searching. This paper presents a novel method that automatically adapts a desktop presentation to a mobile presentation, proceeding in two steps: detecting boundaries between different information blocks and then re-presenting the information to fit in small screens. Distinct from other approaches, our approach analyzes both the DOM structure and the visual layout to divide the original Web page into several subpages, each of which includes closely related content and is suitable for display on the small screen. Furthermore, a table of contents is automatically generated to facilitate the navigation between different subpages. An evaluation of a prototype of our approach shows that the browsing usability is significantly improved.

Categories and Subject Descriptors
H5.2. Information interfaces and presentation: User Interfaces – Screen design / Interface styles.

General Terms
Performance, Design, Experimentation, Human Factors.

Keywords
Mobile Web browser, Adaptive interface for small screens.

1. INTRODUCTION
Using handheld devices like mobile phones and PDAs (Personal Digital Assistant) is very common today. A PDA equipped with a wireless network connection and a Web browser can access the Internet from anywhere at anytime, which could satisfy a huge number of users who need to check their emails, read the latest news or even access travel guides. However, many users still hesitate to use mobile devices because of their small screens. Users of such devices have to scroll the screen both vertically and horizontally to find the desired content, which makes information searching and browsing frustrating because most of the Web pages are designed for desktop displays.

AVI'08, 28-30 May , 2008, Napoli, Italy

A global increase in PDA and cell phone ownership and a rise in the use of wireless services have caused mobile browsing to become an important means of Internet access. Hence, it is necessary to provide a method for more user-friendly Web browsing on small screens. One straightforward method is to redesign and reconstruct Web pages, using a standard format like WML (Wireless Markup Language) to fit Web presentations on small screens. However, redesigning hundreds of Web pages requires much time and effort. In addition, maintaining the consistency of content for mobile and desktop pages is even more complex.

Therefore, the more desirable method is to automatically adapt existing Web pages from desktop presentation to mobile presentation. A practical solution is to detect *closely related content*, each of which forms a *topic* within the Web page, and to reorganize those topics in a style suitable for mobile browsing. In general, the related content detection approaches can be classified into two categories: (a) structure-based approaches [2,9,14,21] that analyze the HTML elements and Web page structure, and (b) layout-based approaches [5,11,13,32,33] that analyze the Web page layout. However, the structure-based approaches suffer from unstructured Web pages while inconsistent layout templates can cause a false detection in layout-based approaches. After different topics are discovered, the original layout can be adapted to a different presentation such as a long narrow layout [25,26] or a set of small pages [31]. However, those adaptive layouts lack an efficient navigation facility. Some researchers [3,23,24,29] provide a scaled-down overview of a Web page along with zoom in and out mechanisms to facilitate the navigation between different topics. However, users found the overview disturbing when they were glancing at the Web page [29].

This paper introduces a novel method that adapts Web pages to fit in the small screen of handheld devices and to increase browsing efficiency and user satisfaction. Different from other methods, our approach applies both the structural and visual layout information of a Web page to detect related content. Briefly, our approach proceeds in two major phases: related content detection and layout adaptation. The detection phase first identifies the common areas in a Web page, such as navigation bars and the main content, by analyzing the overall layout of the Web page. The recognized navigation links and menu bars are used to construct the global navigation in the adaptive layout, providing quick access to different pages within a Web site. Then, blocks of related content within the main content are detected based on a set of heuristic rules that summarize common patterns used to organize and present information in Web pages. Based on the detected blocks, the adaptation phase generates an adaptive style in three steps: (a) the Web page is optimized by removing redundant HTML elements (e.g. an empty element) and clutter (such as an advertisement) and by resizing images; (b) the

optimized page is divided into several subpages, each of which contains closely related content and includes navigation links to other subpages so that users can easily go back and forth between subpages; and finally (c) a table of contents is constructed, which gives an overview of the information and provides quick access to subpages for a detailed reading. We have developed a prototype (i.e., a small screen device browser – *SSD browser*) to implement our approach. The evaluation result proves that the browser improved the browsing performance.

In summary, our contributions can be summarized as the following:

- We develop a set of heuristic rules, which are used to detect boundaries between different topics from both structure and visual layout. The hybrid approach overcomes the limitation of a pure structural analysis or a pure visual analysis.
- Our approach provides an efficient navigation facility by organizing navigation links in two levels: the global level and the local level. The global level navigation takes advantage of existing navigation links (such as a navigation bar) in the original Web pages, while the local navigation is made up of a table of contents, providing a quick switch between different topics within a Web page.

The remainder of the paper is organized as follows: Section 2 compares other related work with our approach; Section 3 gives an overview of our approach; Sections 4 and 5 discuss two major phases of our approach in detail; Section 6 demonstrates a prototype system; Section 7 presents an evaluation and analyzes the results; and Section 8 provides our conclusion and plans for future work.

2. RELATED WORK

Many researchers have discussed the problem of browsing desktop-size Web pages on small screen devices and have proposed different solutions. In this section, we describe those solutions and discuss their advantages and disadvantages.

One of the most common solutions of adapting Web pages is to extract logically or semantically related content of the Web page and create an adaptive layout so that the content can be browsed more easily on small displays. Several approaches have been developed to extract the related content. One approach is to analyze the HTML DOM (Document Object Model) tree [12] or HTML elements to detect the related content [2,9,10,14,16,21, 22]. DOM is a standard for creating and manipulating in-memory representations of HTML (and XML) content. By parsing a Web page's HTML into a DOM tree, we can both extract information from large logical units and manipulate smaller units such as a specific HTML element within the structure of the DOM tree. In addition, a DOM tree is highly editable and can be easily used to reconstruct a complete Web page. However, only analyzing the DOM tree and HTML elements may not always lead into a satisfactory result because not all Web pages follow the HTML grammar to group the related content. The authors of [6,7,8] propose three methods to extract semantic textual units such as page-summarization, keyword-driven summarization, and automated view transition. They eliminate the graphics such as images and only provide a text version of the Web page. The *Semantic Partitioner Algorithm* [28] traverses the DOM tree representation of a Web page in a top-down fashion to segment the content based on *entropy*. This work focuses on gathering and separating meta-data and their instances from various kinds of Web pages, which can empower the information retrieval [28].

Analyzing the grammar rules and language syntax is another approach to extract content. One study [17] presents a visual approach with a formal foundation based on the *Spatial Graph Grammar*. This approach uses the graph grammar to analyze Web pages and automatically generates a hierarchical structure representing the composition of objects in the Web page. Then, the Web page is partitioned based on this structure. Another study [34] shows that a small set of top-ranked patterns are frequently used to form a content block. This study develops a special grammar and parser, called *2p grammar* and *best-effort parser*, respectively, to extract the content. Nevertheless, it is hard to define a generic grammar which covers different types of Web sites.

Some methods discover related content according to Web page layout and visual cues [5,11,13,32]. The *VIPS* [33] utilizes useful visual cues to obtain a partition of the page at the semantic level. The *Slicing*-Tree based transformation* [31] uses the VIPS to transform a Web page into a set of small pages, each of which fits within the small screen. This approach performs well in most of the cases. However, visual cues may not be enough to detect all the boundaries in some cases. In addition, various Web page templates with different graphic designs and dynamic objects make it hard to detect all the boundaries by using only the visual cues. A hybrid approach that uses both structural and layout analysis can provide better performance.

The layout of a Web page can be adapted by resizing and repositioning the HTML elements. The *Opera mobile browser* [26] transforms a Web page into a one-column layout with an adjusted width to fit the screen size. Although the appearance of the adaptive layout is similar to that of the original page, users must scroll a long column to find the desired content. The *ThunderHawk* [1] uses font resizing and graphic scaling to adapt the Web page. The *Skweezer* [25] reformats pages by reorganizing the physical layout of the Web page while retaining all original contents. In order to fit the small screen, the *Minimap* method [29] changes the Web page *Cascading Style Sheets* (CSS) by modifying the size of the text and limiting the maximum width of the text paragraphs. This method provides a scaled-down version of the Web page, called an *overview*, on the top of the browser viewport. However, the study showed that the overview is disturbing when the user is glancing at the page [29]. The *Gateway* [20] adapts the Web page designed for a large screen by reducing the Web page in scale to fit the small screen. Selecting a special element in the scaled Web page results in a detailed view of that element. Although the users familiar with the Web page may find the information of interest within the scaled-down overview easily, a new user may have difficulty recognizing different parts of the Web page because of the small font size and pictures. In general, the layout adaptation for small screens reorganizes and resizes the content of the Web page to fit the small screens of constrained devices, but it lacks an efficient navigation facility.

A Web page can be adapted as thumbnail images so that users can select one of the images and focus on a specific part of the Web page presented by that image [3,4,19,23,30]. The *SmartView* [24] interface provides both a document overview (e.g., a scaled-down

version of the document or a document thumbnail) and a detailed view of the selected section of the document. Thumbnails are suitable for users of handheld devices with a large display and a pointing device so that they can recognize the content of each image and select the desired one easily.

3. EFFICIENT WEB BROWSING: A HYBRID METHOD

The majority of Web page adaptation methods group closely related content together and then correspondingly generate a new style suitable for mobile browsing. However, these methods either only analyze the HTML structure of the Web page or use visual layout information. To the best of our knowledge, no existing method uses both structural and visual analysis to detect boundaries between different topics within a Web page. Our approach integrates structural analysis with visual detection to recognize closely related content, and then generates an adaptive style with an efficient navigation facility. Our approach proceeds in two major phases: related content detection and layout adaptation as shown in Figure 1.

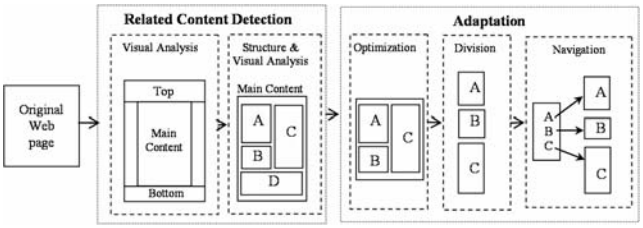


Figure 1. Web page adaptation

3.1 Related Content Detection

In order to present information clearly, Web page designers, in general, follow common patterns to organize and present information. For instance, menu bars are usually placed on the top of a Web page, or a topic usually starts with a title which has a larger font size than the rest of the text. In addition to the visual layout, HTML tags also provide useful clues for information organization. For example, <table> element is mostly used to organize a set of closely related content together, and <hr> element is used to separate different topics. We have investigated both the structural organization and layout presentation from a large number of popular Web sites in three categories (i.e., news, travel, and shopping) to recover those common patterns and summarize them as a set of heuristic rules. Although the existing patterns are not limited to those common patterns, it is very difficult to define a set of rules to cover all of them.

Based on those heuristic rules, our approach groups the related content in two steps. The first step recognizes the following major sections: (a) top, (b) main content, (c) left and right menus, (d) bottom, and (e) clutter such as an advertisement. The second step traverses the DOM tree and further partitions the main content section into several subsections, each of which contains closely related content. Those subsections will be displayed as subpages in the adaptation phase, which is discussed in detail in section 5.2.

3.2 Adaptation

The second phase of our approach generates an efficient adaptive style for mobile browsing in three steps. The first step optimizes the Web page by removing unnecessary HTML elements (e.g. empty element) and unrelated content (e.g. advertisements). Simplifying the Web page and removing clutter can facilitate the next step and highlight the main content. Then all the visible HTML elements, such as images, are resized in both width and height to fit into the small screens of handheld devices.

The second step is to divide the page into several subpages based on the screen size. The dividing procedure splits the page in a way that all closely related content detected in the previous phase is placed in the same subpage. Furthermore, each subpage provides navigation links to other subpages.

The last step is to construct navigation links between different subpages. The global navigation is set up based on navigation links and menu bars recognized in the first phase (i.e., related content detection). Then, a table of contents is built to provide a quick switch between different topics in the main content.

The next two sections describe two phases of our approach in detail.

4. RELATED CONTENT DETECTION

Content detection first partitions the original Web page into several major sections and then detects different topics within the main content.

4.1 Visual Analysis

A Web page is normally organized to include several common sections. For example, the top section usually includes the title of the page and a menu bar; the left or right section may include navigation links; and the main content is placed in the center of the Web page (see Figure 2). Clutter usually contains images and links to other Web sites, and it is normally placed in specific locations such as on the bottom or side of a page [3,15]. The content detection starts with a visual analysis, which recognizes those major sections. Then, the navigation links and menu bars (which usually contain a list of hyperlinks pointing to different Web pages within a Web site) are used to construct the global navigation which provides quick access to different pages; the main content is further divided into different topics in the next step; and the detected clutter will be removed in the optimization procedure (described in section 5.1).

In order to recognize the major sections of a Web page, the following heuristic rules are applied:

1. If a list of hyperlinks (i.e., a menu bar) or a table including a list of hyperlinks is placed within the top 200 pixels of the page [18], it is considered to be the top section.
2. If a table is placed within the lowest 150 pixels of the page

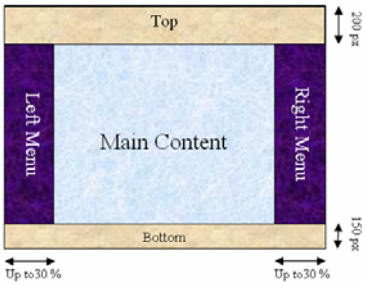


Figure 2. Common major sections of a Web page

[18], it is considered to be the bottom section.

3. If a list of hyperlinks or a table including a list of hyperlinks is placed on the left (right) side of the page, occupying up to 30 percent of the page width [18], and its upper boundary is below the top section and its lower bound is above the bottom section, then it is considered to be the left (right) menu section.
4. The remaining area is considered to be the main content.

Except for the main content, a Web page could miss some of the above sections. In that case, if an image or a text with a hyperlink pointing to a different domain is placed within the area of a missing section, it may indicate an advertisement.

The next subsection describes how to recognize different topics in the main content.

4.2 Heuristic Rules and Content Analysis

Our study indicates that HTML elements play different roles in organizing information. For instance, the element is commonly used to group related content. Therefore, a DOM node that includes several elements is probably presenting several different topics. On the other hand, some HTML elements are mainly used for layout purpose, such as . Therefore, we can detect different topics by analyzing the functionality of an HTML element. We classified the functionalities into four categories:

- Structure functionality: the and <table> elements are commonly used to organize information in a Web page, and thus provide a natural boundary between different topics. These elements do not directly include the actual content but usually contain other HTML elements (e.g. <img>, <p>, <ul> and <tr>) that actually enclose semantically related content. Moreover, the /<table> element may contain other /<table> elements, which allows designers to organize information hierarchically.
- Formatting functionality: some of the HTML elements are mostly used for formatting purposes. This category includes , <b>, <u>, <i>, <big>, <small>, and .
- Header functionality: <hi> elements, such as <h1> and <h2>, are normally used to highlight important information such as titles. In general, information presented under the same title is semantically related and should be displayed in proximity after adaptation. Therefore, we can consider <hi> as a separator between different topics.
- Separator functionality: some HTML elements like <hr> and
 are used to put separators or spaces between information. When a DOM tree node includes separator elements, it usually contains more than one topic. Moreover, continuous usage of separator elements (i.e.,
) can indicate a clear boundary between different topics.

4.2.1 Heuristic Rules

According to above functionality categories, we decide whether a DOM node contains different topics (i.e., should be divided) based on the following heuristic rules:

1. Since structure elements are usually used to group semantically related content, any DOM node that directly or transitively includes more than one structure element is considered to be a *structure node* and needs to be divided further. For example, a DOM node that encloses some <table> or elements is considered to be a structure node.
2. The header elements are commonly used to present titles. Hence, the fact that a DOM node includes multiple <hi> elements implies the presentation of different topics. Therefore, it should be divided.
3. If a DOM node includes at least one separator element, it probably contains different topics. If such an element is not the first or the last child of the node (i.e., placed at the beginning or at the end of the content presented by the node), the node should be divided.
4. If a DOM node includes only the real content (such as text) or if its direct children are of formatting elements, the node should not be divided.

Despite well-defined and common Web page design standards[1], some Web pages do not follow standard structures and are not well-formatted. For example, some designers use a bigger font size, instead of the standard <hi> element, to present a title. Therefore, the following complementary rules are added to help to detect the unstructured contents within the Web page:

5. If a DOM node has two children with different background colors, the node should be divided [33].
6. If the font size of a DOM node's child is bigger than that of the other children, the node could be divided.

We have manually evaluated the above heuristic rules on 20 popular Web sites in three categories of news, travel, and shopping (see Figure 3). The first rule successfully detected over 75 percent of boundaries between different topics in the news category and about 70 percent in the shopping category. However, the travel category had a relatively low accuracy rate, which was caused by the information organization of flight schedules. In an airline Web site, flight schedule information is normally enclosed in some <table> elements within a element (these tables include the departure time, arrival time, etc). Thus, the first rule falsely treated information enclosed in different tables as different blocks, although they belong to the same undividable topic. The second rule was very successful in the news category because the news titles and subtitles are usually enclosed in header elements. The third rule detected more than 80 percent of content boundaries on average in all three categories. The fourth rule showed a lower accuracy in the news category, which was caused by some formatting elements. For example, a news Web site could include several top news stories, each of which is enclosed in a table. Those tables are further enclosed in a formatting element, such as <b> or <font>, to highlight those top news stories. Rule 4 will falsely group those different top news stories together. This problem was solved by applying the rules in a certain order from rule 1 to rule 4. Therefore, in the previous example, the node is recognized as a dividable one since rule 1 is applied, which voids rule 4. The fifth rule has a low accuracy when it is used solely to detect the boundary because a Web page, in general, has a limited number of background colors, which can only give a preliminary division. The last rule has a satisfactory

[1] http://www.w3.org/TR/2007/WD-html-design-principles-20071126/

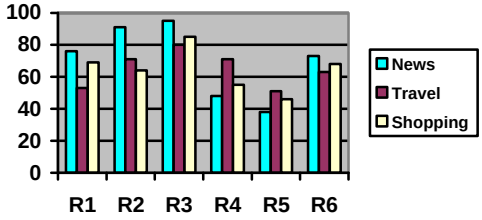


Figure 3. The average success rate of heuristic rules

accuracy because a topic usually starts with a title whose font is bigger than the rest of the text. Therefore, a bigger font strongly indicates the start of a new topic. This rule is especially useful to detect titles which are not enclosed in <hi>.

4.2.2 Content Analysis

Based on the above heuristic rules, the DOM tree is traversed to decide whether or not a DOM node includes different topics. Each non-dividable node is marked as an *atomic block* while others are marked as *composite blocks*. Blocks are organized in a hierarchy. Therefore, a large composite block can contain other composite/atomic blocks. Each block is assigned with a unique identifier, which allows quick access from a table of contents (discussed in section 5.3). Consequently, we converted the DOM tree into a new tree called a *block tree* in which leaf nodes are atomic blocks, the intermediate nodes are composite blocks, and the root indicates the complete content conveyed in a Web page. The block tree is then fed to the adaptation phase to generate an adaptive layout.

5. ADAPTATION

This section describes how to generate an adaptive layout from the block tree in three steps: (1) optimizing the structure and layout, (2) splitting the page into subpages, and (3) constructing navigation links.

5.1 Optimization

The optimization procedure optimizes both the structure and layout of the page.

```
<li class="tt2"><a href="… ">Age of Empires III</a><div></div></li>
```

Figure 4. An empty element in the Yahoo

In order to optimize the structure, we eliminate all unnecessary elements in a Web page. For instance, an empty or <table> element, which was probably used for adjusting purposes, is redundant and can be removed (Figure 4 shows a portion of the source code in the Yahoo home page that contains an empty element). Invisible elements, such as images or tables with zero width and height, are removed as well. In addition, the related content block detection procedure might generate some composite blocks with only one child (either a composite block or an atomic block). These blocks can be replaced with their child to reduce the depth of the block tree.

After the structure optimization, the layout of the page is optimized. All the visible elements, such as images, are resized in both width and height to fit the screen size. Resizing the tables is different from other elements. If a table is much bigger than the screen size, the information in the table will be presented vertically in one column.

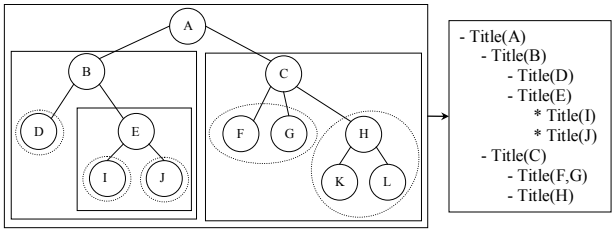


Figure 5. Constructing the page tree from a block tree and the correspondent table of contents

Another optimization is removing clutter such as advertisements detected in the visual analysis. In order to achieve more accurate advertisement detection, the <src> and <href> elements were analyzed during the traversing of the DOM tree to determine the servers to which the links refer. If an address matches a common advertisement server, the element is considered to be an advertisement [15]. This optimization is optional because some users/designers may prefer to have advertisements on the Web page.

5.2 Splitting the Page into Subpages

To avoid vertical and horizontal scrolling, we split the page into several subpages, each of which contains closely related information and is suitable for display on mobile devices. In other words, each subpage should include as much related information as possible while the screen is still large enough to display the information without scrolling. The splitting procedure traverses the block tree from top to bottom, and determines which blocks are placed in the same subpage based on the size of each block:

1. If the block tree node is a composite block and its size is larger than the screen size, the information enclosed in the block is too large to be displayed in a single subpage. This node is marked as a *composite subpage*.
2. If the block tree node is a composite block and its size is close to the screen size, the information enclosed in the block is displayed in a single subpage. The node and its direct/transitive nodes are marked as an *atomic subpage*. For instance, in Figure 5, the composite block H and its children are marked as an atomic subpage.
3. If a composite block is smaller than the screen size, its siblings are traversed and combined with the block to construct a single subpage until the last sibling is reached or the combined size is larger than the screen. All the selected nodes are marked as an atomic subpage.
4. If an atomic block is reached, mark it as an atomic subpage.

In Figure 5, dotted circles represent atomic subpages and solid squares represent composite subpages. Atomic subpages present non-overlapping information between each other while composite subpages are made of composite/atomic subpages. A *page tree* reflects the hierarchical relationships among composite and atomic subpages.

The next step constructs a table of contents based on the page tree and provides navigation links for each subpage to facilitate the navigation between subpages. In addition, at the top portion of

each subpage, there is a navigation path which allows users to quickly go back to the page at a higher level.

5.3 Constructing Navigation Links

A complex Web page can generate a large number of subpages. Therefore, efficient browsing requires a quick switch between different subpages. Our approach supports two levels of navigation: the global navigation and the local navigation. The global navigation, which can direct users to different Web pages, reuses the navigation links or menu bar, detected in the visual analysis procedure, in the home page. After directing to a specific Web page, the local navigation is presented as a table of contents, which can direct users to different topics within the page. In other words, clicking on any global navigation link directs the user to the table of contents of the designated Web page. Then users can further click on an entry in the table of contents to read through a detailed subpage. Each entry in the table of contents is mapped to a unique node in the page tree, and provides quick access to that specific subpage. The entries are organized in the same hierarchical structure as their corresponding nodes in the page tree (see Figure 5). However, due to the limited space of mobile devices, it is impractical to display all the entries in the table of contents at one time. Therefore, the table of contents is visualized as a tree. The user can expand an entry in the table of contents (if it is an intermediate node) to read its children entries or click on the link to navigate to the correspondent subpage.

After determining the hierarchical structure of the table of contents, it is critical to generate a good title for each subpage. If an information block contains a header element or includes a text whose font size is notably larger than the rest of the text, the header element or the text is considered to be the title. Otherwise, a keyword extraction method is used to determine the title [7,8]. The keyword extraction method relies on the importance of words, which depends on how often the word occurs within the text and within a larger collection of which the text is a part. However, in order to eliminate most frequently used words (such as *the*), we can design a stop list. The words in the stop list are ignored during the keyword extraction process.

6. SSD BROWSER

A prototype called the *SSD Browser* (Small Screen Devices Browser) has been developed to implement our approach. The browser allows users to customize the adaptive layouts: (a) change the size of the display window in the browser, (b) remove or preserve dynamic objects and clutter, (c) remove or preserve the formatting style of the Web page, and (d) change font sizes to small, medium, or large.

Figure 6 depicts an adaptive presentation of a Yahoo Web page generated by the SSD browser. The original Web page is presented on the left side while the subpage 1 presents the table of contents for the main contents in the original page. Clicking a title in the table of contents can direct users to a subpage which includes the detailed information. For example, clicking the "Marketplace" in the table of contents page can navigate to the subpage 2 in Figure 6 while the subpage 3 presents the topic of "news". The subpage 2 removes the original formatting style and the subpage 3 preserves the formatting style. On one hand, the formatting style can cause some cluttered adaptive layout since a style could define a fixed width/length for an element. On the other hand, the style allows users to browse the adaptive subpages in a consistent style with the original Web page. Therefore, our approach provides the flexibility to keep or remove the formatting style upon users' requests.

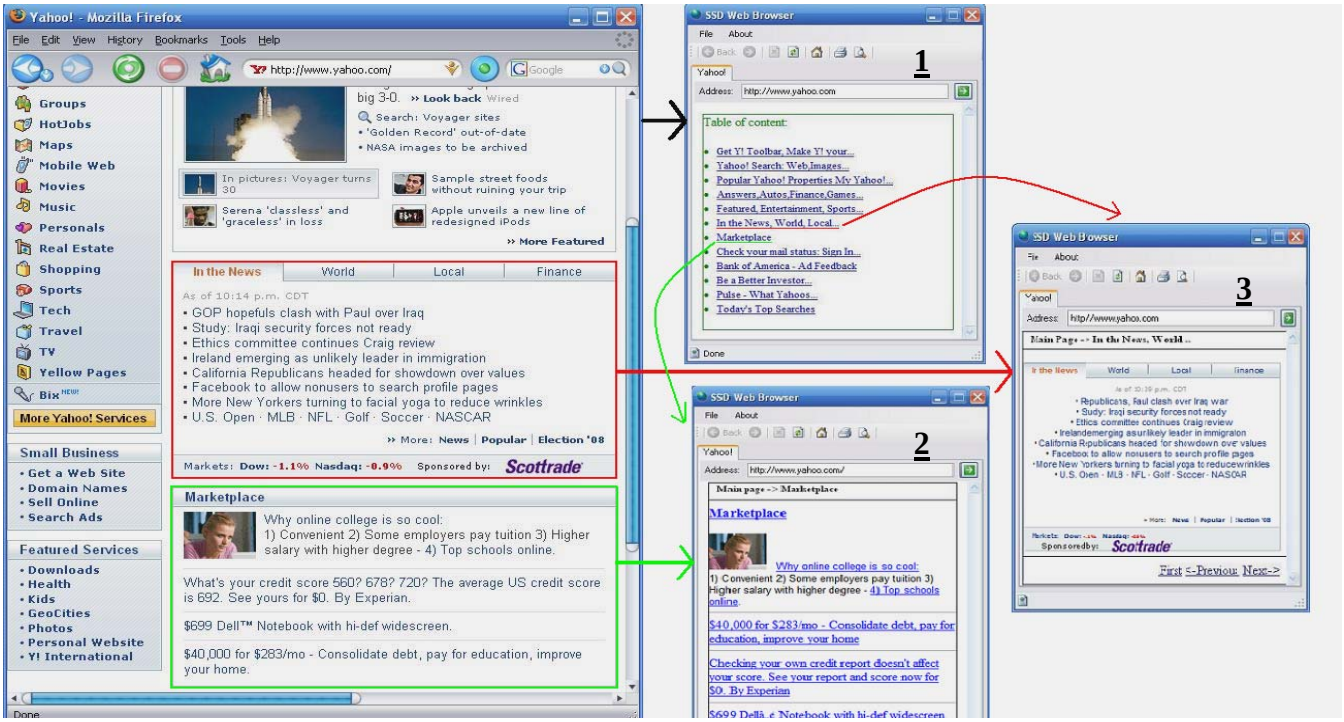


Figure 6. Adaptive presentations of a Yahoo Web page

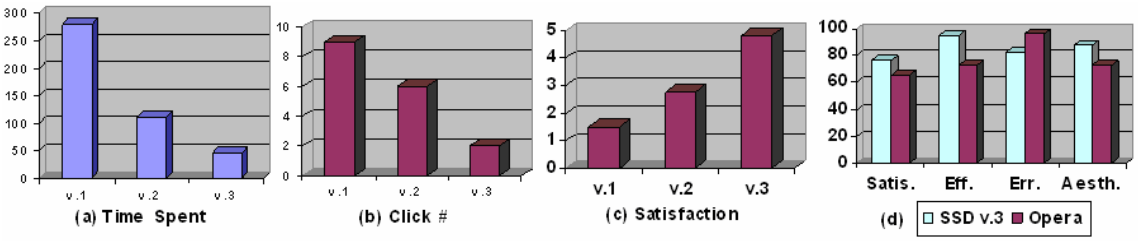


Figure 7. Evaluation Results

7. EXPERIMENTS AND EVALUATION

This section presents an empirical study of the performance and the usability of the SSD browser and compares the SSD browser with the Opera mini browser.

7.1 Study Participants

We selected 15 graduate students taking the "Human Computer Interaction" course as participants since this group of users had learned how to evaluate an interface. In order to avoid potential bias during the evaluation, the SSD browser was not introduced in the class. According to our personal interview with participants, we estimated that 25 percent of participants had never used any mobile Web browser; 45 percent had occasionally browsed the Web on mobile devices; and the remaining 30 percent had frequently accessed the Web via handheld devices.

7.2 Evaluation and Result

We conducted a two-session evaluation using PCs with a simulator.

During the first session, we evaluated the usability of the table of contents and navigation links as well as the participants' subjective satisfaction of the SSD browser. Three different versions of the SSD browser, which use the same content detection technique, were developed. The first version (v.1) provides neither the table of contents nor navigation links. The second version (v.2) provides the table of contents but not the navigation links. The last version (v.3) provides both the table of contents and the navigation links. The participants were asked to browse popular Web sites using different versions of the SSD browser and complete some tasks (e.g. the participants were asked to find a definition of human computer interaction in the HCI bibliography Web site[1]). The three versions of the SSD browser can automatically record the time participants have spent on each task, count the number of clicks to accomplish the task, and ask the user to rate his/her subjective satisfaction. Those results are saved in a textual file for further analysis. The evaluation results presented in Figure 7.a and 7.b indicate that the table of contents and navigation links significantly improved the browsing efficiency. Figure 7.c presents the participants' subjective satisfaction of each version (one indicates "not satisfied at all" and five indicates "completely satisfied").

In the second session, we compared the performance and usability of the SSD v.3 with the Opera mobile browser [26]. The Opera browser was chosen because it provides a *small screen view* (a long, one-column layout with no overview) of Web pages, which can be simulated on PCs. In addition, according to our interview with participants, the Opera browser was one of the most well-known mobile browsers. The participants were asked to complete the same tasks with the Opera mobile browser and to compare the two browsers from four aspects: subjective satisfaction, efficiency, error prevention, and aesthetics. The results are presented in Figure 7.d.

7.3 Discussion

Mobile users are likely to have an immediate, goal-directed intention of finding particular information rather than reading through a long document. Therefore, an efficient method of information searching and browsing is vital for users. From the first evaluation, the table of contents and navigation links improved the information search significantly since the average time and the number of clicks for the SSD v.3 are much less than those of the other two versions. The table of contents presents an overview of the information within a Web page, so the user can easily find the desired content and navigate to an appropriate subpage. The navigation links provide easy and efficient navigation between different subpages, so the users feel comfortable browsing different subpages.

As indicated in the first session of our evaluation, the table of contents and navigation links increase the performance of information finding. Therefore, we are not surprised that the second study shows that the users were more satisfied with the SSD v.3 than the Opera mobile browser in the aspects of subjective satisfaction, efficiency, and aesthetics. However, the Opera mobile browser provides better performance on error prevention. The reason could be the fact that the Opera has been used for a long time and users are familiar with the interface while users in this study had never used the SSD browser before. In addition, an inappropriate title in the table of contents can also increase the error rate in the SSD browser.

8. CONCLUSION AND FUTURE WORK

We introduced a novel approach that analyzes the structure and layout of a Web page, and generates a set of small pages, each of which groups closely related content. In addition, using a table of contents and navigation links improves the efficiency of information search and navigation. In the adapted presentation, our approach is able to keep the original page appearance and remove irrelevant contents. The experimental results showed that the prototype of our approach (i.e., the SSD browser) improves the efficiency and user satisfaction significantly.

We will continue to develop more sophisticated heuristic rules in order to detect semantically related content. Some of the heuristic rules have lower performance on specific Web site categories (e.g. rule 1 on the travel category). Therefore, a customization in

[1] http://www.hcibib.org

heuristic rules could conceivably improve the performance. We also believe that the block importance criteria [27] can help us to choose the most important information blocks in an adaptive layout. Dynamic Web technologies (e.g., flash) have been commonly used in the Web pages and raised a challenging issue in adaptive layouts since most of the handheld devices do not have enough resources (such as memory) to support such technologies. One practical solution is to provide alternative presentations for those dynamic contents.

9. ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their insightful and constructive comments that have helped us to significantly improve the presentation. Thanks also go to Mary Pull at NDSU for her proofreading of the present version.

10. REFERENCES

[1] A Mobile Web Browser from Bitstream, http://www.bitstream.com/wireless

[2] Baluja, S. Browsing on small screens: Recasting Web-page Segmentation into an Efficient Machine Learning Framework. *Proc. WWW '06*, 2006, 33-42.

[3] Baudisch, P., Xie, X. Wang, C., Ma, W. Collapse-to-Zoom: Viewing Web Pages on Small Screen Devices by Interactively Removing Irrelevant Content. *Proc.UIST'04*, 2004, 91-94.

[4] Bjork, S., Bertan, I., Danielesson, R., and Karlgren, J. WEST: A Web Browser for small Terminals. *Proc. UIST'99*, 1999, 187-196.

[5] Burget, R. Visual HTML Document Modeling for Information Extraction. *Proc. RAWS'05*, 2005, 17-24.

[6] Buyukkokten, O., Garcia-Molina, H., and Paepcke, A. Accordion Summarization for End-Game Browsing on PDAs and Cellular Phones. *Proc. SIGCHI'01*, 2001, 213-220.

[7] Buyukkokten, O., Kaljuvee, O., Garcia-Molina, H., Paepcke, A., and Winograd, T. Efficient Web Browsing on Hanheld Devices Using Page and Form Summarization. *TOIS* 20(1), 2002, 82-115

[8] Buyukkokten, O., Garcia-Molina, H., and Paepcke, A. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. *Proc. WWW'01*, 2001.

[9] Chen, J., Zhou, B., Shi, J., Zhang, H., and Fengwu. Q. Function-Based Object Model towards Website Adaptation. *Proc. WWW'01*, 2001, 578-596.

[10] Chen, Y., Ma, W., and Zhang, H. Detecting web page structure for adaptive viewing on small form factor devices. *Proc. WWW'03*, 2003.

[11] Chen, J., Xie, X., Ma, W., Zhang, H. Adapting Web Pages for Small-Screen Devices. *IEEE Internet Computing* 9(1), 2005, 50-56.

[12] Document Object Mode, W3C, www.w3.org/DOM.

[13] Gu, X.D., Chen, J.L., Ma, W.Y., Chen, G.L. Visual Based Content Understanding towards Web Adaptation. *Proc. AH'02*, 2002, 64-173.

[14] Gupta, S., Kaiser, G., Neistadt, D. Grimm, P. DOM-based Content Extraction of HTML Documents. *Proc. WWW'03*, 2003, 207-214.

[15] Gupta, S., Kaiser, G., Stolfo, S. Extracting Context to Improve Accuracy for HTML content extraction. *Proc. WWW'05*, 2005, 1114-1115.

[16] Kaasinen, E., Aaltonen, M., Kolari , J., Melakoski , S., and Laakko, T. Two Approaches to Bringing Internet Services to WAP Devices. *Computer Networks* 33(1), 2000, 231-246.

[17] Kong, J., Zhang, K., Zeng, X. Spatial Graph Grammars for Graphical user interfaces. *TOCHI* 13(2), 2006, 286-307.

[18] Kovacevic, M., Diligenti, M., Gori, M. and Milutinovic, V., Recognition of Common Areas in a Web Page Using Visual Information: a possible application in a page classification, *Proc. ICDM'02*, 2002.

[19] Lam, H., Baudisch, P. Summary Thumbnails: Readable Overviews for Small Screen Web Browsers. *Proc. CHI'05*, 2005, 681-690.

[20] Mackay, B. The Gateway: A Navigation technique for Migrating to Small Screens. *Proc. CHI '03*, 2003, 684-685.

[21] Maekawa, T., Hara, T., and Nishio, S. A Collaborative Web Browsing System for Multiple Mobile Users. *Proc. PerCom*, 2006, 22-35.

[22] Milic-Frayling, N., Sommerer, R. Enhanced Document Viewer for Mobile Devices. *MSR-TR-2002-114*, 2002.

[23] Milic-Frayling, N., Sommerer, R., Rodden, K., and Blackwell, A. F. Web Viewing and Search for Mobile Devices. *Proc. WWW '03*, 2003.

[24] Milic-Frayling, N., and Sommerer, R. SmartView: Flexible Viewing of Web Page Contents. *Proc*. *WWW'02*, 2002.

[25] Skweezer, http://www.skweezer.net.

[26] Small Screen Rendering (Opera Software ASA), http://www.opera.com/products/mobile/smallscreen

[27] Song, R., Liu, H. Wen, J., Ma, W. Learning Block Importance Model for Web Pages. Proc. *WWW'04*, 2004.

[28] Vadrevu, S., Gelgi, F., Davulcu, H. Semantic Partitioning of Web Pages. *Proc. WISE'05*, 2005, 107-118.

[29] Virpi, R., Popescu, A., Koivisto, A. Vartiainen, E. Minimap: A Web Page Visualization Method for Mobile Phones. *Proc. SIGCHI'06*, 2006, 35-44.

[30] Wobbrock, J., Forlizzi, J., Hudson, S., Myers, B. WebThumb: interaction techniques for small-screen browsers. *Proc. UIST '02*, 2002, 205-208.

[31] Xiao, X., Luo, Q., Hong, D., Fu, H. Slicing*-tree based web page transformation for small displays. *Proc. CIKM'05*, 2005, 303-304.

[32] Yang, Y.D., Chen, J.L. and Zhang, H.J. Adaptive Delivery of HTML Contents. *Poster Proc. WWW'00*, 2000, 24-25.

[33] Yu, S., Cai, D., Wen, J., Ma, W. Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation. *Proc. WWW'03*, 2003, 11-18.

[34] Zhang, Z., He, B., Chang, K. C. Understanding Web Query Interfaces: Best-Effort Parsing with Hidden Syntax. *Proc. SIGMOD'04*, 2004, 107-11.

Bridging the Gap Between Real Printouts and Digital Whiteboard

Peter Brandl, Michael Haller, Juergen Oberngruber, Christian Schafleitner
Media Interaction Lab
Upper Austria University of Applied Sciences
Hagenberg, Austria
firstname.lastname@fh-hagenberg.at

ABSTRACT

In this paper, we describe a paper-based interface, which combines the physical (real) with the digital world: while interacting with real paper printouts, users can seamlessly work with a digital whiteboard at the same time. Users are able to send data from a real paper to the digital world by picking up the content (e.g. images) from real printouts and drop it on the digital surface. The reverse direction for transferring data from the whiteboard to the real paper is supported through printouts of the whiteboard page that are enhanced with integrated Anoto patterns. We present four different interaction techniques that show the potential of this paper and digital world combination. Moreover, we describe the workflow of our system that bridges the gap between the two worlds in detail.

Categories and Subject Descriptors

H5.2 [Information interfaces and presentation]: User Interfaces – Graphical user interfaces.

General Terms

Design, Human Factors.

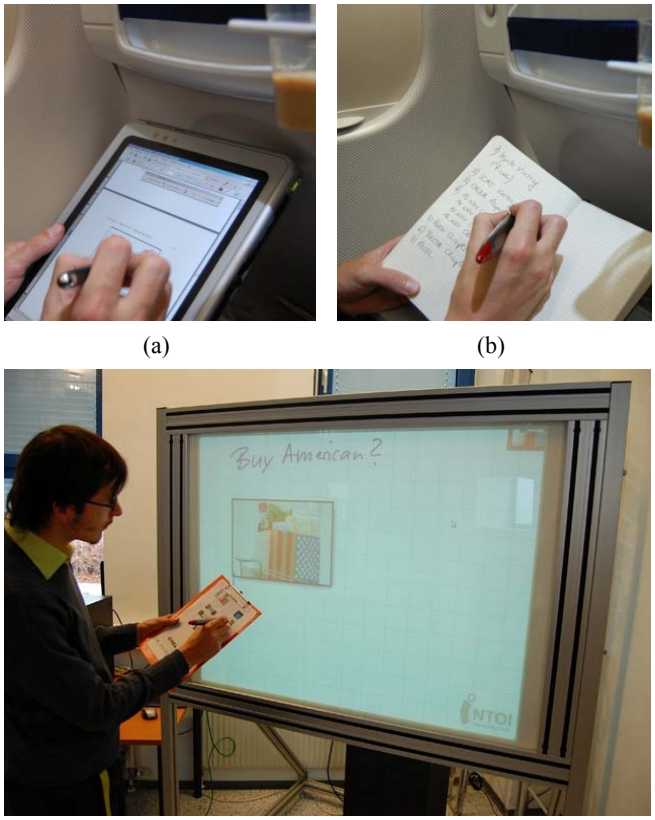
Keywords

Paper interface, digital pen, interactive paper.

1. INTRODUCTION

Designers commonly work in a studio plastered with sketches, which are either pinned on a wall or placed on flat surfaces. New drafts are designed directly on the table or a whiteboard, before creating a digital model on the computer. Many users still prefer real printouts and paper to capture rough ideas [8]. On the other hand, large interactive displays are becoming increasingly popular. Instead of replacing the current environment, we propose an approach where we integrate traditional paper into a digital environment. The support of information exchange between computer and non-computer devices seems to become more and more important. In this context, the design of solutions that seamlessly bridge the gap between these two worlds is the key factor for practical applications.

Sketching ideas and taking notes is a basic task that is performed frequently in the phase of preparing or during a meeting or presentation. For this reason, tablet PCs have been used as a good alternative to notebooks, because they allow an easy-to-use interface for sketching ideas. However, they are currently too heavy and too big to be used in different environments (e.g. people still don't like to use a tablet PC during a flight for making a quick note – instead, they still prefer pencil and paper).



(a) (b)

(c)

Figure 1: Instead of using a tablet PC during a flight (a), users still prefer pencil and paper (b). Moreover, users can go to the meeting and present their ideas to the audience either by transferring the real ink data to the digital whiteboard or by transferring printed information of the printout to the digital whiteboard (c).

AVI'08, 28-30 May , 2008, Napoli, Italy

This is the reason why paper still has a lot of advantages: it is light-weight, easy to navigate, people get a fast overview, it is easy to annotate, it is socially well accepted, and it doesn't need any power.

The usage of real paper and digital information combines the advantages of paper and additionally enhances them through the possibilities of the digital world.

In this paper, we present a new paper-based interaction device which enables a seamless usage of a digital pen for manipulating real printouts and for controlling a digital whiteboard. Users can simply pick up printed items (e.g. images, text elements) from the real printout and drop them on the digital whiteboard, as proposed by Rekimoto [11].

We propose a solution where the same pen device can be used for making notes on the real printout as well as for interacting on a digital surface. From users' observations we noticed that this feature is of importance because switching input devices during a workflow affects negatively the users' experience. Using only one device for all interactions guarantees a seamless transition between real paper and digital environment.

The pen can be also used in combination with tangible objects that act as a remote controller for the digital surface. We describe three different variations of this idea in form of an acrylic palette with embedded Anoto pattern, an enhanced ID card, and active areas located at the bottom of a printout.

2. RELATED WORK

Paper-based interfaces are becoming increasingly popular. In 1993, Johnson et al. already presented a new technology, called XAX, for bridging the paper and the electronic world [7]. The system was built on a pen-based interface and demonstrates a great framework. For tracking pen input, XAX used Dataglyphs [5], a pattern of forward and backward slashes representing ones and zeroes. Wellner's DigitalDesk [14] was one of the first interactive tables that combined both real and virtual paper into a single workspace environment. Wellner used computer vision technology to track user input.

Graphics tablets and clipboards for capturing the writing on the paper notebook are also becoming more and more popular. Mackay et al. propose a tablet prototype designed for biologists to write on real printouts, while the system automatically also creates an indexed and searchable on-line digital version [9]. The real ink was captured by the graphics tablet underneath the printout.

An increasing number of researchers are working with digital pens from Anoto[1]. The PapierCraft application from Liao et al. demonstrates an innovative combination of real and digital content using printouts and the Anoto pen [8]. Similarly, the ButterflyNet project shows a system that integrates paper notes with information captured in the field [15]. They implemented the transfer of data over a docking station, which is connected with the PC over USB. In contrast, our implementation allows streaming data from the pen to the PC over Bluetooth (BT). Although the Anoto technology has been available for more than six years, only in the last year it became possible to use a Bluetooth connection to retrieve the pen data in real-time. With a special streaming paper, the pens can send the data (position, time, pressure value, state) to the PC with a refresh rate of 50Hz. Notice that the original SDK from Anoto just allows a single connection to the PC. In contrast, our system can handle up to seven BT-pens at once.

Signer and Norrie presented a novel way of interacting with Microsoft Powerpoint [12]. The printed Powerpoint handouts are becoming an interactive paper (PaperPoint) interface for controlling the slides. PaperPoint was influenced by Palette, presented by Nelson et al. in 1999 [10]. While PaperPoint uses digital pens and the Anoto-tracking environment, Palette is based on a scanner technology for encoding the slide information. More paper-based interfaces are presented in the PhD thesis of Signer [13]. However, all of these demonstrations are always used isolated – thus, he never moved the data from one world to the other and vice versa. In contrast to his work, we support a seamless combination of the real and digital data. While making notes on a traditional paper/notebook, people can move the sketched information to the digital whiteboard and continue the discussion adding digital ink. The final results can again be stored and printed for continuing the discussion using real printouts.

More recently, Hull et al. presented "Paper-Based Augmented Reality", an interactive paper. Users can simply get additional information (e.g. website) on a mobile phone while the device to a real printout focusing on a printed website link. The advantage of their system is that they do not use real-time OCR for capturing the text; instead they are matching bounding boxes of the blurred text captured by the mobile phone with the bounding boxes stored in a huge database [6].

Our approach is influenced by two different research works: firstly, by Guimbretière's work, who presented a system where real notes are seamlessly transformed to the digital world, and vice versa [4]. In his system, users create digital documents and manipulate them either on a computer or on the paper using Anoto's technology. Users have to make their comments on the real printouts – once finished they can transfer the data to the computer over a USB-based docking station. Secondly, we got influenced by Rekimoto's Pick-and-Drop metaphor, where users seamlessly transferred digital data from one device to the other [11]. In contrast to his work, we postulate not to use tablet PCs or PDAs, but to use real printouts and real notebooks.

Our work is influenced by the previous work, but it is different in a number of important ways. Our system benefits from the following features:

- Seamless combination of both real and virtual data combined with augmented content; in contrast to related work, we allow a seamless switch between the digital and real data. Users can start with a sketch on a real paper, another person can add further annotations on the digital whiteboard, and in parallel the first person can continue the sketch on the real printout. Thus, we support a simultaneous, multi-user interaction in both the real and digital world,
- Users can simply drag-and-drop data from the real printout and move it to the digital environment (e.g. digital whiteboard); thus, we also use the same pen-based interface for interacting with the digital whiteboard,

[1] www.anoto.com

- For both worlds, users can use the same input device, a digital pen with an embedded IR-camera,
- Our system allows a high degree of accuracy with approximately 670 dpi. The accuracy is independent of the shape and size (this feature is important for the digital whiteboard),
- And finally the setup is relatively inexpensive to be manufactured. One pen costs around 200 USD and the printout only has to be printed on a paper or foil.

3. PAPER-BASED INTERACTION

We combine traditional input devices, such as pen & paper, with a digital environment. Designers can create imagery and notes on their real notebooks, make printouts with legacy software (e.g. Powerpoint, Excel, Firefox, etc.), and move them to the interactive wall for further discussion. The pen can be either used as inking or pointing device that allows selections on the paper document and data manipulations on the digital whiteboard. To change the mode for the pen, we integrated special control elements at the bottom of each page (see Figure 2). By clicking on them, the pen can change its mode or selected data can be sent to the whiteboard. In addition, we offer some options for defining the ink style including colors and stroke widths. Notice that by changing the color or stroke width, only the digital ink will be changed accordingly, but the real ink still has the same color or width.

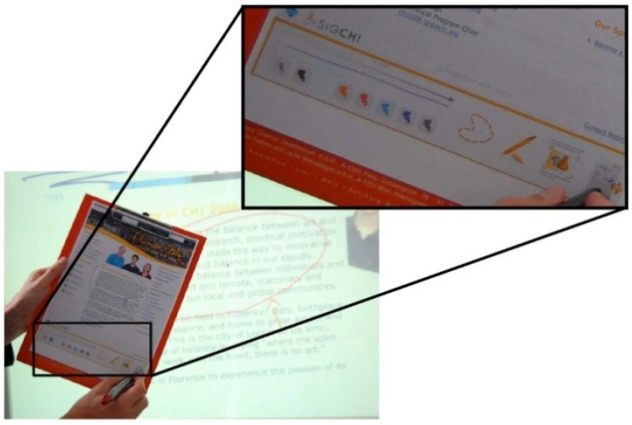


Figure 2: Special control elements printed at the bottom of the page can be used for further interaction.

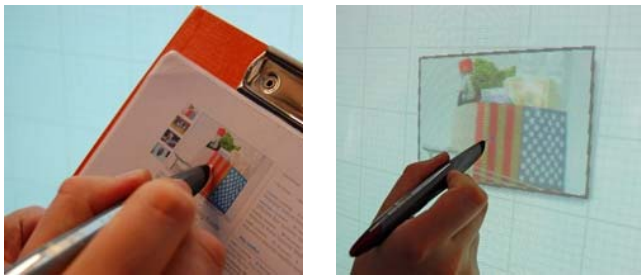
The control elements at the bottom of the page are customizable by our software and allow the integration of further interaction possibilities.

Combining the real paper with control elements and the connection to the digital whiteboard offers a variety of interesting options. Our approach is characterized by the following interaction techniques:

- Pick-and-Drop,
- Remote Control,
- Sketch-and-Send, and
- Present-and-Interact.

3.1 Pick-and-Drop

Similar to Rekimoto's Pick-and-Drop metaphor with mobile devices [11], users can pick up data from a printed document and drop it on the interactive surface, the digital whiteboard. Once in selection mode, each item of the printout becomes a selectable content and can be transferred without losing quality– since we transfer the raw data. In our scenario, users have to click with the pen on the corresponding data of the real printout. By using the digital pen, we can calculate the exact position and we can identify the according item. The data gets transferred when clicking again on the digital whiteboard (see Figure 3).



(a) (b)

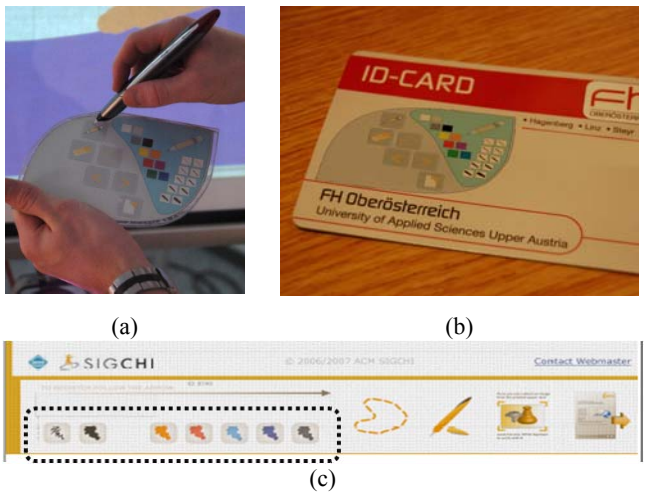
Figure 3: Users can pick up content from the real printout (a) and drop it on the digital surface (b).

Alternatively, selections on the real paper document can be sent to the digital whiteboard by clicking on the send button which is located at the bottom control panel on the paper (see Figure 2). In this case, users do not have to stand up and walk to the whiteboard to drop the selected data, but can accomplish this from their remote location.

Summarizing, users can select objects by changing the pen's mode from inking to selecting, define the corresponding part of the page, and finally move it to the digital whiteboard by directly dropping the selection with the pen or sending through the "send control" printed at the bottom of each page.

3.2 Remote Control

Influenced by the ideas of PaperPoint [12], the real printout can also be used as an alternative *input device*, where all sketched notes are sent to the digital whiteboard in real-time over BT.



(a) (b)

(c)

Figure 4: Different possibilities for the additional interaction. We either support a unique palette (a) or special ID cards where the additional functions are printed on the backside of each card (b). Alternatively, we also tested the same functions

by placing the control elements on the bottom of each printout (c).

In addition, special printed control elements on the paper allow further operation with the digital wall (e.g. adding a new page/changing the ink color of the digital flipchart etc.). In our demonstration, we implemented different possibilities for changing the ink properties (see Figure 4).

We tested our application by using a tangible tool palette, which was either embedded in an acrylic palette (a), or by adding the functions on the back of an ID card (b). Alternatively, we also printed these functions on the bottom of each printout (c).In each scenario, we simply had to put the Anoto pattern on the corresponding surface (e.g. embed it into acrylic, or to put it on the backside of the ID card). Therefore, our solution is really cheap and does not require any additional electronic sensors.

3.3 Sketch-and-Send

Our system supports additional annotations on the real printout that can be performed with the real ink of the pen. The digital version of the ink can be either visualized in real-time on the digital whiteboard or stored on the pen's integrated memory. In both variations, all data that is entered with the pen while in inking mode is processed in one or the other way.

Real-time streaming is mainly used in scenarios, where the paper printout and the digital whiteboard are in the same location. Annotations on the paper are also immediately visible on the digital whiteboard. The data transfer is accomplished through BT streaming from the Anoto pen to the whiteboard PC. Figure 5 shows an example where a user is annotating with real ink on the paper document. The results are simultaneously visible as digital ink on the whiteboard.

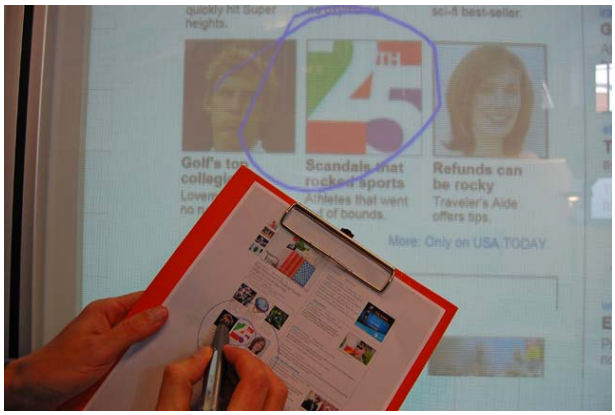


Figure 5: Annotations on the real printout are immediately visible on the digital whiteboard.

In this case, the audience can immediately see all changes done on the paper by the writing person. While all manipulations on the real paper are also immediately visible on the digital whiteboard, the system does not support a visual feedback on the real printout in the case of changes on the digital whiteboard. The only possibility is to create a new printout from the sketches done on the whiteboard.

Offering remote sketching in our system allows the participants of a meeting to keep seated around a table and share their ideas by sketching with real ink directly on a paper while the digital whiteboard acts as presentation area. This means that the users have two possibilities: they can either sit at the table and work on the digital whiteboard from their place; or they can stand up, go to the flipchart but still make their comments on the paper, which also automatically get transferred to the digital whiteboard. In both cases, all sketched information is sent to the whiteboard in real-time, regardless of the user's location. In our system, multiple people (we tested the scenario with 7 participants) can interact simultaneously – independently if they are sitting or standing.

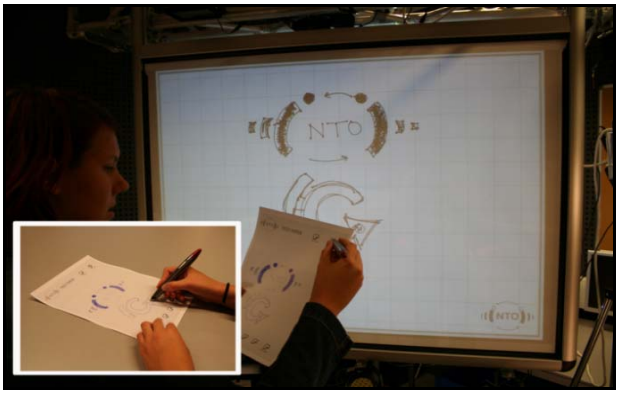


Figure 6: Users can create new sketches on the paper and send the ideas to the whiteboard for the audience for further presentation.

Working in offline mode, the sketched notes can be stored in the pen's integrated memory in advance and moved seamlessly to the whiteboard during a presentation. People can sketch offline on the real paper (e.g. during a flight as described before), come to the meeting and send all sketched data to the digital whiteboard. In this case, the pen allows to store up to 70 full-written pages. In Figure 6, we demonstrate a case where a user is preparing a sketch offline (embedded figure) and later in a meeting sends the stored data to the digital whiteboard. This whole functionality can of course also be used during a meeting to prepare sketches on the paper without displaying them in real-time on the whiteboard; presenting it to the audience can be done at any time later during the meeting.

3.4 Present-and-Interact

Finally, notes that are sent to the whiteboard can further be modified with digital ink. In addition, transferred images can be arranged and transformed on the digital surface (see Figure 7). In this scenario, we use the same pens for the interactive whiteboard as for the interaction with the real paper, so users do not have to switch to another device. Another advantage is the quality of digital data: sent data still has the same high quality as the item from the printout (e.g. the image from a website printed on the paper and sent to the digital whiteboard still has the same quality as the original image of the website).

Figure 7: The sent data (e.g. image) can be moved / rotated / scaled on the digital whiteboard.

4. THE INTERACTIVE PAPER

To capture the ink on the real printout (see Figure 8), we are using the Anoto digital pen system in combination with a Maxell pen DP-201[2]. Sketched notes can either be stored on the pen and transferred over BT using the OBEX File Transfer Protocol or directly be streamed via BT in real-time to the digital environment. Users simply have to click special checkboxes, printed on the real paper. Each page has its own paper ID. In combination with the pen ID and the position, we can easily track each ink stroke and send them to the digital whiteboard.

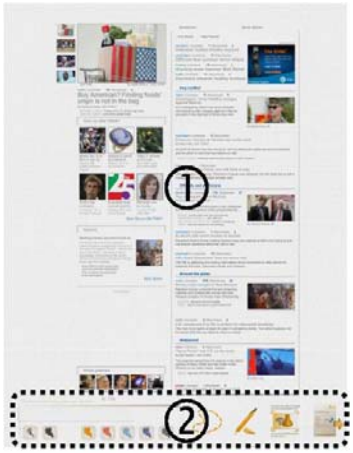


Figure 8: Each printout consists of the original content (top of the page, e.g. website) and a control panel (bottom of each page) for additional interaction possibilities and commands.

Figure 9 depicts a close-up of the printout. Anoto tracking is based on the information the pen retrieves from the dot pattern printed on the paper. Since the colors of the image are changed accordingly, our system can easily track the black dots (even the dots on the black pupil of the eye can be tracked with the digital pen without any problems).

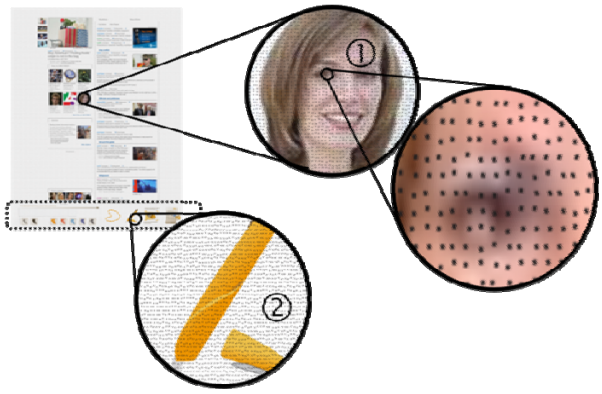


Figure 9: After exporting to an XPS file, we add an additional layer with two patterns on top of each printout for tracking the strokes with the digital pen. While the upper part of the layer (1) is used for tracking the ink strokes on the page, the lower one (2) contains a unique ID for the control elements. This pattern is equal for all pages.

We use a layer with two different kinds of pattern as overlay on the page content. The upper part of the layer contains the pattern for the "interaction" region. This pattern has to be different for each page and contains a continuous number (ID). The lower part, a unique page pattern (which is equal for all pages), is used for the special checkboxes, which are printed at the bottom of each page as depicted in Figure 8.

5. WORKFLOW

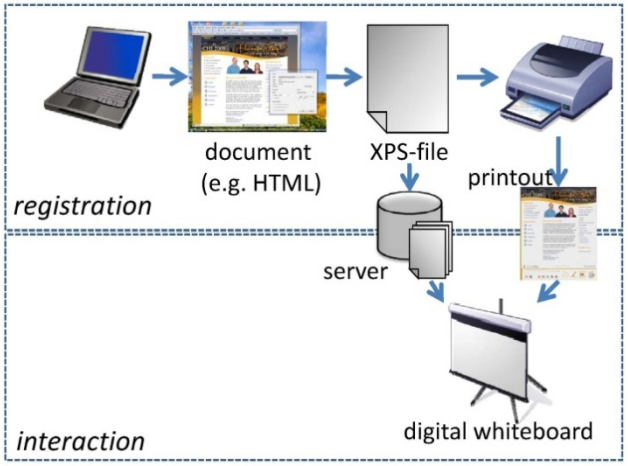


Figure 10: The document is printed with two Anoto pattern layers and also sent to the server for further interaction.

Figure 10 depicts the workflow of creating an interactive printout including a registration and an interaction phase. If users want to interact with their printout, they simply have to generate an XPS file, which is supported by all Windows applications, once the .NET Framework 3.0 is installed. This file usually contains multiple pages, which again include further content (the file can be seen as a container with different elements, such as text, images, strokes or containers again). In the next step, this file has to be printed on a color printer with our application (see Figure 11).

[2] http://partner.anoto.com/obj/docpart/43983a3deac3e.pdf

Figure 11: Our application can print an XPS document which automatically generates the Anoto pattern embedded in the printout. Moreover, the according page will be stored on the server.

In our system, we used an HP6940. As described by Guimbretière in [4], the printing process can be very complicated and time-consuming, because of the special requirements of the Anoto-based pattern. The digital pens have an embedded infrared (IR) camera. While the pattern should be printed with the black ink cartridge (which is not IR transparent and therefore visible for the IR camera), the content should be printed only with Cyan, Magenta, and Yellow (without K); the colors C, M, Y (even composed) are invisible for the IR camera. Usually, printouts contain black content and we need to find a way to make this content invisible for the IR camera. Several solutions have been discussed in [4].

Instead of removing the ink cartridge and printing the document pages twice (once with the pattern using the black ink and again with the content with C, M, Y), we propose to modify dark colors within the page, e.g. RGB (255, 255, 255) to a brighter RGB grey value, such as (169,169,169). The pages still look good and can be printed easily without any complicated hardware changes on the printer. However, the automated color management of the printer has to be switched off. Unfortunately, it is not possible to change the CMYK values directly within the XPS document.

We also store the XPS document on the server with the according ID. The server handles all documents and the corresponding pages including the page IDs used for further operation with the digital flipchart.

After the registration of the paper, users can click on the check boxes for further interaction. There are two ways of interaction: XPS content (e.g. images, paragraphs) can be easily transferred to the digital whiteboard. The objects of the corresponding XPS file are extracted and transferred accordingly. By using the XPS API, we identify digital content in the document and allow the pick-and-drop metaphor to transfer the content from the real paper to the digital world. Users can also select parts of a printed document and drop them on the digital whiteboard. Alternatively, users can make additional notes on the printout with different colors, change the stroke width, select a user-defined region, and transfer the data again to the whiteboard. For both devices, the real printout and the digital wall, we are using the pattern, which allows an easy integration of the real notebook interface. Thus, users don't have to switch the device while working with the printout or with the digital whiteboard. A closer description of the digital whiteboard can be found in [2].

6. EARLY USER FEEDBACK

In our initial pilot study we tested 6 employees from our University, who were not affiliated with this project. The overall participants' reaction was very positive. Users really liked the idea of grabbing content from the real printout and using it on a digital whiteboard. It is more convenient since people don't have to use a heavy Tablet PC. Participants also had the impression to work within *one* world.

Giving feedback on the real paper is really challenging and still a problem. The pen, used in our system, gave a vibration feedback only on errors and whenever the information has been sent successfully to the digital whiteboard. However, people asked for a better visual feedback. Especially, when they selected different ink colors, they were not sure if the system accepted their selection or not. Although the system always worked fine during the test, they expected to get a feedback. Giving feedback in the meeting room (in combination with the digital whiteboard) would be easy; in this case, we can provide audio and visual feedback on the digital whiteboard or on an interactive table. We don't have a solution yet for users working offline. However, we also have to ask how often users would change the digital color if they can't do it with the real ink of the pen.

Participants often felt lost while working with the different modes (e.g. users didn't recognize immediately that they were in the mode of annotating the paper or that they were in the grabbing mode). One of the participants proposed to have an audio feedback or a visual feedback on the digital whiteboard since the system is mainly used in combination with the digital presentation tool. Another idea, proposed by a participant was to modify the pen with corresponding LEDs.

We used two types of pens: In a first scenario, participants worked with a digital pen that had a stylus tip, which didn't leave a real ink on the paper. Consequently, participants could also use the same pen while working with the printout and while interacting with the digital whiteboard. In the second scenario, participants worked with a ballpoint tip based digital pen, which did leave a real ink on the paper. However, using a ballpoint tip would leave ink on the flipchart. On the other side switching pens would be really cumbersome. Therefore, we propose to modify the pen where users can switch between the two modes (ballpoint tip and stylus tip). Another solution would be to use ink repellent surface on the digital whiteboard.

Finally, participants also would appreciate it to get a feedback on the real printout once they change the digital content. One solution would be to track the paper and to augment the changes accordingly on the paper. Anoto tracking is not able to accomplish this without additional information about the position and orientation of the paper document on the table. We only retrieve relative coordinates on each page, but no absolute values that would allow us to align the digital overlay with the real document. For that reason, we ran first experiments with an ARTag [3] marker applied on top of each printout (see Figure 12).

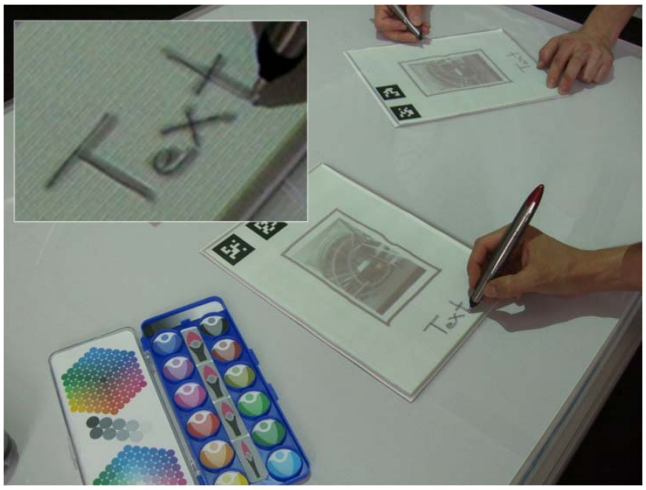


Figure 12: ARTag markers printed on each printout help to track the paper on the interactive table. However, the tracking is not accurate enough and the jittering can be cumbersome while working with the real printout. The close-up shows that the real ink of the ballpoint tip is also visible and matches with the digital ink projected from the top.

In this scenario, participants could modify the page on the digital whiteboard (e.g. embed a digital image) and the data was also visible on the real printout. Figure 12 shows that multiple users can join a session – all of them see the visualization of the same data and the printouts can be moved seamlessly on the table's surface.

Both users get the correct content visualized on the printout. We got a framerate of 30fps, which was sufficient. The markers, however, were too big. The ARTag markers are designed to track objects in 3d. In our scenario, however, we only used them to track objects which were always planar on the table's surface. We believe that we still have to spend more effort on this problem. Neither putting large markers on each page nor superimposing content on the real printout are ideal solutions for this problem.

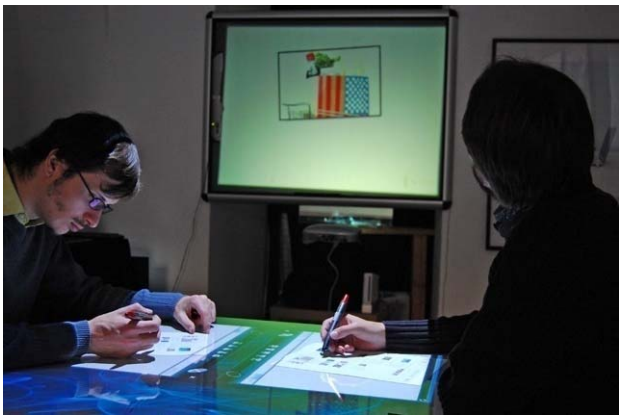


Figure 13: Users can also sit around the interactive table and interact with the digital whiteboard either through the real printout or the digital table.

An interesting observation we made was that participants discussed in a different way once they had to work together (e.g. in a brainstorming session). In a classical presentation with a flipchart or a whiteboard, where the audience is sitting around a table, the presenter is automatically the leader of the session. Usually he/she moderates the session and the audience is almost acting in the background. In our setup, however, everybody has the chance to interact immediately (see Figure 13). Everybody can send sketches, notes, and data to the digital whiteboard without standing up and going to the whiteboard. This raises the question about rights and control management, which we addressed through a social protocol in our first prototype.

7. CONCLUSION & FUTURE WORK

The integration of real notes in a digital environment seems to be a good solution for improving the performance of current digital walls and interactive tables. It combines the affordances of paper and electronic data. Related researchers found already that we will still use real printouts in the future – the myth of paperless office environment will still be true a myth in the next couple of years – in contrast, we are currently producing more paper compared to several years ago. In some domains, paper is still necessary (e.g. medical reports etc.).

Our proposed interface provides an intuitive and easy-to-use manipulation of digital information while working together on large vertical/horizontal electronic displays. Our approach is easy and inexpensive to construct and allows a scalable and multi-user environment, where simultaneous work is supported. In contrast to most related work, we use a system that allows working with the same digital pen in different situations. In this paper, we presented the usage on real printouts, on a digital rear-projected whiteboard, and on a tabletop surface.

In the future, we are further exploring the combination of real and digital paper with a special focus on how both versions can be used simultaneously.

Finally, we also have to find a better feedback for the users while working with the real printouts and find a pen-solution where the pen can be used as a stylus and as a ballpoint pen simultaneously.

8. ACKNOWLEDGEMENT

This project is sponsored by the Austrian Science Fund FFG (FHplus, contract no. 811407), voestalpine Informations-technologie GmbH, and Team 7. The authors would like to express their gratitude to the users who tested our first implementation and all the team of the Media Interaction Lab.

9. REFERENCES

1. Arregui, D., Fernstrom, C., Pacull, F., Rondeau, G., Willamowski, J., Crochon, F., and Favre-Reguillon, F. Paper-based communicating objects in the future office. 2003.
2. Brandl, P., Haller, M., Hurnaus, M., Lugmayr, V., Oberngruber, J., Oster, C., Schafleitner, C., Billinghurst, M., 2007. An Adaptable Rear-Projection Screen Using Digital Pens And Hand Gestures, in IEEE ICAT 2007, pp. 49-54, November 2007.
3. Fiala, M. ARTag, a fiducial marker system using digital techniques. In CVPR '05: Proc. of the 2005 IEEE Comp. Society Conf. on Comp. Vision and Pattern Recognition (CVPR'05) - Volume 2, pages 590–596, Washington, DC, USA, 2005. IEEE Computer Society.
4. Guimbretière, F. 2003. Paper augmented digital documents. In *Proceedings of the 16th Annual ACM Symposium on User interface Software and Technology* (Vancouver, Canada,

November 02 - 05, 2003). UIST '03. ACM Press, New York, NY, 51-60.

5. Hecht, David L. 2001. Printed embedded data graphical user interfaces. In *Computer,* 34(3): 47-55.
6. Hull., J., Erol, B., Graham, J., Ke, Q., Kishi, H., Moraleda, J., Olst, D., Paper-Based Augmented Reality. In Proceeedings of the 17th International Conference on Artificial Reality and Telexistence (Esbjerg, Denmark, November 28-30, 2007). ICAT '07. IEEE, 205-209.
7. Johnson, W., Jellinek, H., Klotz, L., Rao, R., and Card, S. 1993. Bridging the paper and electronic worlds: the paper user interface. In *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems,* IOS Press, Amsterdam, The Netherlands, 507-512.
8. Liao, C., Guimbretière, F., and Hinckley, K. 2005. PapierCraft: a command system for interactive paper. In *Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology* (Seattle, WA, USA, October 23 - 26, 2005). UIST '05. ACM Press, New York, NY, 241-244.
9. Mackay, W. E., Pothier, G., Letondal, C., Bøegh, K., and Sørensen, H. E. 2002. The missing link: augmenting biology laboratory notebooks. In *Proceedings of the 15th Annual ACM Symposium on User interface Software and Technology* (Paris, France, October 27 - 30, 2002). UIST '02. ACM, New York, NY, 41-50.
10. Nelson, L., Ichimura, S., Pedersen, E. R., and Adams, L. 1999. Palette: a paper interface for giving presentations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit* (Pittsburgh, Pennsylvania, United States, May 15 - 20, 1999). CHI '99. ACM Press, New York, NY, 354-361.
11. Rekimoto, J. 1997. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th Annual ACM Symposium on User interface Software and Technology* (Banff, Alberta, Canada, October 14 - 17, 1997). UIST '97. ACM Press, New York, NY, 31-39.
12. Signer, B. and Norrie, M. C. 2007. PaperPoint: a paper-based presentation and interactive paper prototyping tool. In *Proceedings of the 1st international Conference on Tangible and Embedded interaction* (Baton Rouge, Louisiana, February 15 - 17, 2007). TEI '07. ACM Press, New York, NY, 57-64.
13. Signer, B. Fundamental Concepts for Interactive Paper and Cross-Media Information Spaces, *Dissertation, ETH No. 16218*, Zurich, Switzerland, 2006.
14. Wellner, P. 1993. Interacting with paper on the DigitalDesk. *Commun. ACM* 36, 7 (Jul. 1993), 87-96.
15. Yeh, R., Liao, C., Klemmer, S., Guimbretière, F., Lee, B., Kakaradov, B., Stamberger, J., and Paepcke, A. 2006. ButterflyNet: a mobile capture and access system for field biology research. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* ACM Press, New York, NY, 571-580.

Exploring Blog Archives with Interactive Visualization

Indratmo, Julita Vassileva, and Carl Gutwin
Department of Computer Science
University of Saskatchewan
Saskatoon, SK S7N 5C9, Canada
{j.indratmo, julita.vassileva, carl.gutwin}@usask.ca

ABSTRACT
Browsing a blog archive is currently not well supported. Users cannot gain an overview of a blog easily, nor do they receive adequate support for finding potentially interesting entries in the blog. To overcome these problems, we developed a visualization tool that offers a new way to browse a blog archive. The main design principles of the tool are twofold. First, a blog should provide a rich overview to help users reason about the blog at a glance. Second, a blog should utilize social interaction history preserved in the archive to ease exploration and navigation. The tool was evaluated using a tool-specific questionnaire and the Questionnaire for User Interaction Satisfaction. Responses from the participants confirmed the utility of the design principles: the user satisfaction was high, supported by a low error rate in the given tasks. Qualitative feedback revealed that the decision to select which entry to read was multidimensional, involving factors such as the topic, the posting time, the length, and the number of comments on an entry. We discuss the implications of these findings for the design of navigational support for blogs, in particular to facilitate exploratory tasks.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces; H.4.3 [**Information Systems Applications**]: Communication Applications – *information browsers*.

General Terms
Design, Experimentation, Human Factors.

Keywords
Blog visualization, social interaction history, social navigation.

1. INTRODUCTION
Blogs have emerged as a new medium for communication. According to a survey report [16], in the US, about 12 million people maintain blogs, and about 57 million Internet users read blogs. Blogs promote conversation between the bloggers and their audiences by allowing users to comment on published entries, or to cite the entries of interest in their own blogs and then send notification to the source using the TrackBack protocol [27].

A blog archive contains a collection of entries arranged in reverse chronological order. Typically, only a few of most recent entries are displayed on the front page of a blog. Useful content, however, is not limited to the most recent entries; there may be many old entries in a blog that are worth reading and that offer valuable information. The value of an article about designing good visualization, for example, does not necessarily decrease with time. Therefore, users may find useful entries by browsing a blog archive.

Browsing a blog archive, however, is not well supported. Typically, blogs only provide links to monthly archives and a list of tags for selecting a subset of entries assigned with a specific keyword. This navigational support does not offer the users any cue where to find potentially useful entries—entries that have sparked lively discussion or have been read by many people. As a result, users have to rely on their own capability to select entries and navigate through a blog archive. This task becomes tedious when users browse a large collection of entries.

To support exploration of blog archives, we developed iBlogVis—an interactive visualization tool that offers a new way to browse a blog archive (see Figure 1). There are two key features of iBlogVis. First, it provides a rich overview of a blog to enable users to reason about the blog at a glance. Second, it visualizes the history of social interaction in a blog to help users identify potentially useful entries in the blog. The evaluation showed that the prototype was successful in realistic tasks.

The contributions of this research are the synthesis and application of existing visualization and interaction techniques to a new domain (blogs), and the evaluation of the resulting prototype. There are two generalizable principles embodied in the tool that can be applied to other domains. First, we show that visualizing social interaction history is useful for supporting exploratory tasks; second, our results add further evidence that providing an overview is valuable. Both quantitative and qualitative results of the study confirmed the utility of these principles. These principles can have immediate impact on real-world sites such as Blogger (www.blogger.com) and LiveJournal (www.livejournal.com) to complement their current user interface designs.

In this paper, we discuss the design and implementation of iBlogVis and present the results of our usability study. We conclude the paper by discussing the implications of this study for the design of navigational support for blogs, especially to support exploratory tasks.

AVI'08, May 28–30, 2008, Naples, Italy.

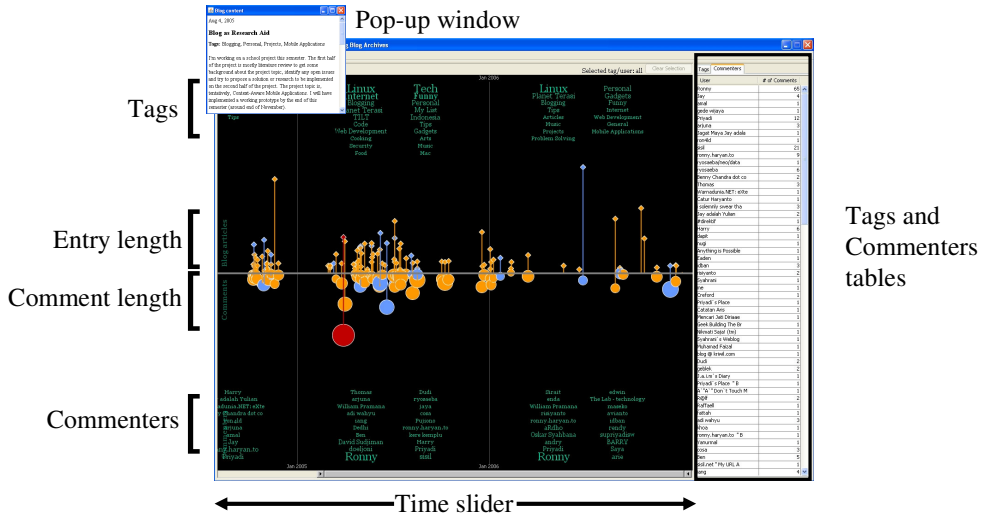


Figure 1. A screenshot of iBlogVis visualizing a collection of blog entries along a timeline.

2. RELATED WORK

Research on blog visualization currently focuses on analyzing and visualizing the link structure and the content of blogspace—a large-scale collection of blogs. Link analysis, for example, has been used to track information flow through blogspace [1, 9]. Gruhl and colleagues [9] developed a model to track the routes of topic propagation through individual blogs. Their model is similar to disease epidemic models—a blogger gets "infected" with a topic and then spreads the topic further to his/her contacts. In this context, contacts refer to the audience of a blog. Similarly, Adar and Adamic [1] developed a technique to infer the source of information spread in blogspace based on the timestamps of entries and the link structure of blogs. They visualize the inferred routes as "infection trees" where the nodes and the edges represent blogs and propagation paths.

Herring and colleagues [14] analyze link structure of blogspace to examine the interconnectedness among blogs. They use social network visualization to plot information such as inbound and outbound links and whether the links are one-way or reciprocal. The visualization reveals clusters of topic-oriented blogs that are more interconnected and reciprocally linked than "A-list" (most popular) blogs. A-list blogs, however, have more one-way inbound links, and hence are more reachable than other blogs.

Using visualization, social network analysis, and a 'sense of community survey,' Chin and Chignell [3] developed a model to discover communities in blogspace. Unlike other link analysis methods, their method does not use links mentioned in blog entries. Instead, it uses links provided by bloggers while leaving comments on other blogs (the links usually point to the bloggers' personal blogs). Chin and Chignell consider such links more explicit in suggesting a social relationship among the bloggers.

As more and more people use blogs to express their opinions, mining blog content can yield useful information. Companies are interested in knowing what people say about their products or what the hot topics are in blogspace. Analytical tools and search engines have been developed to meet such demands, including BlogScope (www.blogscope.net), Digg (digg.com), and Technorati (www.technorati.com).

Tirapat and colleagues [26] developed an interactive tool to examine whether there is a correlation between the success of a movie and the "buzz" in blogspace. They use document-clustering techniques to analyze blog entries and construct a topic map, capturing associations between movies and blog entries. Based on the topic map, the tool generates multiple views to allow users to explore different aspects of the data set.

Harris and Kamvar [12] developed a tool that looks for the phrases "I feel" and "I am feeling" in blog entries and extracts human feelings from the entries along with information about the bloggers (age, gender, and location) and the local weather conditions while the entries were posted. This information is saved; the feeling is identified (e.g., happy, sad) and then visualized as a particle. The attributes of a particle (e.g., color) represent some encoded information. A particle can be clicked to display a full sentence that describes a human feeling. The visualization tool allows users to search and sort data by feeling, gender, age, weather, geographic location, and date.

Other related projects, not specific to blog visualization, include LifeLines [20], TagLines [6], and Timeline [25]. These systems provide an interactive environment or an application programming interface (Timeline) to visualize data sets along a timeline. Specifically, LifeLines visualizes personal histories such as medical records, whereas TagLines identifies most representative tags at Flickr (www.flickr.com) during a certain time period and visualizes the evolution of these tags.

Compared to the existing work on blog visualization, our project differs in the following way. Instead of analyzing and visualizing blogspace, our work focuses on supporting exploration of individual blogs. To some extent, our goal is similar to that of Box Grid [19]. Box Grid visualizes blog entries in a grid, where the position of an entry in the grid is determined by the entry's category on the vertical axis against its posting date on the horizontal axis. Unlike our visualization tool, however, Box Grid neither visualizes the social interaction history preserved in a blog archive nor allows the users to filter the visual items. In the next section, we discuss the design and rationale behind iBlogVis.

3. DESIGN AND RATIONALE

3.1 Design Objective

The main design objective of iBlogVis is to facilitate exploration of individual blog archives. To achieve this objective, iBlogVis provides an overview of a blog and visualizes not only the content of the blog, but also the history of user interaction, thereby providing cues for social navigation [5]. We expect that users will be able to answer the following questions:

- *Content*: What is the blog about? Does the blogger post entries regularly? Do the blogger's interests change over time?
- *Social interaction history*: Did the blog receive many comments from the audience? Which entries received many comments? When did the blog start getting popular? Who are the regular commenters?

Without a tool, gathering such information normally requires that users follow a blog: reading entries and comments regularly, observing interests of the blogger, and identifying active users in the blog. Even so, some of this information is an aggregate of values over a long time, which cannot be estimated easily.

3.2 Visual Design

Blogs display entries in reverse chronological order. To reflect this organizational structure, iBlogVis arranges visual items along a timeline (see Figure 1). The visualization panel consists of two main parts. The upper part (above the timeline) visualizes the content of a blog: entries and the associated tags. The font size of a tag represents the tag's popularity during a certain year. The larger a tag, the more frequently the tag is used during the corresponding year. A blog entry is represented by a diamond shape and a line. The diamond shape provides an interface to view the content of an entry, while the length of a line represents the number of characters in an entry.

The lower part (below the timeline) displays the social interaction history contained in a blog archive. The line length and the font size have similar meanings as above, except that these visual items represent comments and commenters, respectively. The length of a line represents the total number of characters in all comments received on a particular entry, while the area of a circle represents the number of comments in that entry.

The right panel contains two tables. One displays a list of all tags and the number of entries labeled with each tag. Another table displays a list of all commenters and the number of comments they have written on the blog. These tables can be sorted alphabetically (by tag or commenter) or numerically (by the number of entries or comments). These sorting functions are to ease retrieval and reveal the popularity of tags and commenters. Users may select an item from these tables to filter items displayed in the visualization panel.

Visualization of blog content allows users to get an overview of the blog's subjects and the length of the entries. Tag visualization is supported because tagging is currently the main method for classifying entries. Tag visualization is useful for assessing whether a blog matches a user's interests, while the length of entries can serve as a retrieval cue. Furthermore, the decision to select which messages to read (in Usenet newsgroups) is influenced strongly by the subject and the size of a thread [8]. Tags and the length of entries and comments in blogs are similar to the subject and the size of a thread in Usenet, and hence may be influential as well in the exploration of a blog.

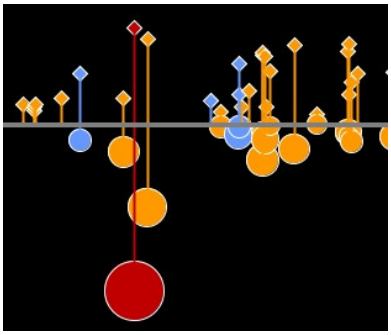


Figure 2. Read wear and social navigational cues.

Visualization of social interaction history can serve as social navigational cues. As shown in Figure 2, entries that receive many comments can be spotted easily by looking at the size of the circles. Since other users have left many comments on these entries, the entries may offer useful information. With hundreds of entries available in a blog, such social navigational cues can help users select potentially interesting entries without spending too much effort on skimming over every entry in the blog.

To further facilitate the browsing activity, iBlogVis uses the idea of read wear [15] to help users keep track of entries that have been read (blue), have not been read (orange), or the one that is currently being read (red) (see Figure 2). Read wear and its variants have been shown to improve navigation through information spaces [23, 30].

3.3 Interactive Components

The interactive components in iBlogVis are designed based on heuristic guidelines on information visualization: "overview first, zoom and filter, then details-on-demand" [22]. These components also serve as a dynamic query interface [2], which allows users to formulate queries dynamically and get feedback immediately by clicking on the items in the visualization panel (e.g., tags, commenters), selecting items from the tables in the right panel, or adjusting the time slider (see Figure 1). Highly interactive interfaces are engaging and support exploratory tasks [18], and hence fit the characteristics of browsing a blog archive.

The visualization tool starts by providing an overview of a blog archive (see Figure 1). It displays all entries and comments, most popular tags, and most frequent commenters along a timeline. The popularity of tags and commenters is aggregated on a yearly basis. This overview reveals temporal posting patterns of the blogger and enables viewers to scan the content of a blog quickly and to reason about its general structure and community dynamics.

Using iBlogVis, users can filter entries by tag, commenter, and posting time. Displaying a subset of entries labeled with a particular tag (filtering by tag) or those commented on by a particular person (filtering by commenter) can be done by clicking on the tag or the commenter. To perform this task, users can click on a tag or a commenter either in the visualization panel or in the right panel (listed in tables). These filters exist to allow users to

limit their search space by removing irrelevant or uninteresting items. Furthermore, since current blogging tools use tagging as the main mechanism for labeling, organizing, and retrieving blog entries, users would expect to be able to explore an archive by tag.

Filtering by posting time is supported by a time slider, located at the bottom of the visualization panel (see Figure 1). The slider is positioned across the visualization panel to give users a good sense of the possible query range, as the range corresponds directly to the width of the visualization. This design allows users to derive the query range visually, simply by looking at the timeline in the visualization. Users, therefore, can pay their attention to the visualization while adjusting the time slider. The time slider is also used to zoom in on the area of interest in the visualization. As the visible time range decreases, the level of details in the visualization increases: more tags and commenters are displayed whenever there is more space on screen.

Finally, users can view the content of an entry through a pop-up window by clicking on a diamond shape (representing an entry) or a circle (representing the number of comments on the entry) in the visualization (see Figure 1). When users view an entry, iBlogVis changes the color of the entry to indicate its read wear status.

4. EVALUATION METHODOLOGY

Browsing a blog is an exploratory activity in which people wander around the information space to find the entries of interest. To match the nature of this activity, we conducted an exploratory study to evaluate the usability of our visualization tool. Our tool offers a new way to browse individual blog archives, and to the best of our knowledge, there was no other blog browser that we could use for comparison. Thus, we focused on soliciting feedback on the visualization design and assessing the utility of our approach to exploring blog archives. Due to the similar purpose of the tool (i.e., facilitating exploration of an information space) and the characteristics of user tasks (i.e., exploratory search), our methods were based on earlier studies [17, 28].

The usability of iBlogVis was evaluated using both objective and subjective performance measures. The objective measure was the error rate of the completed tasks, while the subjective measure was the user satisfaction with the tool. The user satisfaction was measured using a set of tool-specific questions adapted from earlier work [17, 28] and a short version of the Questionnaire for User Interaction Satisfaction (QUIS 7.0) [4, 11]. We included only relevant QUIS items from the following categories (the number indicates the number of questions used in that category): overall user reactions (6), screen (3), terminology and system information (5), learning (4), and system capabilities (4). Besides using Likert-scale items to measure user satisfaction, we also asked the participants to explain their ratings. These exploratory questions were intended to shed some light on user practices in exploring a blog and to get constructive feedback about the tool.

Since iBlogVis offers an alternative way to browse a blog, the decision whether to use the tool depends much on the user satisfaction with the tool, which makes this criterion important. Quantitative results from this measurement, combined with observation and comments from the participants, can give valuable feedback for improving the visualization tool.

The tasks in the study were designed to evaluate three aspects of the visualization tool: (1) how effective the tool was in giving the users an overview of the content and community dynamics of a blog; (2) how well the tool worked in helping the users do things that they could do with typical blogs; and (3) how effective the tool was in giving the users information about community dynamics in a blog (e.g., revealing regular commenters and popular entries).

4.1 Participants

Nineteen students (13 males, 6 females) from the University of Saskatchewan participated in the study. Subject ages ranged from 23 to 37 years old. Each participant received a $10 honorarium. As a prerequisite, the participants had to be familiar with browsing the web and have some experience reading blogs. On a scale of one (beginner) to five (expert), the participants self rated their computer skills as an end-user at level three or above (two at level three, six at level four, and eleven at level five). On average, twelve participants browsed the web between one to five hours daily, while the rest spent more than six hours daily.

4.2 Apparatus

iBlogVis was implemented as a desktop application using Java and the prefuse toolkit [13]. It has a pre-processing module that computes the aggregate values required by the visualization and transforms the data structures of a blog into tables and the GraphML format [24]. The data set used in the study contained approximately 100 entries and 300 comments posted from October 2004 to December 2006.[1] The blog attracted a regular audience and received comments regularly. All information presented in the visualization was publicly available.

4.3 Procedure

Participants were introduced to the purpose of the study and asked to sign an informed consent form. After that, they were introduced to the features and the meaning of the visualization using an example data set. This demonstration took approximately five minutes. Then participants were given an opportunity to familiarize themselves with the visualization tool.

After a short period of practice, participants were given a blog data set and a set of tasks. There was no time limit for completing these tasks. There were 16 tasks divided into three categories:

- *Overview tasks*: understanding the timeline visualization, identifying the main topics of the blog, recognizing regular commenters, and getting a sense of popularity of the blog.
- *Typical browsing tasks*: finding the most recent entry, filtering entries by tag (by browsing and selecting a tag from the tag cloud or the tag table), and browsing monthly archives.
- *Social navigational tasks*: identifying popular entries and finding entries commented on by specific persons.

During the data collection session, we observed how the participants used the visualization tool, and took notes of comments and difficulties faced by them.

[1] Thanks to R. Haryanto for providing the data set.

Table 1. Summary of the results of the tool-specific questionnaire (n = 19).

No	Questionnaire items	Not at all easy	Not easy	Easy	Extremely easy
1.	Understanding the timeline visualization was ...	0	3	9	7
2.	Identifying the main topic of the blog was ...	0	0	8	11
3.	Identifying regular commenters in the blog was ...	0	0	5	14
4.	Getting a sense of popularity of the blog was ...	0	2	12	5
5.	Finding the most recent entry in the blog was ...	0	0	5	14
6.	Finding blog entries tagged by a specific keyword was ...	0	0	4	15
7.	Finding blog entries posted in a specific month was ...	0	4	9	6
8.	Finding popular entries in the blog was ...	0	0	8	11
9.	Finding blog entries commented on by a specific user was ...	0	1	7	11

After performing each task, the participants rated their satisfaction with the tool. Some of the questionnaire items were open-ended questions asking the participants to explain their ratings. The participants were also asked about their favorite and least favorite features of the visualization tool, desired functions that did not exist at the time, whether they would be interested in using the tool if it were integrated into blogs, and whether they had privacy concerns with the information presented in the visualization. At the end of the data collection session, the participants completed a short version of QUIS 7.0 [4, 11].

5. RESULTS

Each data collection session took up to one hour. In general, the participants did not have difficulties in learning to use iBlogVis and completing the tasks. After listening to a brief introduction about iBlogVis, all participants needed less than five minutes to feel comfortable in using the tool and ready for the tasks.

Overall, iBlogVis received positive reviews from the participants. Subjective user satisfaction was high, supported by a low error rate in the completed tasks. Quotations included in this section were from written comments from the participants.

5.1 Error Rate

Out of 16 given tasks, the number of errors made by the participants ranged from zero to three. Seven participants performed all tasks correctly; six made one mistake; five made two mistakes; and one participant made three mistakes. The average error rate was 6.25% (1 out of 16 tasks).

The most common mistake occurred when the participants were asked to find the most recent entry in the blog. Among 19 participants, five performed this task incorrectly. As some of the posting dates of the entries were close to one another, there were some overlapping visual items on the overview of the blog. Mistakes occurred when the participants simply selected a visual item on the overview that seemed to be at the right most position of the timeline. The selected item happened to be the second most recent entry in the blog. The more accurate way to perform this task was to zoom in on the area of interest to separate the overlapping items before selecting the most recent entry.

5.2 User Satisfaction

The tool-specific questionnaire for measuring user satisfaction used forced-choice fixed-scale items with four points on the scale: not at all easy, not easy, easy, and extremely easy. Table 1 provides a summary of the results. Overall, the participants expressed high satisfaction with the tool. Most items were rated easy or extremely easy. The results of this questionnaire, however, also indicate room for improvement (e.g., problems related to the timeline—see statement 1 and 7 in Table 1).

Besides evaluating task-specific functions above, the participants gave their overall reactions to the visualization tool. They rated how effective the tool was in giving overviews of the content and community dynamics of the blog. Most participants thought that the tool presented an overview of the blog content effectively. Four participants rated it highly effective; thirteen rated it effective; and two rated it not effective. In terms of providing an overview of the community dynamics, iBlogVis was rated highly effective by four users and effective by fifteen users.

The participants also rated how well iBlogVis helped them do things that they could and could not do with typical blogs. Compared to what the participants could do with typical blogs, iBlogVis was rated extremely well by seven users and well by twelve users. Additional functions of iBlogVis (things that the participants could not do with typical blogs) were rated extremely well by nine users and well by ten users.

All participants agreed that having access to the visualization tool would affect their choices of which entries to read (ten strongly agreed, nine agreed). The most common reason was that, by looking at the visualization, they would be able to see the popularity and the length of entries:

> *"I would like to know both how long an entry is and how popular it is before reading, since I would prefer to read short entries that are thought provoking/noteworthy."*

"I'd probably just read the most popular stories first to get a sense of the blogger's style and focus. Then I'd read a few of the least popular entries for comparison."

All participants thought that, while exploring a blog, having access to an overview of the blog was useful (eleven rated it extremely useful, eight rated it useful):

"It lets me know what the blog is about overall, the author's evolving interests over time, and how popular it is based on how many readers it got."

Most participants also thought that having access to the community dynamics in a blog was useful (nine rated it extremely useful, eight rated it useful, and two rated it not useful):

"It gives me an idea if the entry is worth reading. My experience is that most of the entries I am interested in reading are usually heavily commented. Also, I enjoy reading the comments but prefer many short comments to a few long comments."

Most of the participants did not have privacy concerns regarding information presented by the visualization tool, as the information was already in the public domain, and users had a choice to use pseudonyms while leaving comments. However, two participants indicated privacy concerns, and one was not sure about her attitude to this issue.

The favorite features of iBlogVis included (1) the visualization of the length of entries and comments; (2) the lists of tags and commenters; and (3) the visualization of the number of comments on blog entries.

The least favorite features of iBlogVis were mostly related to the timeline visualization and the use of a time slider to filter visual items by date. Some participants expected to see a clearer boundary between periods (e.g., a monthly boundary). They also wanted to be able to quickly select exact dates or periods by having predefined filters (e.g., by year, month, or week). Other desired features mentioned by the participants included filtering entries by multiple criteria, searching by keywords, and better highlighting the currently selected tag or commenter.

When asked whether they would be interested in using the tool if it were integrated into blogs, all participants showed interest in the tool (ten were extremely interested, nine were interested).

After completing a tool-specific questionnaire, the participants filled out a short version of QUIS 7.0 using a nine-point scale. Table 2 presents the mean score of each QUIS category used in the study and the lower and upper limits of the mean at 95% confidence level. Based on [21], the midpoint scale (five) can be used to represent mediocre user satisfaction. Compared to this value, the results show that, in all categories, the user satisfaction with iBlogVis was significantly higher than mediocre.

Table 2. QUIS results using a nine-point scale (1: low satisfaction, 9: high satisfaction, n = 19).

QUIS Category	Mean	95% Confidence Interval	
		Lower	Upper
Overall	7.670	7.216	8.124
Screen	7.860	7.295	8.425
Terminology	8.316	7.988	8.644
Learning	8.434	8.091	8.777
Capabilities	8.083	7.687	8.479

6. DISCUSSION

iBlogVis was designed based on the hypothesis that providing an overview of a blog and revealing social interaction histories would help users explore a blog archive. As presented in the previous section, both the quantitative and the qualitative responses from the participants supported this hypothesis. A common reason was that an overview and visualization of social interaction history enable users to learn about a blog quickly and to identify popular entries in the blog:

"I can get a quick overview about the blog."

"I can easily know when the blogger posted entries frequently and find which entries are more popular."

"It helps me to choose the entry that might interest me most. There might be hundreds of entries with the tag I want, and the community dynamics can help me to filter them."

Two participants, however, did not perceive having access to social interaction histories as useful. One participant wrote:

"I'm not overly concerned with comments a blog gets. I usually make up my own mind, but it's sometimes useful to know what the most popular/contentious entry was."

While all participants acknowledged the usefulness of having an overview of a blog, the usefulness of visualizing social interaction histories depends on the kind of blogs and what the users look for in the blog. Visualization of social interaction histories is particularly useful when the users want to *explore* an information space. That is, they do not have specific information to retrieve, but want to learn about an information space—its content and social dynamics within it—while hoping to find useful or interesting information within the space. In such cases, social interaction histories can provide navigational cues for the users so that they can follow the crowd to find entries that have attracted a lot of attention in the community:

"Usually the topics receiving many comments are 'hot.' Thus most likely they will be interesting for me too."

Visitors to blogs containing information that is easily outdated may receive less benefit from the visualization of social interaction histories compared to those visiting topic-oriented blogs. For example, consider a diary blog used for sharing news with friends. Information contained in these kinds of blogs may become out-of-date or irrelevant quickly. Knowing that a friend was visiting our city last week is no longer useful, as we could not meet up with him/her. Furthermore, there is little need to revisit old entries in such blogs. The readers may just want to follow the most recent entry in the blog. From their perspective, comments from other people on old entries are not important.

The content of topic-oriented blogs (e.g., programming tips, education, and research) does not easily become outdated. Old entries may still contain relevant information that is worth reading. Comments from the audience can enrich the discussion, as they may offer different perspectives or add new content to the entry. In these kinds of blogs, visualization of social interaction

histories can provide guidance for visitors to select which entries to read and to explore the blogs effectively.

6.1 Design Implications

Responses from the study participants revealed that the decision to select which entries to read was affected by factors such as the posting time, the topic, the length of entries and comments, and the number of comments on entries. To facilitate exploration of blog archives, blogs should provide navigational support that allows users to search for particular entries using these criteria. Relying only on time- and topic-oriented navigation is not enough.

The favorite feature of iBlogVis was the visualization of the length of entries and comments. Some participants mentioned that these aspects influenced their decision whether to read an entry—an initial finding that was similar to the results of a Usenet study [8]. Some users preferred to read a short, popular entry to a long one, while others might have different preferences.

A challenge for designers is how to present various attributes of blog entries effectively to users. Users should be able to see, compare, and analyze entries from different aspects simultaneously. Simply providing additional sorting functions is insufficient, as there are multiple factors involved, and it is hard to maintain all contextual information that is important to the users while the entries are rearranged based on different criteria. Moreover, users do not always want the most popular entry. What they want may be recently published entries that are not too long and receive many short comments. Formulating such queries is complicated because the criteria are vague: recently published, not too long, many short comments. Requiring users to come up with exact criteria, however, will increase their cognitive load, and hence is not a desirable solution.

Information visualization is a viable solution to this problem. Designed properly, a visualization tool can present multiple attributes of blog entries simultaneously while allowing users to compare, analyze, and select entries that match their search criteria intuitively without having to formulate complex query statements. Contextual information can be maintained by providing an overview of a blog and enabling users to interact with the information space through a dynamic query interface [2]. The power of information visualization relies on the fact that human vision is excellent at comparing, extracting, and recognizing patterns [29].

Despite its potential, information visualization is not a panacea for all navigational problems in blogs. For lookup tasks [18] such as fact retrieval, using search engines or keyword-based retrieval is more appropriate than using visualization tools because queries can be formulated easily in these tasks, and there is no or little need to compare the query results. Visualization, therefore, should be seen as a *supplement* to the current navigational support for blogs, especially to ease exploratory tasks.

From our observations, several participants tried to click on items displayed in the visualization panel, such as months, when the given tasks were relevant to the items. For example, when participants were asked to find entries posted in a specific month, some of them tried to highlight the entries by clicking on the corresponding month in the timeline. Repeated attempts to click on items in a visualization panel were also observed in a Usenet study [28]. This observation implies that users expect that each item in a visualization panel, whenever relevant to their task, can be clicked to help them perform the task at hand.

6.2 Critical Reflection

Scalability in general is a challenging issue for visualization, and applies to our tool as well. In practice, however, our experiences suggest that users want to deal with a manageable data set at a time. Using our tool, users can narrow down a larger data set by filtering entries by tag, posting time, or commenter, and then zoom in on the area of interest to reduce visual occlusion. Another approach is to provide another level of overview such as a miniature view—a global overview on a small scale [10]. When the main visualization panel cannot show all entries, a miniature view would allow users to maintain contextual information about their position in a blog archive and then select a smaller set for exploration.

As elaborated by Ellis and Dix [7], empirical evaluation of visualization tools poses several problems, such as the absence of standard data sets and user tasks. Furthermore, people usually use a visualization tool to perform exploratory tasks, making it even more difficult to come up with a set of standardized tasks and reliable, objective performance measures. While our study also inherited these limitations, we have tried to incorporate Ellis and Dix's suggestions into our study methods as follows.

Our study used both quantitative and qualitative methods. The questionnaires gave quantitative results of the study; that is, a set of numbers indicating how satisfied or dissatisfied the participants were with our visualization tool. The observation and open-ended questions, however, produced insightful information beyond these numbers. This qualitative data gave some explanation for *why* users rated certain features of the visualization tool as useful or not useful. This explanation shed some knowledge of user practices in exploring blogs and contributed to understanding in which context revealing social interaction histories contained in blog archives is perceived to be useful.

Our study used a single data set. Although taken from a real blog, the data set might not be representative. There is no guarantee that the participants would give similar ratings if the visualization tool was used to visualize different data sets, particularly those having different characteristics (e.g., photo blogs). Therefore, the utility of iBlogVis is currently limited to a certain class of blogs: that is, topic-oriented blogs that contain mostly textual entries, have medium posting frequencies (a few entries per week), and receive regular comments from the audience. Visualization of social interaction history is one of the main features of iBlogVis. Without using a data set that contains social interaction history, iBlogVis would not be able to deliver its full functionality, which consequently could affect the ratings of its utility.

Finally, most participants considered themselves to be advanced computer users, and most of them had a background in computer science. Their ability to learn and use iBlogVis might not represent the average user's ability. On the positive side, their expertise was valuable in providing constructive feedback about the system.

7. CONCLUSION

We discussed the design, implementation, and evaluation of iBlogVis—an interactive visualization tool for facilitating exploration of blogs. The tool was evaluated using various methods. First, the design rationale was explained and justified based on existing research on human-computer interaction and information visualization. Second, the usability of the tool was evaluated using both subjective and objective performance measures. The results of these measures showed that user satisfaction was high, and the average error rate of the given tasks was low. Third, the study explored the reasons behind the user satisfaction ratings qualitatively, using observation and comments from the participants. These qualitative responses have added to the understanding of blog reading behavior and how to apply visualization techniques to ease exploratory tasks in blogs.

Comments from the participants indicated that the decision to select which entries to read was affected by multiple factors. Besides the topic and the posting time of an entry, the length and the number of comments on the entry also influenced the decision. The important role of these factors was reflected in the participants' responses about their favorite feature of iBlogVis: the visualization of the length of entries and comments. Thus, to facilitate exploratory tasks, blogs should provide additional support beyond the current time- and topic-oriented navigation.

There are several directions to follow up on our initial research. From the development perspective, there are various features that can be refined or added to our prototype, such as advanced search and bookmarking facility. The prototype can be developed further as a web-based application and be used to visualize a blog in real time. Then a field study can be conducted to observe how people actually use the visualization tool to explore a blog archive.

8. REFERENCES

[1] Adar, E. and Adamic, L.A. Tracking information epidemics in blogspace. In *Proc. Web Intelligence*, 207-214, 2005.

[2] Ahlberg, C., Williamson, C., and Shneiderman, B. Dynamic queries for information exploration: an implementation and evaluation. In *Proc. CHI*, 619-626, 1992.

[3] Chin, A. and Chignell, M. A social hypertext model for finding community in blogs. In *Proc. HYPERTEXT*, 11-22, 2006.

[4] Chin, J.P., Diehl, V.A., and Norman, K.L. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proc. CHI*, 213-218, 1988.

[5] Dieberger, A., Dourish, P., Höök, K., Resnick, P., and Wexelblat, A. Social navigation: techniques for building more usable systems. *interactions* 7(6), 36-45, 2000.

[6] Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., and Tomkins, A. Visualizing tags over time. In *Proc. WWW*, 193-202, 2006.

[7] Ellis, G. and Dix, A. An explorative analysis of user evaluation studies in information visualization. In *Proc. BELIV*, 1-7, 2006.

[8] Fiore, A.T., LeeTiernan, S., and Smith, M.A. Observed behavior and perceived value of authors in usenet newsgroups: bridging the gap. In *Proc. CHI*, 323-330, 2002.

[9] Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. Information diffusion through blogspace. *SIGKDD Explor. Newsl.* 6 (2), 43-52, 2004.

[10] Gutwin, C., Roseman, M., and Greenberg, S. A usability study of awareness widgets in a shared workspace groupware system. In *Proc. CSCW*, 258-267, 1996.

[11] Harper, B., Slaughter, L., and Norman, K.L. Questionnaire administration via the WWW: a validation and reliability study for a user satisfaction questionnaire. In *Proc. WebNet*, 808-810, 1997.

[12] Harris, J. and Kamvar, S. http://www.wefeelfine.org/

[13] Heer, J., Card, S.K., and Landay, J.A. prefuse: a toolkit for interactive information visualization. In *Proc. CHI*, 421-430, 2005.

[14] Herring, S.C., Kouper, I., Paolillo, J.C., Scheidt, L.A., Tyworth, M., Welsch, P., Wright, E., and Yu, N. Conversations in the blogosphere: an analysis "from the bottom up." In *Proc. HICSS-38*, 2005.

[15] Hill, W.C., Hollan, J.D., Wroblewski, D., and McCandless, T. Edit wear and read wear. In *Proc. CHI*, 3-9, 1992.

[16] Lenhart, A. and Fox, S. Bloggers: a portrait of the internet's new storytellers. http://www.pewinternet.org/PPF/r/186/report_display.asp, 2006.

[17] Marchionini, G. From overviews to previews to answers: integrated interfaces for federal statistics. http://ils.unc.edu/~march/bls_final_report_99-00.pdf, 2000.

[18] Marchionini, G. Exploratory search: from finding to understanding. *CACM* 49 (4), 41-46, 2006.

[19] Phiffer, D. Box Grid. http://phiffer.org/projects/box-grid/

[20] Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B. LifeLines: visualizing personal histories. In *Proc. CHI*, 221-227, 1996.

[21] Quant QUIS: information about quantitative analysis. http://lap.umd.edu/quis/QuantQUIS.htm

[22] Shneiderman, B. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. IEEE Symposium on Visual Languages*, 336-343, 1996.

[23] Skopik, A. and Gutwin, C. Improving revisitation in fisheye views with visit wear. In *Proc. CHI*, 771-780, 2005.

[24] The GraphML file format. http://graphml.graphdrawing.org/

[25] Timeline. http://simile.mit.edu/timeline/

[26] Tirapat, T., Espiritu, C., and Stroulia, E. Taking the community's pulse, one blog at a time. In *Proc. ICWE*, 169-176, 2006.

[27] TrackBack Technical Specification. http://www.sixapart.com/pronet/docs/trackback_spec

[28] Viégas, F.B. and Smith, M. Newsgroup Crowds and AuthorLines: visualizing the activity of individuals in conversational cyberspaces. In *Proc. HICSS-37*, 2004.

[29] Ware, C. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, 2000.

[30] Wexelblat, A. and Maes, P. Footprints: history-rich tools for information foraging. In *Proc. CHI*, 270-277, 1999.

Content - Focused Applications

Using Subjective and Physiological Measures to Evaluate Audience-participating Movie Experience

Tao Lin
Waseda University
Shinjuku-ku Tokyo, Japan
lintao@aoni.waseda.jp

Akinobu Maejima
Waseda University
Shinjuku-ku Tokyo, Japan
akinobu@aoni.waseda.jp

Shigeo Morishima
Waseda University
Shinjuku-ku Tokyo, Japan
shigeo@waseda.jp

ABSTRACT
In this paper we subjectively and physiologically investigate the effects of the audiences' 3D virtual actor in a movie on their movie experience, using the audience-participating movie *DIM* as the object of study. In *DIM*, the photo-realistic 3D virtual actors of audience are constructed by combining current computer graphics (CG) technologies and can act different roles in a pre-rendered CG movie. To facilitate the investigation, we presented three versions of a CG movie to an audience—a *Traditional* version, its *Self-DIM (SDIM)* version with the participation of the audience's virtual actor, and its *Self-Friend-DIM (SFDIM)* version with the co-participation of the audience and his friends' virtual actors. The results show that the participation of audience's 3D virtual actors indeed cause increased subjective sense of presence and engagement, and emotional reaction; moreover, *SFDIM* performs significantly better than *SDIM*, due to increased social presence. Interestingly, when watching the three movie versions, subjects experienced not only significantly different galvanic skin response (GSR) changes on average—changing trend over time, and number of fluctuations—but they also experienced phasic GSR increase when watching their own and friends' virtual 3D actors appearing on the movie screen. These results suggest that the participation of the 3D virtual actors in a movie can improve interaction and communication between audience and the movie.

Categories and Subject Descriptors
H.5.1 **[Information Interfaces and Presentation (e.g., HCI)**]: multimedia information system.

General Terms
Measurement, Design, Human Factors

Keywords
Audience experience evaluation, physiological measures, audience-participating movie

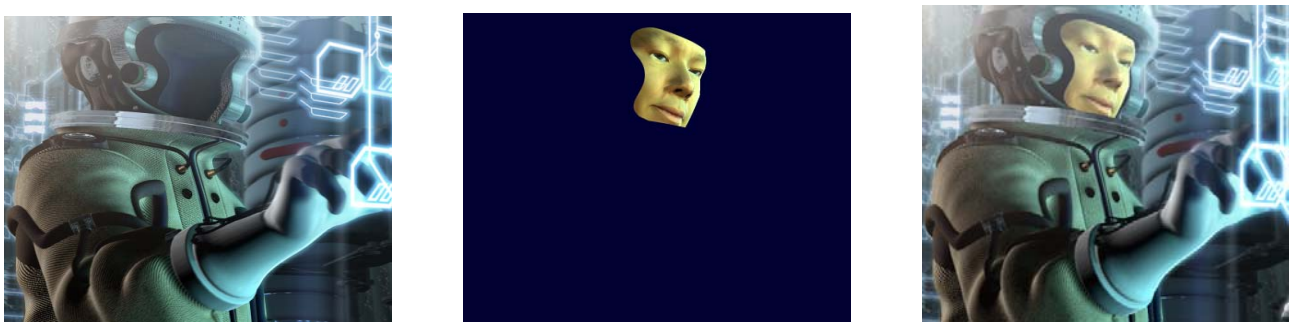
1. INTRODUCTION
The last decade has witnessed a growing interest in design information technologies and interfaces that support rich and complex user experiences, including satisfaction, joy, aesthetics, and reflection, in addition to task accomplishment. Creating audience-participating movie such as interactive drama is a topic of great currency and has been regarded as the ultimate challenge in this area of digital entertainment [1]; significant academic research have focused on developing research prototypes for interactive drama (e.g., [2, 3]). For example, in Bond Experience [3], the player's image is inserted into a virtual world where the player interacts with James Bond in vignettes taken from a longer story. Façade [2] attempts to create a real-time 3-D animated experience akin to being on stage with two live actors, but in a virtual world, who are motivated to make a dramatic situation happen.

Our research team is interested in building a new genre of audience-participating movie called *DIM* (*Dive into the Movie*, *DIM*), in which audience's 3D virtual actor (we call it CG character) is created by CG technologies and act different roles in a pre-rendered CG movie [4]. We constructed the CG characters by embedding audience's photo-realistic, 3D CG faces into the pre-rendered CG "background" roles, as shown in figure 1. *DIM* is in some sense a hybrid entertainment form, somewhere between a game and storytelling; and 1.6 million players enjoyed the *DIM* experience at the Mitsui-Toshiba pavilion at the 2005 World Exposition in Aichi, Japan.

DIM succeeds as a "pure" hedonic experience; however, how to evaluate the entertainment technology remains a research challenge. The first issue prohibiting good evaluation of entertainment technologies is the inability to define what makes a system successful. Traditional evaluation methodologies in human-computer interaction (HCI) are rooted in the productivity environment; however, for a entertainment technology, we are interested in the quality of experience and to what degree this experience is facilitated by the entertainment technology, regardless of performance[5]. Workshops such as Funology [6] have begun to establish theoretical frameworks for design and evaluation of the hedonic aspects of information technologies. Currently, the most common methods of assessing user experience are through subjective self-reporting, including questionnaires and interviews, and through objective reports from video observation and analysis [7].

Subjective evaluation through questionnaires and interviews is generalizable, and is a good approach to understanding the attitudes of users; however, subject responses may not correspond to actual experience [8]. Aware that their answers are being recorded, participants may sometimes respond with what they

AVI'08, 28-30 May , 2008, Napoli, Italy



Pre-rendered CG background role + **CG face of audience** = **CG character of audience**

Figure 1: *Construction of audience's CG character: embedding the highly realistic 3-D CG faces of audience into the pre-rendered CG background role.*

think the experimenter wishes to hear, without even realizing it. Moreover, user experience, in some sense, is process rather than outcome. Utilizing the subjective evaluation of a single data point to represent an entire condition can wash out the detailed variability of user experience, which is not conducive to analyzing the complex effects of entertainment technology on user experience. The use of video to code participants' gestures, body language, facial expressions, and verbalizations to quantify user experience is a rich source of data; however, it is a rigorous process and requires an enormous commitment of time and specialized skills. The analysis time to data sequence time ratio (AT:ST) typically ranges from 5:1 to 100:1 [9]. As a result, few researchers have relied on the methods to evaluate user experience of entertainment environments. Particularly, there is still a knowledge gap of objectively and quantitatively evaluating user experience.

Fortunately, the increasing availability of physiological sensing technologies is opening a real-time window into users' internal states. Evidence from physiology has shown that physiological measurements (e.g., galvanic skin response, heart rate, blood volume pulse) reflect autonomic nervous system (ANS) activity and can provide key information regarding the intensity and quality of an individual's internal experience [10]; moreover, physiological response is involuntary and thus is difficult to "fake" [10]. We argue that capturing, measuring, and analyzing physiological responses could provide continuous and objective access to the audience-participating movie experience offered by *DIM*, in particular used in concert with other evaluation methods (e.g., subjective reports).

For *DIM*, we define a successful experience as one in which the audience experiences greater presence and engagement, and more intensive emotional response than in a traditional movie.

Presence refers to a psychological state, specifically the subjective feeling of being transparently connected to a media experience. Lombard and Ditton define the concept of presence as "the perceptual illusion of non-mediation"[11]. Biocca draws out the distinction between physical (spatial) presence (the sense of "being there"), social presence (the sense of "being with another body"), and self presence (the sense of "feeling one's own body") [12]. In addition, researcher also presented the concept of dramatic presence [13] and it refers to an audience's sense of "being in dramatic situation."

Engagement refers to a person's involvement or interest in the content or activity of an experience, regardless of the medium. The sense of presence is not required to feel engrossed in the content—as one would be engaged in a novel, in the sense of relating to a character or being intrigued about the plot.

Emotional response includes two dimensions: *emotional arousal and valence*. Emotional arousal indicates the level of activation, ranging from very excited or energized at one extreme to very calm or sleepy at the other, and emotional valence describes the degree to which an affective experience is negative or positive [14].

This paper reports an empirical study of investigating the effects of the participation of audience's virtual actor in a movie on audience experiences using subjective and physiological evaluation techniques. Subjects were required to watch three different versions of the same movie: (a) *Traditional* version, (b) self-DIM (*SDIM*) version, and (c) a self-friend-DIM (*SFDIM*) version. *Traditional* version is like what is normally encountered at the cinema. In *SDIM* version, one subject watches her/his own CG character acting a role in the movie. In *SFDIM* version, one subject watches her/his own CG character and 15 friends' CG characters together acting 16 roles in the movie.

Our investigations show that the audience-participating movie *DIM* can support compelling movie experiences and enhance interaction and communication between audience and movie. Specifically, the experimental results show that the participation of audience's 3D CG character indeed causes increased subjective sense of presence and engagement, and more intensive emotional reaction, as compared to the traditional movie form; moreover, *SFDIM* performs significantly better than *SDIM*, possibly due to increased social presence led by the co-participation of audience and friends. More interestingly, when watching the three movie versions, audiences experienced not only significantly different galvanic skin response (GSR) changes on average—changing trend over time, and number of fluctuations—but they also experienced phasic GSR response when watching their own and friends' 3D CG character appearing on the movie screen. In addition, The GSR data can also be mirrored in subjectively reported experience.

2. PHYSIOLOGICAL MEASURES

Researchers in the domain of human factors have been concerned with optimizing the relationship between humans and their technological systems. The quality of a system has been judged not only on how it affects task performance in terms of productivity and efficiency, but on what kind of effect it has on well-being of the user. There are many examples of the use physiological metrics in the domain of human factors (see [15] for an overview).

To provide an introduction for readers unfamiliar with physiological measures, we briefly introduce the measures used, describe how these measures are collected, and explain their inferred meaning. Based on the existing literature, we chose to collect galvanic skin response (GSR) and electrocardiogram (EKG) signals to evaluate audience experience. These physiological data were gathered using ProComp Infiniti hardware and Biograph software from Thought Technologies™. Heart rate was computed from the EKG signal.

2.1 Galvanic Skin Response

GSR is a measure of skin conductivity. There are specific sweat glands (eccrine glands) that cause skin conductivity to change and result in the GSR.

Galvanic skin response can yield direct information on the activation of the sympathetic nervous system (SNS) [16]. Researches have suggested that that GSR is a linear correlate to arousal (e.g., [14]) and reflects both emotional responses as well as cognitive activity [16]. GSR is also used extensively as an indicator of experience in both non-technical domains (see [16] for a compressive review), and technical domains [17-20]. For example, a recent study suggests that change in skin conductance seems to be able to objectively measure presence in the stressful VR environment[21].

We measured GSR using surface electrodes sewn in Velcro™ straps placed around two fingers on the same hand.

2.2 Electrocardiogram (EKG)

EKG records the electrical activity of the heart over time. Cardiovascular measures such as heart rate (HR), heart rate variability (HRV) and interbeat interval (IBI) can be computed from EKG signals. We focus on the measure of HR in the study. Heart rate is dually innervated by both SNS and the parasympathetic nervous system (PNS) [22]. Increased cardiac parasympathetic activity causes the heart to slow down and is associated with information intake and attention engagement, while increased cardiac sympathetic activity causes the heart to speed up and is associated with emotional arousal, generation of preparation for action, and mobilization of various resources [23]. HR has been used to measure attention [24], presence [21], and to differentiate negative and positive emotion [22].

To collect EKG, we placed three pre-gelled surface electrodes in the standard configuration of two electrodes on the left arm and one electrode on the right arm.

3. SUBJECTIVE MEASURES

We used the ITC-Sense of Presence Inventory (ITC-SOPI), a 44-item cross-media questionnaire of presence [25], to evaluate subjective experience. The ITC-SOPI identifies four presence-related elements: spatial presence, engagement; ecological-validity/naturalness, and negative effects. This study employs 35 items addressing only the first three elements. The wording of some items was slightly altered to adapt the instrument specifically to the movie environment. Each of the items was rated on a 5-point scale, ranging from 1 (strongly disagree) to 5 (strongly agree).

Subjects rated their emotional responses in terms of valence and arousal to each of the conditions using 9-point pictorial scales. The valence scale consists of 5 graphical depictions of human faces in expressions ranging from a severe frown (−4, most negative) to a broad smile (+4, most positive). Similarly, for arousal ratings, there are 5 graphical characters varying from a state of low visceral agitation (0) to high visceral agitation (8). Emotional arousal and valence are scored on these two scales, which resemble Lang's Self-Assessment Manikin [26].

4. EXPERIMENTAL DESIGN

4.1 Experimental Tasks

We created a 12-mintute CG movie *"Space Child Adventure"* for experimental tasks completely using computer graphics technologies. The story was set in the distant future and humankind living has moved toward space due to environmental deterioration on the Earth; and a group of young people living in space venture to come back to the Earth by spaceship. During the adventurous journey, they encounter various unexpected events, such as monsters and alien life. In this CG movie, we created 20 CG roles as members of the spaceship crew and they are able to be acted by 20 CG characters of audience. The figure 2 shows an example of an audience and his CG character. In order to ensure that all participants could clearly see their CG characters in this short movie, each character was assigned a short "special-showing-time" during which their large, full-face CG images are clearly shown.

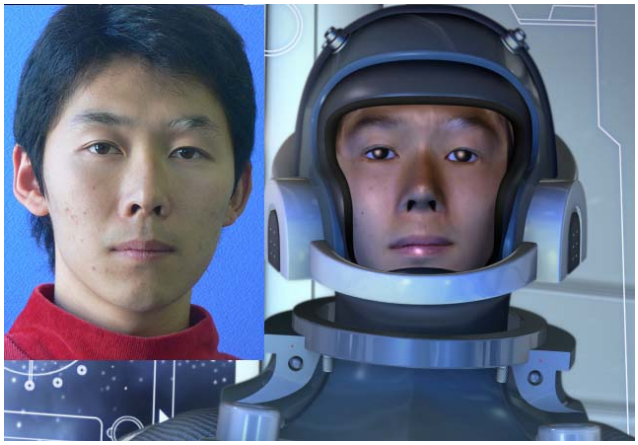


Figure 2. An example of an audience and his CG character

Subjects watched three versions of the CG movie *"Space Child Adventure"* as experimental tasks: (a) *Traditional* version, (b) *self-DIM (SDIM)* version, and (c) a *self-friend-DIM (SFDIM)* version. The order of presentation of experimental tasks is fully counterbalanced (Latin Square Design). *Traditional* version is like

what is normally encountered at the cinema, in which all CG roles were acted by strangers to the subjects. In *SDIM* version, one subject watches her/his own CG character acting a role in the movie. In *SFDIM* version, one subject watches her/his own CG character and 15 friends' CG characters together acting 16 roles in the movie. The 15 people appearing in *SFDIM* version and all subjects were recruited from the same research lab to ensure that they would be familiar with each other and easily recognize the faces appearing in the movie.

4.2 Subjects

Twelve university students, 11 male and one female, ages 21 to 24 participated in the experiment. Before the experiment, all subjects filled out a background questionnaire, used to gather information on their movie preferences, experience with the CG movie "Space Child Adventure," and personal statistics such as age and sex.

Asked how many times they have seen the movie recently, 6 watched over twice, 4 watched once, and 2 did not watch or did not watch the movie in its entirety. Notably, the movie they watched before (we called it *Original Version*) is also different from *Traditional* version of the experiemtnal tasks, since these CG roles in *Traditional* version are acted by new faces who do not appear in *Original* Version. In order to further suppress effects from the novelty of the movie's storyline and scenes, each subject was required to watch *Original* Version once before the experiment. All subjects never watch the *SFDIM* and *SDIM* versions of the movie. In some sense, we aruge that subjects have been familiar with the movie plot but the replaceable CG roles appearing in all the three experiemntal tasks are new to subjects. Thus, we believe that it is reasonable that the subjective and physiological differences among the three task conditions can be attributed to the participation of their own or friends' CG characters rather than the familiarity of the storyline and scenes.

4.3 Experimental Setting and Protocol

The experiment was conducted in a studio at Waseda University which offers a real theater atmosphere. The movies were viewed on a 150" projection-screen. Since other contextual factors such as resolution, brightness, contrast and sound effects could potentially affect users physiologically, we held these factors as constant as possible across the three conditions.

Upon arriving, subjects signed a consent form with a detailed description of the experiment, its duration, and its research purpose, and filled out background questionnaires about their gender, age, movie preference, etc. We then fitted subjects with physiological sensors, tested the placement of the physiological sensors to ensure that the signals were good, and collected physiological readings during a 5-minute resting period as the baseline for normalizing physiological data, and during the movie-watching periods. After completing each task condition, subjects rated their experiences on that task condition using questionnaires, then rested for 4 to 7 minutes. Investigators monitored physiological data during the resting periods to ensure the subjects' physiological responses returned to baseline after rest. During movie viewing, subjects were neither encouraged nor discouraged from talking. At the end of movie viewing, subjects discussed their impressions of the experiment (debriefing).

4.4 Data Analyses

Subjective data were entered into a database and analyzed using SPSS 12.0. EKG data were collected at 128 Hz, while GSR was collected at 64 Hz. Physiological data for each condition were exported into a file. Noisy EKG data may produce HR data in which two beats are counted in one sampling interval or one beat is counted in two sampling intervals. We inspected the HR data and corrected any erroneous samples.

There are other factors affecting physiological response, such as sweating, time of day, age, sex, race, temperature, and humidity, among others [17], in addition to those factors presented by experimental tasks. Thus, it is difficult to directly compare physiological readings across different task sessions, even for an individual. Physiological response should, then, be regarded as relative rather than absolute. We normalized each physiological signal to a percentage between 0 and 100 within each condition using the following formula:

$$\text{Normalized Signal}\,(i) = \frac{\text{Signal}(i) - \text{Baseline}}{\text{Signal}_{\text{Max}} - \text{Singal}_{\text{Min}}} \text{ x } 100 \ , \qquad (1)$$

where $signal_{max}$ and $signal_{min}$ refer, respectively, to maximum and minimum values during movie viewing and the rest period; and *baseline* refers to the average value of physiological data during the rest period. These normalized physiological data reflect physiological changes from baseline.

5. RESULTS AND DISCUSSION

5.1 Subjective Response

One-way ANOVA analysis and post-hoc pairwise comparison (LSD test) were used to examine differences across task conditions. Table 1 summarizes the results for subjective rating for experience.

Table 1. Results of subjectively reported experience among *Traditional (T)*, *SDIM* and *SFDIM* versions. Identifying strongly with that experience is reflected in a higher mean.

	T	*SDIM*	*SFDIM*	*F*	*p*
Spatial Presence	1.91	2.84	2.90	7.43	0.001
Engagement	2.53	3.33	3.51	6.29	0.005
Naturalness	2.79	2.82	2.72	0.19	0.83
Arousal	2.5	3.78	4.86	7.35	0.02
Valence	-0.03	1.65	2.00	6.33	0.01

A one-way ANOVA shows that there are significant differences in the ratings for spatial presence, emotional arousal, and engagement among the three versions (spatial presence: $F(2, 33) = 7.43$, $p = 0.001$; arousal: $F(2, 33) = 7.35$, $p = 0.02$; engagement: $F(2, 33) = 6.29$, $p = 0.01$)). Also, post-hoc pairwise comparisons showed that the three movie versions significantly differed from each other: *SFDIM* version elicited the highest subjective ratings for spatial presence, arousal and engagement, followed by *SDIM* and *Traditional* versions, respectively. For emotional valence, a one-way ANOVA analysis shows significant differences across the three versions ($F(2, 33) = 6.33$, $p = 0.01$); however, post-hoc pairwise comparison showed both *SDIM* and *SFDIM* versions caused significantly higher ratings than *Traditional* version, and

the difference between *SFDIM* and *SDIM* did not reach statistical significance. We did not find any significant difference in the rating for ecological-validity/naturalness. As a general observation, when subjects were asked, given a choice, which version would they choose to watch, all 12 subjects reported that they would choose to watch *SFDIM* version.

5.2 Physiological Response

A one-way ANOVA for normalized GSR shows that there are significant differences in mean GSR changes from baseline among the three task conditions (*Traditional*: 8.47%, *SDIM*: 2.58%, *SFDIM*: - 5.45%; $F(2, 33) = 45.11$, $p < 0.01$). Post-hoc pairwise comparisons also show that mean normalized GSR during the three versions differs significantly from each other: the mean GSR changes from baseline during *SFDIM* version were greatest, followed by *SDIM* and *Traditional* versions. The "contrast pattern" was consistent for 11 of 12 subjects. For normalized HR, we did not find significant average differences in the three task conditions using one-way ANOVA analysis.

One of the advantages of using physiological data to evaluate user experience is that they provide high-resolution, continuous representation. In addition to comparing the means from the three task conditions, we further examined the experiential changes of subjects over time. Using a moving average window (window = 34s), we first plotted the mean changes in GSR from baseline (see Figure 3).

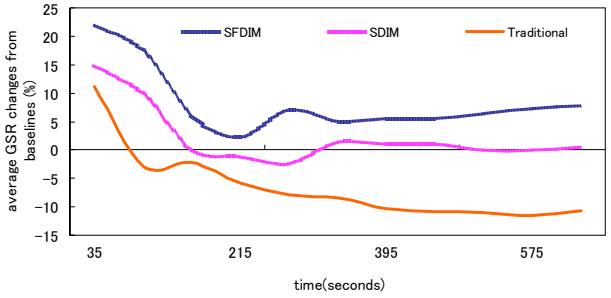


Figure 3. Mean GSR changes from resting baseline over time across the three movie versions

From Figure 3, we make the following general observations. For the first 34 seconds, all three versions produced large increases from baseline: *SFDIM* elicited the greatest GSR response, followed by *SDIM* and *Traditional* versions, indicating a substantial increase in arousal at the start of the task. This "startle effect" at the beginning of each task condition is one factor causing the increased GSR during the first 34 seconds, while the larger increases in *SFDIM* and *SDIM*, compared with *Traditional* version, could possibly be attributed to the novelty of encountering the CG characters in cast presentation during the period. Following these large increases, GSR during each task condition returns to a relatively low and smooth level; it then begins to gradually vary around baseline. Overall, there are clearly distinguishable trends in GSR changes across the three versions throughout the showing of the movies. GSR during *SFDIM* was much higher than baseline; GSR during *SDIM* remained approximately at baseline; and GSR during *Traditional* version was lower than baseline, all suggesting that the subjects experienced different physiological arousal levels across the three versions. Notably, the subjects experienced mean GSR responses lower than the resting baseline. Actually, they have been very familiar with the movie's storyline and scenes before the experiment, and this largely decreased their involvement and motivation while viewing *Traditional* version; according to post interviews, most subjects suggested that *Traditional* version made them fell sleepy like a "cradlesong." This state of "relaxation" seems to cause the lower arousal levels, resulting in lower GSR.

We also investigated HR change over time and did not find differing trends across the three versions.

In addition to the differences in GSR trends, we also observed that there was a large difference in short-term GSR response (i.e., fluctuation in seconds). As a measure of GSR fluctuation, we calculate the percentage of data points showing an increase of 10% or more over 10s. A one-way ANOVA suggests that there are significant differences in fluctuation ($F(2, 33) = 66.03$, $p < 0.01$); moreover, post-hoc pairwise comparisons showed that GSR during *SFDIM* produced the most fluctuations, followed by *SDIM* and *Traditional* versions (see Figure 4).

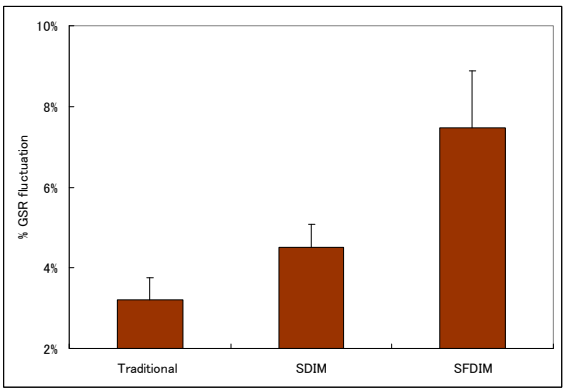


Figure 4. Significant differences in GSR fluctuation among the three movie versions.

5.2.1 GSR Response to "Self- and Friend- mirroring" Events

Establishing awareness (or a subconscious connection) between audience and their own or friends' CG characters is one anticipated way to enhance interaction between movie and audience. Thus we are especially interested in short-term physiological response when subjects watch their own and friends' CG characters appearing on the movie screen. We call these "self-mirroring" and "friend-mirroring" events. In order to inspect short-term physiological responses to these mirroring events, we synchronized physiological data with the movie scenes and examined small windows of time surrounding the events. Specifically, we compared GSR for 10 seconds before and 15 seconds after the events. Each subject could clearly see own full-face CG character three times in *SDIM* versions. That is, there are 36 self-mirroring events (3 x 12 subjects). The results show that there is a significantly larger increase in mean GSR when subjects watch their CG characters appearing on *SDIM* screen, as compared to the GSR change in response to the identical movie scene with CG roles acted by strangers in *Traditional* version (*Traditional*: 0.52%; *SDIM*: 6.15%, $t_{35} = -14.64$, $p < 0.01$).

In addition, we also examined GSR response to friend-mirroring events. We choose *Friend 7* as research target because all subjects consistently reported *Friend 7*'s CG character is most realistic and impressive in *SFDIM* version (see Figure 5). Each subject was able to clearly see full-face CG characters of *Friend 7* twice (i.e., 24 friend-mirroring events). Using the same window technology, the results show that there is a significantly larger increase in mean GSR when subjects watch *Friend 7'* CG character appearing on the screen, as compared with the identical movie scenes in *Traditional* and *SDIM* versions except that the CG role was acted by strangers (*SFDIM* vs. *SDIM*: $t_{23} = -10.59$, $p < 0.01$; *SFDIM* vs. *Traditional*: $t_{23} = -10.11$, $p < 0.01$). An example of GSR changes in response to the friend-mirroring event is shown in Figure 6.

Figure 5. Contrast between *Friend 7* and his CG character

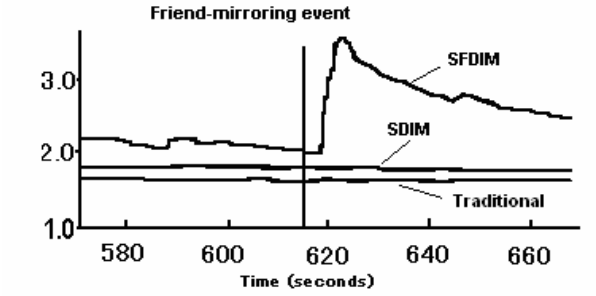


Figure 6. There is a remarkable increase in GSR when the subject watches *Friend 7*'s CG character appearing on the movie screen in *SFDIM*, as compared to the identical movie scenes in *SDIM* and *Traditional* versions except that the CG roles are acted by strangers.

We also investigated HR changes for the two events. The results show that there are no remarkable changes in HR when watching own and friends' CG characters.

5.3 Correlations between Subjective and Physiological measures

Correlations among measures were examined using the Bivariate Pearson Correlation. Normalized physiological measures first were correlated with the subjectively reported measures across 36 task sessions (12 x 3 conditions). Normalized GSR significantly positively correlated with subjective ratings for presence, emotional arousal and engagement (between GSR and arousal: $r = 0.54$, $p < 0.01$; between GSR and engagement: $r = 0.47$; $p < 0.03$; between GSR and presence: $r = 0.34$; $p < 0.05$). We also investigated the correlation between normalized HR and these subjective measures, and did not find significant results. In addition, we investigated correlations among subjective measures. The subjective ratings for engagement and presence are both significantly, positively correlated with the ratings for emotional arousal (between engagement and arousal: $r = 0.71$, $p < 0.01$; between arousal and presence: $r = 0.54$, $p < 0.02$). Between engagement and presence, there is also significantly positive correlation ($r = 0.68$, $p = 0.006$).

6. DISCUSSION AND CONCLUSIONS

SFDIM version leads to the largest mean GSR change from baseline and most GSR fluctuations, followed by *SDIM* and *Traditional* versions; timeline analysis also shows that there are remarkably different trends in GSR change over time among the three versions: GSR during *SFDIM* was much higher than baseline; GSR during *SDIM* remained approximately at baseline; and GSR during *Traditional* version was lower than baseline. These results suggest that the arousal of audience has been heightened when watching self-participating movie *DIM*. The subjective questionnaire shows that there are significant differences in subjectively reported experience: *SFDIM* version of the movie elicits the greatest sense of presence, engagement, and emotional arousal, followed by *SDIM* version, and by *Traditional* version. Moreover, correlation analyses also suggest that normalized GSR can be significantly positively correlated with subjective ratings for emotional arousal, engagement and presence.

We argue that the increased arousal may be a mediating factor and contribute to enhancing presence and engagement. Previous study[27] has suggested that the focused allocation of attentional resources to the mediated environment contributes to experience of presence; thus, given that arousal increases attention [28], the increased arousal in *SFDIM* and *SDIM* may contribute to heightening presence. It is also of note that, as a result of an increased cognitive engagement with *DIM*, there may be less cognitive resources remaining for the processing of cues signaling that the mediated environment is artificial. In conclusion, the subjective and physiological data indicate that *DIM* indeed can improve presence, engagement and emotional arousal of audience, as compared with traditional movie form.

Another important finding is that the co-participation of audience and friends in *SFDIM* version increases social presence, and this makes this version most arousing and thus results in the largest enhancement in the sense of engagement and presence among the three versions. As discussed in the section of introduction, social presence refers to the sense of "being with another body". Actually, sixteen CG characters (one subject and his 15 friends) take part in the movie as different roles in *SFDIM*, and there appears to be some communication and collaboration among these CG characters. For example, the CG character of subject must collaborate with his one of friends to control the spaceship. The post-interviews with subjects also suggest that the collaboration and communication with friends' CG character easily make audience (subjects) feel being together with their friends, and thus largely increase the sense of presence.

The most interesting result is that subjects experienced significantly phasic increases in GSR when subjects watch their own and friends' CG characters appearing on the movie screen. This seems to provide direct evidence that audiences have subconsciously "known" that they and their friends are acting different roles in the movie. We argue that the created "awareness" or the subconscious connection between viewers and their own or friends' CG characters contributes to enhancing interaction and communication between audience and movie.

We also find that *SFDIM* and *SDIM* are subjectively reported to elicit more positive emotional valence than *Traditional* version, but there is no significant difference between *SFDIM* and *SDIM* (contrary to our initial expectations). This suggests that the increased social presence may not always impact emotional valence. In the future work, more rigorous experiments will be conducted to investigate the issue.

However, we did not see significant effects from *DIM* on HR; this is inconsistent with GSR. The physiology of the two measures may explain the discrepancy between the GSR and HR results. The cardiac activity resulting in HR is dually innervated by both the SNS and the parasympathetic nervous system (PNS). Increased cardiac sympathetic activity is related to emotional activity and causes the heart to speed up, whereas increased cardiac parasympathetic activity is related to information intake and cognitive engagement, and causes the heart to slow down [23]. We argue that the increased attention and emotional responses may concurrence during the long movie experience presented, which cause PNS and SNS activity simultaneously. Thus, HR is the resultant effect of PNS and SNS activity and fails to clearly reflect the differences in a single aspect of SNS and PNS activity.

In sum, our study has created the first audience-participating movie experience *DIM* and it indeed succeeds in eliciting a greater sense of presence, engagement, and more intensive emotional response than the traditional movie form, thus enhancing interaction and communication between audience and movie; in addition, the study also shows great potential of physiological evaluation techniques in objectively evaluating user experience.

7. ACKNOWLEDGMENTS

This study was supported by the Special Coordination Funds for Promoting Science and Technology of Ministry of Education, Sports, Science and Technology of Japan.

8. REFERENCES

[1] Szilas, N. 2005. The future of interactive drama. Creativity & Cognition Studios Press.

[2] Mateas, M. and Stern, A. 2003. Facade: An Experiment in Building a Fully-Realized Interactive Drama. In Proceedings of Game Developers Conference 2003.

[3] Cavazza, M., Charles, F. and Mead, S. J. 2002. Interacting with virtual characters in interactive storytelling. In Proceedings of the first international joint conference on Autonomous agents and multi-agent systems: part 1, 318-325.

[4] Morishima, S., Maejima, A., Wemler, S., Machida, T. and Takebayashi, M. 2005. Future Cast System. In Proceedings of International Conference on Computer Graphics and Interactive Techniques.

[5] Pagulayan, R. J., Keeker, K., Wixon, D., Romero, R. and Fuller, T. 2003. User-centered design in games. The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, 883-906.

[6] Blythe, M. A. 2004. Funology: From Usability to Enjoyment. Kluwer Academic Publishers.

[7] Lazzaro, N. 2004. Why we play games: 4 keys to more emotion. In Proceedings of Game Developers Conference 2004.

[8] Marshall, D. C. and Rossman, D. G. B. 2006. Designing Qualitative Research. Sage Publications.

[9] Fisher, C. and Sanderson, P. 2005. Exploratory sequential data analysis: exploring continuous observational data. Interactions, 3, 2, 25-34.

[10] Andreassi, J. L. 2000. Psychophysiology: Human Behavior and Physiological Response. Lawrence Erlbaum Associates.

[11] Lombard, M. and Ditton. 1997. T. At the heart of it all: The concept of presence. Journal of Computer-Mediated Communication, 3, 2, 20.

[12] Biocca, F. 1997. The Cyborg's Dilemma: Progressive Embodiment in Virtual Environments. Journal of Computer-Mediated Communication, 3, 2.

[13] Kelso, M., Weyhrauch, P. and Bates, J. 1992. Dramatic presence. Presence: Teleoperators & Virtual Environments 2(1).

[14] Lang, P. J. 1995. The emotion probe. American Psychologist, 50, 5, 372-385.

[15] Mandryk, R. L., Inkpen, K. M. and Calvert, T. W. 2006. Using psychophysiological techniques to measure user experience with entertainment technologies. Behaviour & Information Technology, 25, 2, 141-158.

[16] Boucsein, W. 1992. Electrodermal Activity. Plenum Press, New York.

[17] Ward, R. D. and Marsden, P. H. 2003. Physiological responses to different WEB page designs. International Journal of Human-Computer Studies, 59, 1-2, 199-212.

[18] Wilson, G. and Sasse, M. A. 2000. Do users always know what's good for them? Utilizing physiological responses to assess media quality. In Proceedings of HCI 2000.

[19] Wilson, G. M. and Sasse, M. A. 2000. Investigating the Impact of Audio Degradations on Users: Subjective vs. Objective Assessment Methods. In Proceedings of OZCHI 2004.

[20] Mandryk, R. L., Atkins, M. S. and Inkpen, K. M. 2006. A continuous and objective evaluation of emotional experience with interactive play environments. In Proceedings of the SIGCHI conference on Human Factors in computing systems, 1027-1036.

[21] Meehan, M., Insko, B., Whitton, M. and Brooks Jr, F. P. 2002. Physiological measures of presence in stressful virtual environments. In Proceedings of the 29th annual conference on Computer graphics and interactive techniques, 645-652.

[22] Papillo, J. F. and Shapiro, D. 1990.The cardiovascular system. Cambridge University Press.

[23] Ravaja, N. 2004. Contributions of Psychophysiology to Media Research: Review and Recommendations. Media Psychology, 6, 2, 193-235.

[24] Weber, E. J., Van der Molen, M. W. and Molenaar, P. C. 1994. Heart rate and sustained attention during childhood: age changes in anticipatory heart rate, primary bradycardia, and

respiratory sinus arrhythmia. Psychophysiology, 31, 2, 164-174.

[25] Lessiter, J., Freeman, J., Keogh, E. and Davidoff, J. 2001. A Cross-Media Presence Questionnaire: The ITC-Sense of Presence Inventory. Presence: Teleoperators & Virtual Environments, 10, 3, 282-297.

[26] Lang, P. J. 1980. Behavioral treatment and bio-behavioral assessment: Computer applications. In J. B. Sidowski, J. H. Johnson, & T. A. Williams (Eds.), Technology in mental health care delivery systems. Norwood, NJ: Ablex Publishing.

[27] Vorderer, P., Wirth, W., Saari, T., Gouveia, F. R., Biocca, F., Jäncke, F., Böcking, S., Hartmann, T., Klimmt, C. and Schramm, H. 2003. Constructing Presence: Towards a two-level model of the formation of Spatial Presence. Unpublished report to the European Community, Project Presence: MEC (IST-2001-37661). Hannover, Munich, Helsinki, Porto, Zurich.

[28] Gatchel, R. J., Gaas, E., King, J. M. and McKinney, M. E. 1977. Effects of arousal level and below-zero habituation training on the spontaneous recovery and dishabituation of the orienting response. Physiological Psychology, 5, 257-260.

Content Aware Video Presentation on High-Resolution Displays

Clifton Forlines

Mitsubishi Electric Research Labs
Cambridge, MA 02139 USA
forlines@merl.com

ABSTRACT
We describe a prototype video presentation system that presents a video in a manner consistent with the video's content. Our prototype takes advantage of the physically large display and pixel space that current high-definition displays and multi-monitor systems offer by rendering the frames of the video into various regions of the display surface. The structure of the video informs the animation, size, and the position of these regions. Additionally, previously displayed frames are often allowed to remain on-screen and are filtered over time. Our prototype presents a video in a manner that not only preserves the continuity of the story, but also supports the structure of the video; thus, the content of the video is reflected in its presentation, arguably enhancing the viewing experience.

Author Keywords
Video playback, digital video, entertainment technology.

ACM Classification Keywords
H5.1. Information interfaces and presentation (e.g., HCI): Multimedia Information Systems - *video*.

1. INTRODUCTION
Despite the large number of hours that a typical person spends watching television and videos each year, little research exists within the CHI literature on improving and understanding video consumption, with some notable exceptions [1][2]. In recent years, personal video recorders, peer-to-peer file sharing, and portable video devices have begun to change the way that consumers interact with digital video. While televisions, projectors, and computer monitors have become physically larger and capable of displaying an increased number of pixels, the manner in which videos are displayed on these surfaces has remained the same. When creating new content for these devices, creators can choose to take advantage of these high-resolution displays; however, videos originally produced for smaller displays are simply scaled up to fill larger displays. Little is done to take advantage of a large display surface or a multi-display device. For example, a high-definition computer monitor, with a resolution of 1600 x 1200 pixels, displays a standard definition television signal, with a resolution of 640 x 480 pixels, by simply scaling the low-resolution video to fill the high-resolution display.

AVI'08, 28-30 May , 2008, Napoli, Italy

Typically, each frame of a video is displayed in place of and covers the entirety of the previous frame. One assumption that conventional video players make is that they should never display more than one frame from the same video at any one time. A similar assumption is that they never display the same frame from a video in multiple locations on the screen during playback. Finally, they never move the presented content around the large display space.

Our proof of concept prototype is an example of *Content Aware Video Presentation*. It converts an input video to an output video with the aim of challenging the above assumptions about video playback for the purpose of improving the experience. We take advantage of the increased pixel and physical size of large displays that modern computers and high-definition televisions have to offer. The input video can be thought of as a series of frames that are normally displayed sequentially. The output video is the same series of frames that have been scaled, rotated, filtered, and displayed in parallel on different regions of the display(s) in a manner that not only preserves the continuity of the story, but also supports the structure of the video.

The manner in which the frames are selected, the length of the frames, and the treatment of previously displayed frames are based on the structure of the input video. We determine the structure by using a variety of known techniques from the fields of video and image processing in conjunction with a new method for scene detection to find the relationship between shots, the content of individual shots, and camera motion. By displaying the frames of a video in this manner, the context of the video is reflected in its presentation, and the viewing experience is arguably enhanced.

We look toward other media, such as music and visual arts, which have both accepted the presentation of another's work in alternative forms, as a justification for our prototype. The techniques described in this paper are needed to explore this new style of video presentation and to determine if such alternative video presentation methods are desirable for high-resolution displays and non-traditional consumption of video content.

Figure 1. A frame from a Content Aware Video. The current shot is displayed in the foreground while the final frame from two previous shots remains in portions of the background.

2. RELATED WORK

Several attempts at improving the consumer's viewing experience through understanding the characteristics of the video have been explored.

Boreczky et al. [1] presented a technique for summarizing video that extracted keyframes from the video and scaled these images according to their importance. The differently-sized images were then packed together in a comic-book like layout. Viewers were presented with a graphical overview of the video and could navigate to an interesting part by clicking on any keyframe in the layout. While participants in a study were not able to find specific parts in a video faster using this layout than when using other summarization methods, participants did express a preference for the comic-book like technique.

Philips recently introduced "Ambient Light Technology" (Ambilight) for televisions. Ambilight illuminates the wall behind the television with backlight, and adjusts the brightness and color of this light based on the qualities of the frame currently being displayed on the television. Philips claims that this backlighting aids the visual perception system and enables the human eye to perceive more picture detail, contrast, and color. By filling the periphery of the viewer's vision with content, the designers of Ambilight hope to create a more immersive viewing experience.

Mitsubishi Electric recently released a DVD Recorder [9] that provides a "highlight playback" feature for sporting events. Highlights are extracted from the video during recording by analyzing the audio channel and looking for a characteristic mixture of cheering and the commentator's excited speech. Each second of the program is assigned an importance level, and the interface enables the user to set an importance threshold so only the portions of the program that exceed the threshold are played. The length of the summary corresponding to the choice of threshold is displayed, and the user can choose a desired summary length by moving the threshold up or down as needed.

Whittenburg et al. [15] presented an interface that used rapid serial visual presentation of the individual frames from a recorded program to support fast-forwarding and rewinding through video. Using this technique, the frames from the video are presented in a 3D trail leading away from the viewer, and upcoming shot changes are clearly visible when looking at the trail. By seeing the location of these changes and some of the details from upcoming frames, a viewer is better able to rapidly traverse to a desired location in the video.

Shamma et al describe an interesting use of the closed-captioning included in a television broadcast [11]. In their multi-display environment, while a video is being displayed on the main monitor, a background process is decoding the closed-captioning stream from the input video and using the words that the viewer is listening to as query terms for image searching. The results of these searches are displayed on the surrounding monitors, and the viewer is thus presented with a carousel of auxiliary material related to the video. These images provide context for the main program.

Fan et al [5] describe their approach to viewing video clips on limited resolution small screen devices. In one sense, they are addressing the opposite problem that we are. They detect the saliency of different objects in a frame of the video by measuring the local contrast. Once the interesting objects are identified, the player can zoom into that region of the video, allowing for an optimal use of the limited display space.

Many steps in our technique rely upon related work in video analysis, image comparison, and camera motion reconstruction. We will describe this work in the following sections as we describe the steps in our technique.

3. SYSTEM OVERVIEW

Figure 2 shows a high-level overview of our content aware video presentation prototype. The input to this system is a video and the output is a converted video. The output video has a resolution and aspect ratio that fills the entire high-resolution display space of the computer monitor(s) or television on which the video is to be viewed.

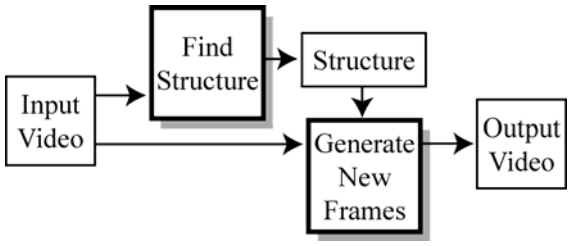


Figure 2. System overview. First, the structure of the video is found. Second, a new video is rendered from the frames of the original video and the structure.

This conversion has two high-level stages. In the first stage, we analyze the video to determine its structure (shot boundaries, related shots, scenes, camera motion, etc.). In the second stage, we use this structure to render a new output frame for each input frame in the original video.

For the purpose of describing our technique, we will commonly refer to a video that includes a scene in which two people are talking to one another (a very common scene in videos). Conventionally, shots in this type of scene alternate between close-ups of the two individuals with occasional overview shots showing both actors. The scene may begin with a foundation shot showing the location in which the conversation between these two characters is taking place. Our prototype converts this alternating sequence of shots into a video in which both actors remain on screen for the entirety of the scene.

This two-person conversation is just one of the many scene structures that occur repeatedly across different videos, and a detailed description of the many other common structures one observes across videos is outside the scope of this paper. We hope that the reader will see how the specific instances described in this paper can generalize to many types of scenes with many different patterns of shots.

4. STAGE 1 – VIDEO STRUCTURE

Video is often described as having a four tier hierarchical structure, as shown in Figure 3. A video is composed of one or more scenes, each of which includes one or more 'shots', each of which includes one or more frames. A 'shot' is a sequence of frames taken by a single camera over a continuous period of time. The shots are separated by shot boundaries.

Figure 4 shows an overview of how we determine the structure of the input video. We use a variety of previously known techniques in our system from the field of video analysis to determine this structure. The video is first segmented into shots by detecting shot boundaries. Each shot is compared to previous shots in the video to detect sets of visually similar shots, for example a series of shots of the same person or object. Similar shots are combined to form shot 'chains'. Chains that overlap in time are combined to

create scenes. In a side step, camera motion, which is present in many but not all shots, is estimated from the motion vectors of the video. In this section, we will describe each of these steps of Stage 1 in detail.

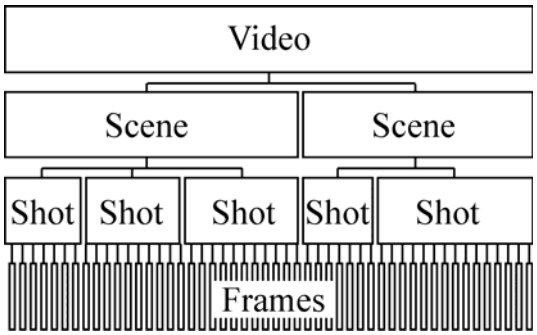


Figure 3. The four tier hierarchical structure of video. The entire video (top) is composed of a series of non-overlapping scenes, each of which is a series of camera shots, which are each composed of a series of individual frames.

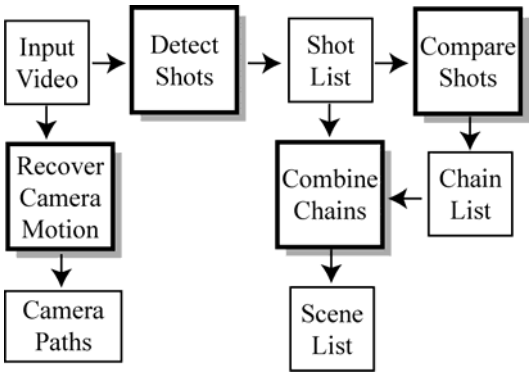


Figure 4. Overview of Stage 1, in which the structure of the video is detected. The shots are detected in the input video, and then these shots are compared to one another to find chains of visually similar shots, which are then combined into scenes. Separately, the camera motion is recovered for each frame of the video.

4.1 Shot Detection

A shot is defined as a continuous series of frames captured by one camera in a single continuous action in time and space. A number of processes are known for segmenting videos into shots by detecting shot boundaries. The methods can be based on color-histogram comparison, pixel differences, encoded macroblocks, and changes in detected edges between consecutive frames.

Lienhart [7] provides an excellent overview and comparison of several shot boundary detection techniques. Cabedo et. al [3] compared several shot detection techniques based on color-histogram comparison. These techniques are similar to one another in that they all create a histogram for two consecutive frames in the video, and then compare these histograms for dissimilarity.

Another promising new method for detecting shot boundaries in a video is presented by Cernekova et. al [4]. Their technique uses the joint entropy between frames to detect cuts, fade-ins and fade-outs. They presented an experiment in which their technique more accurately differentiate fades from cuts, pans, object or camera motion and other types of video scene transitions than previously known techniques.

All of these processes are similar in that they compare adjacent frames to detect when there is a significant difference between the frames that is indicative of a shot boundary. Any technique, or combination of techniques, that produces a list of shots from an input video is compatible with our system.

Our prototype system uses a modified color-histogram comparison algorithm. We first construct a color histogram for each frame of the input video. Each histogram has 256 bins for each RGB color component. We compare the histograms of adjacent frames as follows.

For each of the three color components, we sum the absolute differences between the values for each corresponding pair of bins giving us total differences for red, green, and blue between two frames. Each of the three total differences is compared with the average difference for the respective color for the previous five pairs of frames. If the difference for any of the three colors is greater than a predetermined threshold value times the average difference for that color, then a shot boundary is detected. To handle errors in an encoded video, shots that include fewer than five frames are combined with the following shot. The input of this step is the frames of the input video; the output is a list of shots.

It is worth repeating that our method of shot detection is not presented as a novel contribution to the field, but rather as a means toward an end. Any technique that produces a list of shot boundaries within an input video is compatible with our prototype.

4.2 Scene Detection

While a list of shots is a good first step in understanding the structure of a video, this list does not provide enough understanding for content aware playback. It is as if we have divided all of the words in a book into paragraphs, but have not yet divided these paragraphs into chapters. Further analysis is needed.

A scene, as in our example scene of two characters talking, is typically a contiguous sequence of shots that are logically related according to their content. Scene detection within videos is an active area of research. Yeung et al [16] introduced not only a pioneering piece of scene segmentation work, but also a means to visualize a video's structure. Zhao, et al [17] present an overview of the two major approaches for grouping shots together into scenes. The first approach looks at the boundary of shots and labels a shot boundary as a scene boundary if the visual and aural content change simultaneously. Lu, et al [8] present a scene detection technique that measures the continuity of visual, aural, and textual (closed captioning) elements in a video, and labels a shot boundary as a scene boundary when these continuities drop. The second approach compares the similarity between two shots by looking at the similarity of the shots' frames as a whole. Variations of this approach use different frame similarity measurements similar to the frame comparison metrics described in the previous section.

While many methods for scene detection exist in the literature, the uses of scene detection are less varied. For the most part, the output of a scene detection algorithm is used for indexing and summarization. Because our technique uses scene structure for playback, our needs are different. We need to not only gain an understanding of scene segmentation, but also an understanding of the relationships among the shots within a scene. We need an understanding of the chains of related shots that exist within a single scene.

Our prototype uses a two step approach to scene detection. In the first step, we find chains of related shots within the video. In the second step, we combine these chains into scenes.

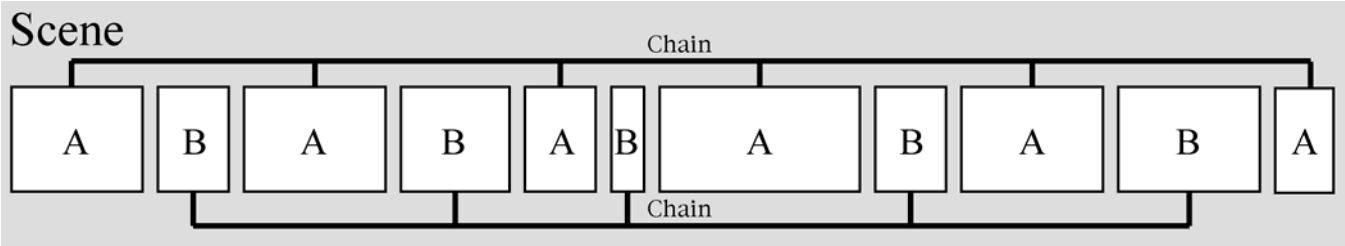


Figure 5. This figure shows a series of shots that have been grouped together into two overlapping chains of shots. The width of a shot is indicative of its length. One chain is all of the similar shots of one character talking, and the other chain is all of the shots of another character talking. Because these chains overlap in time, they are grouped together into a scene for the output video.

4.2.1 Step 1 - Finding "Chains" of Related Shots

For comparing the similarity of shots, our prototype again uses color histograms. We compare the first frame in a current shot with the last five frames of each of the previous five shots in the manner described in the 'Shot Detection' section of this paper, only using a more relaxed threshold. If a shot begins with a frame that is visually similar to the last five frames of a previous shot, then the shots are likely to be of the same person or object. A chain of shots is created whenever two or more shots are found to be visually similar. Chains can include many shots, and the similar shots in a chain do not need to be contiguous in time.

Any technique or combination of techniques that produce a chain of visually similar shots that are located relatively close together in time is compatible with our technique.

4.2.2 Step 2 - Combining Chains into Scenes

Figure 5 shows a series of shots in a video, which have been grouped into chains as described in the previous section. In this example, there are two chains, A and B. One chain is all the similar shots of one character talking, and the other chain is of all of the similar shots of another character talking. Because these chains overlap in time, we group them together into a scene for the output video.

Figure 6 shows a more complex example. In this figure, we see a series of shots, containing six chains, two and four of which overlap into two scenes.

Of course, not every shot is part of a chain, and we refer to these shots as orphans. Orphans that lie between the first and last shot of a scene and are not included in a chain are added to that scene (Figure 6, *left*). This shot is visually unrelated to its close neighbors, and is often an overview shot of the area surrounding the action taking place or a shot of both the subject of chain A and B. Orphans that are surrounded on either side with a scene are added to the trailing scene (Figure 6, *right*). In our experience, this type of orphan shot is almost always a foundation shot, in which the director tells the audience where the scene is taking place. In this case, the orphan may be a shot of the outside of the building in which the conversation between A and B is about to occur.

4.2.3 Handling Errors in Scene Detection

It is worth noting that errors in scene segmentation are less problematic for our task than for traditional scene segmentation tasks such as indexing and summarization. A summary that cuts off the last shot of a scene will leave the viewer wondering about the resolution of the story; on the other hand, because our technique is intended for playback, the result of a misplaced scene boundary is simply a less-than-ideal layout of the shots within the scene; in fact, when testing our prototype with users, such segmentation errors often passed unnoticed as viewers became engrossed in the story. Our method for scene detection is motivated by our use of scene detection and our need for the chaining of similar shots within a scene. A comparison between the performance of our method of scene detection and other methods is outside of the scope of this paper.

4.3 Estimating Camera Motion

Estimating camera motion with video analysis is another active research area. Videos encoded according the MPEG standard include motion vectors in B-frames and P-frames, and a number of techniques are known for estimating camera motion from the motion vectors.

Jones et al [6] describe a technique for stitching together the frames from a video into a single mosaic image. In the appendix of this work, the authors detail how they reconstruct camera panning and zooming through calculating the average of all motion vectors from the macro blocks within each frame of an MPEG video. Pilu describes a camera motion reconstruction technique [10] in which he first weights the motion vectors of each macro block by their reliability in predicting motion and then fits the filtered motion vector field to common velocity fields for common camera movements. Both of these techniques are appealing in that they

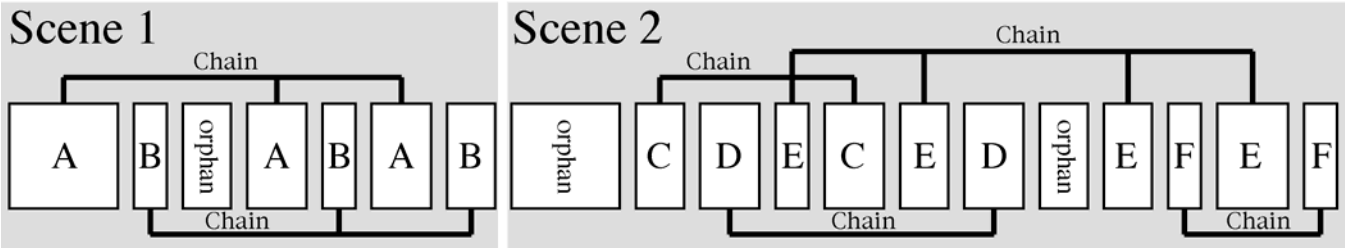


Figure 6. This figure shows the structure of two scenes. The scene on the left contains two overlapping chains of shots. The orphan shot in the middle of these two chains is added to the scene. The scene on the right contains four overlapping chains. The orphan shot in the middle of these chains is added to the scene, and the orphan shot that lies in-between these two scenes (which is most likely a foundation shot) is added to the scene on the right.

effectively piggyback on an already occurring process (the presence of motion vectors for the purpose of video compression) to provide a computationally inexpensive means of reconstructing camera motion. Other techniques for estimating camera movement include feature based tracking [10] and optical flow [12].

Our prototype parses motion vector data directly from the input video, which is encoded according to the MPEG-2 standard. For each frame in a shot, the variance for the X-Y motion is determined for all of the motion vectors. If the variance is below a predetermined threshold, then the average motion for all motion vectors is recorded. In other words, if the most of the motion vectors for a single frame are all more or less pointing in the same direction, then we assume that the camera is moving in the opposite direction and we record the motion. If the variance is above the threshold, then we record a vector of length zero. Currently, our prototype only handles camera panning, but others have reconstructed zooming and rotation and the detection of these types of camera movement are left for future work.

In this manner, we produce an average motion vectors for each frame in the video. These camera paths are used when we render the converted video in the second stage.

5. STAGE 2 – RENDERING NEW FRAMES

Figure 7 shows an overview of Stage 2, in which our technique generates the new frames of the output video. The input for this stage is the original input video and the chains, scenes, and camera paths from Stage 1. In this second stage, for every scene in the input video, a new scene of equal length is rendered. Finally, these scenes are combined along with the audio tracks from the input video into the output video.

5.1 Templates for Frame Layout

For each scene in the list of scenes, we compare the structure of that scene to predetermined templates in order to select a most appropriate rendering for the frames of that scene. By structure, we mean the number and pattern of chains in the scene, the presence of shots in the scene not included in a chain, the length of the chains, and the amount of overlap of the chains of a scene. The templates are ranked based on how closely the characteristics of the scene match an ideal scene represented by the template. Our prototype then uses the template that most closely matches the scene to render a new image for each frame in that scene.

As shown in Figure 8, each template initially generates a blank image that is the size and has the aspect ratio of the high-resolution display on which the output video will be viewed. Then, the first frame from the input video is rendered into a region of the blank image, perhaps filling the entire image. This image is then saved as the first frame of the new scene in the converted video. While there are frames remaining in the input scene, the next frame from the input video is rendered into a new region of the image. The region that this next frame is drawn into may or may not overlap the previous region, and the previous image may or may not be cleared of content.

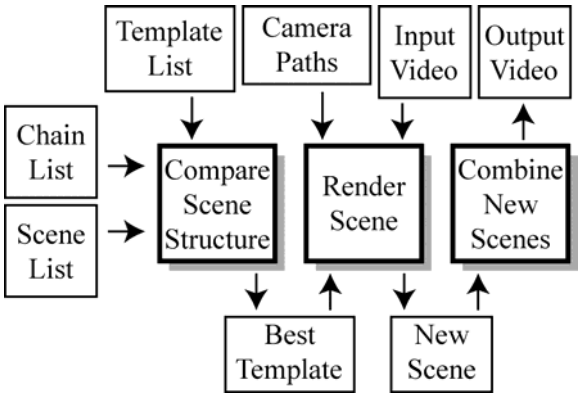


Figure 7. Overview of Stage 2, in which a new frame is rendered for each frame of the input video. For each scene in the input video, the structure of that scene is compared to a list of templates. The best matching template then renders a converted scene using the frames from the original video, and optionally the recovered camera motion. Finally, these scenes are combined into the output video.

As shown in Figure 9, the example scene includes two characters talking to one another. In the original video, the shots alternate sequentially between the two characters as they speak, with no one shot showing both people. The template for rendering this scene renders each frame from the first chain into a region on the left side of the screen, and each frame from the second chain into a region on the right side of the screen.

The result is a sequence of images in which the talking characters appear on the left and right side of the images. During playback, a viewer of this sequence of images alternately sees the actively talking character in either the left or the right region, and the non-speaking character displayed as a still frame in the other region. The still frame corresponds to the last frame of the shot in which that character is talking.

Some templates filter the previous frame from the output video before drawing the current frame. In a variation of the two-person conversation example, the still frame on the right can slowly fade to black while the active shots on the left continues, until the still frame on the right becomes an active shot again, and the left region shows a slowly fading still frame.

Figure 9. The top row shows the first frame from five consecutive shots in the original input video. The bottom row shows five rendered frames from the output video. The frames from the five shots above are painted into either the left or the right region of the screen. The final frame of the previous shot remains frozen on screen on the opposite side.

Figure 10. The top row shows four frames from the first shot in this scene followed by the first frame from the second shot in this scene. The dotted line indicates the shot boundary between these two shots. The bottom row shows the animating region of the screen that the template rendered the original frames into. The effect presented in the converted output video is that the shot begins playing back full screen, and then slowly animates to fill only the left region of the screen. The second shot is then displayed in the right region of the screen, and the final frame from the first shot remains frozen on the left.

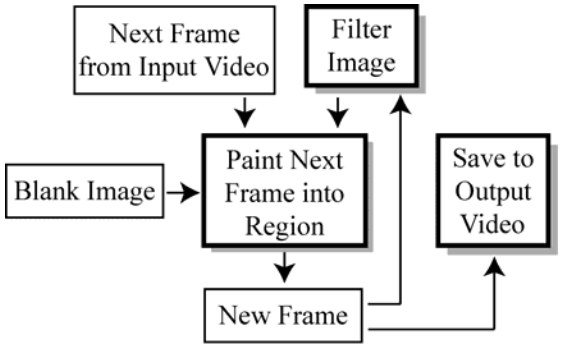


Figure 8. Templates recursively paint frames from the input video into regions of the output video. The background of these new frames is the previous frame from the output video, which may be filtered.

In addition to a simple fade, any number of conventional image filtering techniques can be used. Still frames can reduce their color saturation over time, i.e., change into a black-and-white image, or can be blurred, pixilated, or converted to a sepia tone.

Some templates are designed to animate regions of the output images into which frames from the input video are rendered. Figure 10 (*bottom row*) shows five consecutive output images generated by the template. The template used to render this scene renders each frame from this shot into an animating region. Note that the regions vary in size and location to give the effect of animation. In addition to varying the size and location of the region over time, templates could distort, rotate, and/or reflect the original video frames.

As shown in Figure 11, a template can animate the region into which frames are painted according to the stored camera motion described in the previous section of this paper. In this example, the camera pans from left to right across the scene to reveal a boat that is originally off-camera, right. Therefore, the region into which frames from the input video are rendered moves across the screen, animating according to the camera path. The reader will recall that each frame of the shot has a 2D vector associated with its movement, in pixel units. Each frame in the shot is translated by the summation of all the vectors up to an including that frame, and then all of these translated frames are combined into a single image. A scale factor is then computed by examining the ratio between the size of the composite image and the size of the output video. This scale factor is then applied to the 2D vectors and used to resize the input frames as they are rendered into an animating region of the output video. In this way, as much of the area of the output video is used as possible.

5.2 Creating the Output Video

After a template is matched to each scene in the input video and a new output frame is rendered for each frame in the input video, the rendered images are arranged sequentially and encoded according to the MPEG-2 standard to produce the output video. Our prototype then copies the unchanged audio track from the input video. The output video is now ready for playback on the high-resolution computer monitor, high-definition television, or multi-monitor system using a conventional video playback device.

6. LAYOUT DESIGN GUIDELINES

In building example templates, we have come up with several guidelines to follow when designing new templates for layout.

6.1.1 Time is constant

The first constraint on template design is that the input and output scenes must be equal in length. Cutting pieces out of the original video may drastically affect the story. Speeding up or slowing down portions of a video may be desirable in some situations, but we did not see a clear mapping between scene structure and story pacing. Changing the speed of playback also makes the audio track less recognizable. One exception to this guideline that works

Figure 11. This row shows five frames from the output video. The original five frames were taken from a shot in which the camera panned across a large room to reveal a boat on the right. The template charged with rendering this scene used the recovered camera motion for this shot to inform the animation of the region into which these frames were painted. The effect in the converted output video is that the content of the shot (the large room) is remaining stationary and that the animating frame is providing a keyhole like view into the room. In the background, we see the final frame of the previous shot fading to black.

well in some situations is shot repetition. A template that recognizes an important shot may present it multiple times in succession, perhaps altering the size or scale of the frames.

6.1.2 Current frame is shown in entirety

An early template that we designed animated a shot from an off-screen location, which ended up hiding an important feature of the shot from the viewer. Similarly, another early template presented shots in such as way that they sometime appeared partially occluded by a previous shot shown in another location on the screen. These observations led us to the guideline that, while a template may scale or filter the current frame, the entirety of the current frame is always visible in some region of the screen.

6.1.3 Never show frames ahead of the current time

Many templates leave frames from previous shots on screen, often to create a background content for the currently displayed shot. When we experimented with showing frames from upcoming shots along with the current shot, we began to violate the cause-and-effect relationship between sequential shots. Showing effect before cause was extremely disorienting in many cases (although oddly intriguing in a few cases). This observation led us to the guideline that templates should never show frames from upcoming shots, only from previous ones or the current one.

7. Future Work

A better understanding of the variety of scene structures that occur in commercial programming is needed to generate a more complete list of templates. Our prototype was designed with the adding of templates in mind – and we plan on adding more templates to translate different types of scenes in a content appropriate manner. Analysis of the contents of the frames themselves can be useful in informing frame layout. In this section, we lay out a means by which the frames within a scene can aid in frame layout in the converted video.

7.1.1 Gaze Direction Detection

Layout templates could use a gaze direction detection process on the frames in each of the chains. A number of techniques are known for estimating gaze direction of faces in images [13]. Such a process would recognizes that the woman in Figure 7 is facing to the right and that the man in Figure 7 is facing to the left. The frames in the chains can then be combined so that the two characters appear to face one another.

The gaze direction of characters can also inform the system as to the "angle" of the shot. By "angle" we mean the relationship between the viewer and the characters, a relationship that tells us something about the intent of the content creators. A "high-angle" shot is one in which the camera is above the eye-level of the subject. The consequence of this point of view is that the character appears small and weak. A "low-angle" shot has the opposite effect. By pointing the camera up at the character, the character appears powerful and large. A template that could classify shots as high, neutral, or low-angle shots could use this information to present shots accordingly – growing low-angle shots to fill the screen, thus enhancing certain characters' power and presence and shrinking high-angle shots to amplify other characters' weakness.

7.1.2 Face Detection and Recognition

Robust face recognition would greatly aid in the finding of chains of related shots. Knowing that a specific character is present in nearby shots would suggest that the shots are part of the same scene. Unfortunately, robust face recognition is an open research area without established techniques that work well in various lighting conditions. Advancements in face recognition should complement our prototype as they arise.

While robustly recognizing specific faces is an unsolved problem, techniques exist that provide robust face detection [14]. These techniques do not provide information about who is present in the frame; however, they do provide information about the presence or absence of faces, as well as the number of faces and the relative size of these faces within the image. Knowing the number of faces in a shot could help in the layout of a scene. For example, a scene containing three chains of shots, two of which have one face and one of which has two faces probably contains a conversation between two people. The chains with one face are close-ups of the two people, and the chain with two faces is an overview shot of the two characters together. A template recognizing this pattern could render the close-up shots of the left and the right side of the screen, and render the overview shots in the middle.

Knowing size of faces within the frames of a shot could be very helpful in informing a template as to the "length" of the shot. By "length" we are not referring to the duration of a shot, but rather the length of camera lens, which relates to the depth of focus. A very small face would indicate a "long" shot, one in which a character is shown in relation to their surroundings. A face that fills are large portion of the screen would indicate a "short" or "close-up" shot. Since a close-up is used to show the physical details of the actor's face, and gain an understanding of how the character feels or to clarify an action, templates would want to render this type of shot into a large region to preserve this detail.

The ordering of shot lengths could also inform the layout. For example, a medium or long shot that is followed by a close-up is probably a two shot sequence meant to first show the context of a person, and then show the details. A template recognizing such a structure would want to render the medium or long shot into a full screen region, and then leave the final frame of this shot on screen as it renders the close-up into an overlapping region of the display.

7.1.3 File Format

A variation of our prototype could generate an XML file rather than a second video file. A modified player application would read from the original video file as well as this XML file, which would include the region on screen into which the current frame would be painted as well as descriptions of any filtering that should take place during each frame of the video. This variation would have the benefit of requiring much less disk space; however, playback would become a more computationally expensive operation.

7.1.4 Audio Structure

All of the structure that our prototype uses to inform the layout of frames comes from examining the video track of the original video. The audio track(s) and closed-captioning are copied unchanged to the converted video and are not used to inform the structure. Certainly, a more sophisticated version of our prototype would examine the audio tracks for content aware presentation.

8. EARLY REACTIONS

We cannot conclude a paper on a new method for video presentation without some discussion of the question of whether or not such a presentation is desirable. To begin answering this question, we have presented several videos generated by our prototype to many coworkers, colleagues, guests of our lab, television manufacturers, and members of the content creation industry. Reactions have been mixed, but almost everyone seems to have a strong opinion, either enthusiastic or uneasy.

The most common positive adjective has been “fun”. Several people mentioned that this type of presentation might be a way to enjoy previously-viewed programming in a new way. Several viewers have stated that viewing a video in a context aware manner makes video watching a more active experience as they follow the sequence of scenes around the screen. The television manufacturers that we spoke with recognized this increase level of activity, and expressed concern that their customers might not want to maintain a high-level of mental activity when watching videos. There was a concern that context aware video playback “could wear the viewer out” by “over engaging them.”

Not too surprisingly, the members of the content creation industry expressed uneasiness with the idea of presenting another person’s work in an alternative fashion. When questioned about their uneasiness, the most common source was the feeling that the time and effort put into the original by the director and cinematographer were being disregarded by altering the presentation of the video. In our defense, we look to other media and the means in which people derive art from other people’s art. With visual art, we see an analogy between our prototype and the art of collage. A collage draws upon a multitude of previous pieces and combines parts into something new. While the viewer may recognize the source of the pieces within a collage, he never confuses the pieces with the original. Similarly, musical pieces are often covered by other artists, and even sampled, filtered, looped, and remixed to create derivative work.

9. CONCLUSION

We have presented a prototype video presentation system that presents a video in a manner consistent with the video’s structure. By reconstructing scene structure through shot detection and shot comparison, we take advantage of the large display and pixel space that current high-definition computer monitors and televisions provide by displaying shots from the video in various locations on the display. By reflecting the content of the video in its presentation, we hope to add to the viewing experience.

REFERENCES

1. Boreczky, J., Girgensohn, A., Golovchinsky, G., and Uchihashi, S. 2000. An interactive comic book presentation for exploring video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM Press, New York, NY, 185-192.
2. Brown, B. and Barkhuus, L. 2006. The television will be revolutionized: effects of PVRs and filesharing on television watching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Canada, 2006). CHI '06. ACM Press, New York, NY, 663-666.
3. Cabedo, X.U. and Bhattacharjee, S.K., Shot detection tools in digital video. In *Proceedings of Nonlinear Model Based Image Analysis 1998* (Glasgow, July 1998). Springer Verlag, 121-126.
4. Cernekova, Z., Nikou, C., and Pitas, I. Shot Detection in Video Sequences Using Entropy-Based Metrics. International Conference on Image Processing 2002 (ICIP2002), Vol. 3, 421-424.
5. Fan, X., Xie, X., Zhou, H., and Ma, W. 2003. Looking into video frames on small displays. In *Proceedings of the Eleventh ACM international Conference on Multimedia* (Berkeley, CA, USA, November 02 - 08, 2003). MULTIMEDIA '03. ACM Press, New York, NY, 247-250.
6. Jones, R. C., DeMenthon, D., and Doermann, D. S. 1999. Building mosaics from video using MPEG motion vectors. In *Proceedings of the Seventh ACM international Conference on Multimedia (Part 2)* (Orlando, Florida, United States, October 30 - November 05, 1999). MULTIMEDIA '99. ACM Press, New York, NY, 29-32.
7. Lienhart, R. Comparison of Automatic Shot Boundary Detection Algorithms. In Image and Video Processing VII 1999, Proc. SPIE 3656-29, Jan. 1999.
8. Lu, X., Ma, Y.-F., Zhang, H.-J. and Wu, L., An Integrated Correlation Measure for Semantic Video Segmentation. In *Proceedings of IEEE International Conference on Multimedia and Expo – ICME ‘02*, (Lausanne, Switzerland, 2002), 57- 60.
9. Mitsubishi Electric. RakuReko DVD Recorder. http://www.mitsubishielectric.co.jp/news/2006/0601.htm
10. Pilu, M., On Using Raw MPEG Motion Vectors To Determine Global Camera Motion, Digital Media Department, HP Laboratories, August 1997.
11. Shamma, D. A., Owsley, S., Hammond, K. J., Bradshaw, S., and Budzik, J. 2004. Network arts: exposing cultural reality. In *Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers &Amp; Posters* (New York, NY, USA, May 19 - 21, 2004). WWW Alt. '04. ACM Press, New York, NY, 41-47.
12. Teodosio, L. and Bender, W. 1993. Salient video stills: content and context preserved. In *Proceedings of the First ACM international Conference on Multimedia* (Anaheim, California, United States, August 02 - 06, 1993). MULTIMEDIA '93. ACM Press, New York, NY, 39-46.
13. Varchmin, A. C., Rae, R., and Ritter, H. 1998. Image Based Recognition of Graze Direction Using Adaptive Methods. In *Proceedings of the international Gesture Workshop on Gesture and Sign Language in Human-Computer interaction* (September 17 - 19, 1997). I. Wachsmuth and M. Fröhlich, Eds. Lecture Notes In Computer Science, vol. 1371. Springer-Verlag, London, 245-257.
14. Viola, P. and Jones, M.J. Robust Real-Time Face Detection. International Journal of Computer Vision, 57 (2). 137-154.
15. Wittenburg, K., Forlines, C., Lanning, T., Esenther, A., Harada, S. and Miyachi, T., Rapid serial visual presentation techniques for consumer digital video devices. in *Proceedings of the 16th annual ACM symposium on User interface software and technology*, (Vancouver, Canada, 2003), ACM Press, 115-124.
16. Yeung, M. M. and Yeo, B. 1996. Time-Constrained Clustering for Segmentation of Video into Story Unites. In *Proceedings of the international Conference on Pattern Recognition (ICPR '96) Volume Iii-Volume 7276 - Volume 7276* (August 25 - 29, 1996). ICPR. IEEE Computer Society, Washington, DC, 375.
17. L. Zhao, W. Qi. Y.J. Wang, S.Q. Yang, and H.J. Zhang. Video Shot Grouping Using Best-First Model Merging. Proc. 13th SPIE symposium on Electronic Imaging - Storage and Retrieval for Image and Video Databases, SPIE vol. 4315, pp.262-269. Jan. 2001.

SparTag.us: A Low Cost Tagging System for Foraging of Web Content

Lichan Hong, Ed H. Chi, Raluca Budiu, Peter Pirolli, and Les Nelson
Palo Alto Research Center (PARC)
3333 Coyote Hill Road, Palo Alto, CA 94304, USA
{hong, echi, budiu, pirolli, lnelson}@parc.com

ABSTRACT
Tagging systems such as del.icio.us and Diigo have become important ways for users to organize information gathered from the Web. However, despite their popularity among early adopters, tagging still incurs a relatively high interaction cost for the general users. We introduce a new tagging system called SparTag.us, which uses an intuitive Click2Tag technique to provide in situ, low cost tagging of web content. SparTag.us also lets users highlight text snippets and automatically collects tagged or highlighted paragraphs into a system-created notebook, which can be later browsed and searched. We report several user studies aimed at evaluating Click2Tag and SparTag.us.

Categories and Subject Descriptors
H5.2 [**Information Interfaces and Presentation**]: User Interfaces – Graphical User Interfaces. H5.3 [**Group and Organization Interfaces**]: Collaborative Computing.

General Terms
Design, Human Factors.

Keywords
Tagging, highlighting, social bookmarking, Web 2.0, annotation.

1. INTRODUCTION
Vannevar Bush's vision of the Memex [1] has inspired the evolution of information systems that augment and enhance human abilities to find, store, organize, understand, retrieve, and share knowledge. The areas of information retrieval, personal information management, and the Web (to name just a few) have, for the most part, historically been focused on supporting *individual* information foraging and sensemaking. Recently there has been an efflorescence of systems aimed at supporting *social* information foraging and sensemaking. These include social tagging/bookmarking systems for photos (e.g., flickr.com), videos (e.g., youtube.com), or web pages (e.g., del.icio.us). Tagging systems provide a means for users to generate labeled links (*tags*) to content that, at a later time, can be browsed and searched. A unique aspect of tagging systems is the freedom that users have in choosing the vocabulary used to tag objects: any free-form keyword is allowed as a tag. Tags can be organized to provide meaningful navigation structures, and, consequently, can be viewed as an external representation of what the users learned from a page and of how they chose to organize that knowledge.

AVI'08, May 28–30, 2008, Napoli, Italy.

Empirical research [12] shows that people use tagging systems primarily for private, individual information storing and management. Other researchers [2,11,21] suggest that the success of social and collaborative systems is dependent on the architecture of interaction, as well as on the costs and benefits of interaction to the individual user. Grudin [11] discusses how costly interactions for the individual affect groupware adoption. Further, social tagging systems are instances of *networked information economies* [2], involving the production, distribution, and consumption of information by decentralized users operating over a network. Generally, as the costs of interaction are driven down, more users participate in the production and use of knowledge (e.g., tags). As more users participate, the value of the social tagging system to the participants is improved. Based on this reasoning, our research focuses on reducing the interaction costs of tagging.

In this paper, we describe a web content tagging system called SparTag.us. Our main objective is to provide low cost tagging and highlighting capabilities to users while they are reading web content. This paper has three main contributions:

(1) Presenting a web content tagging system called SparTag.us that supports in situ tagging and highlighting during reading;

(2) Introducing an intuitive *Click2Tag* technique to lower the overall costs of tagging; and

(3) Showing (through user studies) that Click2Tag provides lower tagging costs than typing.

In Section 2, we present the need for low cost tagging and introduce the Click2Tag technique that allows users to tag a web page by clicking on words of that page. To validate the intuition behind Click2Tag, in Section 3 we analyze the provenance of tags in del.icio.us. After presenting the design and implementation of SparTag.us in Section 4, we report several user studies in Section 5. These studies investigated the impact of Click2Tag and collected user feedback on the usability of SparTag.us, as well as on its potential uses at the *individual* level. The studies were part of an iterative design process intended to improve the SparTag.us interface. Finally, we finish with some concluding remarks and future work.

2. MOTIVATION

The social bookmarking space has become quite crowded in recent years. Popular bookmarking systems include del.icio.us, MyWeb (myweb.yahoo.com), Diigo (diigo.com), Clipmarks (clipmarks.com), Magnolia (ma.gnolia.com), Bluedot (bluedot.us), and Google Notebook (google.com/notebook). Most of these systems allow tagging at either the page level (i.e., URL) or the sub-page level (e.g., individual paragraphs). A few of them, including Diigo, Clipmarks, and Google Notebook, also support other forms of annotation (e.g., highlighting).

To understand the costs of tagging, for each of these systems, we performed a GOMS-like analysis [6] of the interface and identified the overall number of steps involved in tagging. We count these steps to get a gross measure of the tagging costs, as shown in Table 1. Due to space constraints, we cannot discuss most of this analysis in detail. So, in the following, we exemplify this cost analysis for Diigo, since it offers features closest to what we had in mind.

Table 1. Tagging costs of social bookmarking systems.

System	*Cost*
del.icio.us	6
MyWeb	7
Diigo	8
Clipmarks	10
Magnolia	6
Bluedot	6
Google Notebook	11

To annotate a paragraph with Diigo, a user has to go through a sequence of steps as shown in Figure 1a. We identify two types of cost involved in this process: (1) *interaction cost* (mouse clicks, button presses, typing) in steps 1, 2, 3, 6, and 7; and (2) *attention-switching cost* (moving attention from one window to another) in steps 4 and 8, and possibly 5 (if the user has to go back to the original page to decide what tags to use). These costs reflect the interruption of reading and the switch to a different interface.

In SparTag.us, we aim to reduce these two types of cost by integrating tagging into the flow of reading. Specifically, with the Click2Tag technique we make two types of sub-page objects live and clickable: paragraphs of the web page and words of the paragraphs. According to our analysis of del.icio.us data (see Section 3), a considerable portion of tags comes directly from the web pages. When people tag a web page in del.icio.us, there is about 50% chance that the tag word appeared in the content of the page. Making each word of the paragraphs clickable allows users to simply click on a word to add it to the tag list without typing, thus lowering the interaction costs of tag typing. This input method can be especially useful in cases where keyboard input is not the primary input means (e.g., iPhones, tablet readers).

We also mitigate the attention-switching cost by enabling in situ tagging. With the Click2Tag interface, users can directly click on any paragraph to start tagging while they are reading it. When tags are entered, they are inserted at the end of the paragraph and displayed within the same rendered web page. This design was inspired by annotation studies [14,19] and by in-document collaboration systems [8], in which the content, and not some external form, is used as the setting for information sharing. Thus,

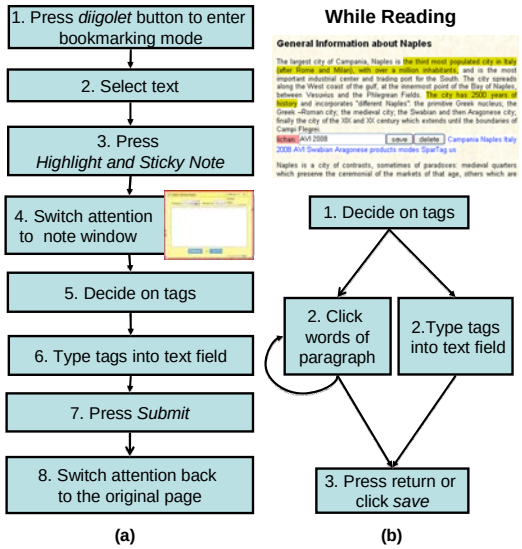


Figure 1. Cost analysis in (a) Diigo and (b) SparTag.us.

Click2Tag combines synergistically with in situ tagging, ensuring that the focus of attention remains on the content at hand.

Figure 1b shows a cost analysis of tagging that we strive to achieve in SparTag.us. Because users tag as they read, they do not need to shift attention to a dedicated tagging window, thus eliminating the attention-switching costs discussed earlier. We also eliminate the switching of attention back to the web page being read after the tags are applied. The attention-switching cost is reduced for those relying on consulting the document to decide which tags to apply (users get ideas for tags faster because the text is in front of their eyes). The interaction cost is also diminished: users can select tags from text instead of typing them. Moreover, compared to Diigo, we eliminated the initial button press since there is no need to invoke the tagging interface explicitly.

Studies have shown the need for richer forms of annotations due to the wide variety of annotation practices employed by readers [14,19]. Particularly for personal use and re-reading, Marshall and Brush [15] recognized the need for various forms of highlighting such as sentence highlighting, underlining, circles and margin bars. Many popular social tagging systems such as del.icio.us do not currently support other forms of annotations beyond tagging. In addition to tagging, SparTag.us also supports text highlighting. Indeed, our studies show that the combination of tagging and highlighting was perceived by users to be more valuable than either tagging or highlighting alone (see Section 5).

3. USING CONTENT WORDS AS TAGS

To further motivate the design of Click2Tag, we now present an analysis of how often tag keywords appear in the page content. Tagging is a process that associates keywords with specific content. This raises a question regarding the appropriateness of a content-driven approach such as Click2Tag: How many of the tags that people use actually come from tagged content? To help assess how Click2Tag relates to current practices employed in social bookmarking systems, we crawled del.icio.us and computed how often a keyword used by a user to tag an URL appears in the page content. We did this by the method described as follows.

First, we sampled tag activity data from the 29,978 most frequently bookmarked URLs up through June 2006. For each

URL, the body content was retrieved at the time of analysis (December 2007) if available and pruned to its textual content (i.e., the HTML tags were removed). Of the 29,979 URLs, we excluded 1,254 that had become invalid at the time of analysis. This left us with 28,724 valid URLs.

Next, for each URL, we converted its tag activity data into a set of tuples of the form (user, URL, tag). Each tuple corresponds to a single tagging event representing that a user bookmarked a specific URL using a particular tag. The contribution of a tuple is set to be 1 if the tag matches any word in the text body of the URL; otherwise, it is set to be 0. We computed the probability that a user tagged a page with a content word by first summing up the contributions of all the tuples for that specific URL and then dividing the sum by the number of tuples.

That is, for each URL, we collected all of the tuples involving that URL to give us a count of the total number of tuples for the URL. From this tuple collection, we then counted the number of tuples with content words as tags. Dividing the second count with the first count will give us the probability that a user tagged a page with a content word for that particular URL:

$$P(URL) = \frac{\#tuples - with - content - tags - for - URL}{\#tuples - for - URL}$$

By computing this probability for all of the 28,724 valid URLs, we get the distribution as shown in Figure 2 (see also the Color Page). Table 2 summarizes the data collection. On average, the chance that a tag comes from the content is 49%. This process produced a *conservative* estimate of tag occurrence in content, since we did not account for situations such as content changes for a given URL (e.g., dynamic content), typos (e.g., "Ajaz" instead of "Ajax"), abbreviations (e.g., "ad" instead of "advertisement"), compound tags (e.g., "SearchEngine"), and tags written in languages other than that of the content.

Table 2. Summary statistics of del.icio.us data analysis.

# of URLs sampled	29,978
# of invalid URLs	1,254
# of valid URLs (searched for tags)	28,724
Probability that a user tagged an URL with a word occurring in page content:	
Mean	0.49
Standard deviation	0.26

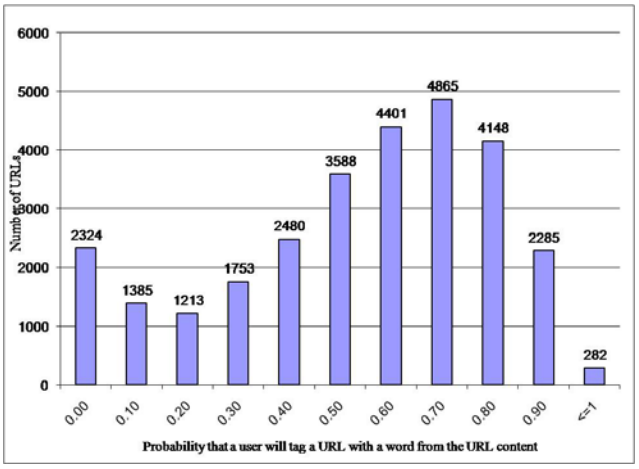


Figure 2. In del.icio.us, probability of a user tagged an URL with a word occurring in the page content of the URL.

To some extent, we were surprised by the relatively high probability of tag occurrence in del.icio.us tagged content. Clearly, people bookmark and tag pages for a wide variety of reasons [9]. While our analysis is not exact, it is reasonable to assume that when a word is used as a tag for a URL, there is a decent chance it derives from the content itself.

4. DESIGN OF SPARTAG.US

Having motivated the design of SparTag.us, we now describe its various functionalities and implementation in detail. In what follows, we use the word "annotation" to refer to both tagging and highlighting.

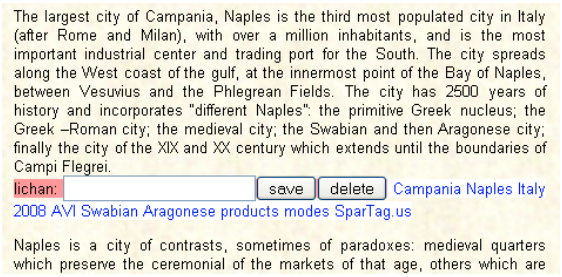
4.1 Lowering the Costs of Annotation

As mentioned earlier, in SparTag.us, we bring the tagging capability into the same browser window displaying the web page being read. When a user loads a web page in his browser, we augment the HTML page with AJAX code to make the paragraphs of the web pages as well as the words of the paragraphs live and clickable. Specifically, we identify the paragraphs by their HTML tags (e.g., elements enclosed within <*p*> or <*h2*>, etc). In addition, we alter the Document Object Model (DOM) tree of the page by enclosing each word of those paragraphs with the HTML tag <*span*>. Furthermore, we attach mouse event listeners to the document object.

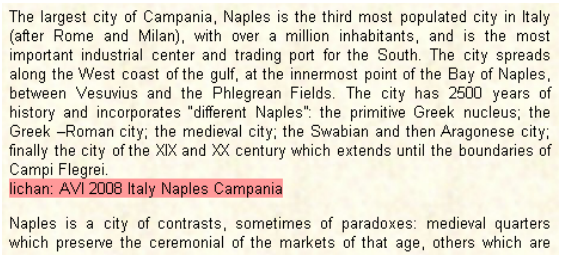
Our AJAX augmentation essentially converts each paragraph into a taggable object. As users read a paragraph, they can simply click anywhere on the paragraph to tag it. Consequently, a tagging widget is dynamically inserted at the end of the paragraph, as shown in the top part of Figure 3a (see also the Color Page). At this point, the user can input any tags into the text field of the widget. Clicking the *save* button or pressing the return key completes the tagging operation, with the tags displayed at the end of the paragraph (see Figure 3b). This Click2Tag interface provides a low cost way to invoke the tagging functionality while reading.

To add tags to the text field of the tagging widget, the user may simply click on words of the paragraph. Alternatively, the user can click on one of his most recently used tags (displayed in blue after the *delete* button in Figure 3a) to add it to the tag list. Of course, one can still type in the text field to add or modify tags, in case a desired tag does not appear in the paragraph. Recognizing the debate concerning single-word tags versus compound multi-word tags [13], we currently support single-word tags in SparTag.us, as in del.icio.us. To add compound tags in SparTag.us, the user needs to modify the existing tags in the text field or type directly.

In our augmentation, extra care is taken to handle special cases such as punctuations and hyperlinks. Punctuations are automatically filtered out. And by default, when a hyperlink is clicked, we let the browser follow the hyperlink to a new page. We use a word of the hyperlink as a tag only when the word is clicked with the control key held down.



(a)



(b)

Figure 3. (a) Clicking on the first paragraph inserts a tagging widget to the end of the paragraph. (b) Tags are displayed as part of the first paragraph when tagging is finished.

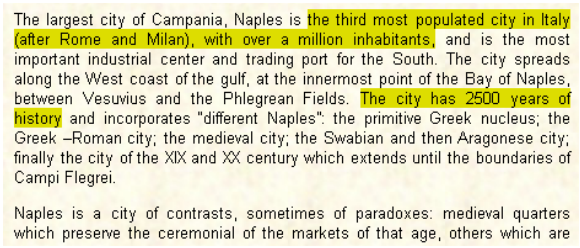


Figure 4. Selecting a text snippet highlights it in yellow.

We can support highlighting fairly easily as well, since each word of the paragraphs is augmented with interaction handlers. To highlight a piece of text in yellow as shown in Figure 4 (see also the Color Page), the user first points the mouse at the beginning of the snippet. While pressing and holding the left button, he moves the mouse to the end of the desired text region and releases the button. This sequence of mouse movements is exactly the same as what it takes to select a piece of text in the browser, an operation that the user is already familiar with. Indeed, in our implementation we overload the text selection functionality of the browser to support text highlighting. We chose this implementation after several iterations of design, as discussed in Section 5.2. It also to some extent reflects the challenge of our AJAX augmentation.

At the end of the highlighting operation, we capture the highlighting region automatically and send the corresponding data to our web server asynchronously. In other words, the user does not have to press any *save* button to record the highlight. To allow the user to erase previous highlights and perform the default text selection operation, we create separate modes for highlighting, highlight erasing, and text selection, respectively.

Although in SparTag.us we mainly focus on paragraph annotation, it is conceivable that Click2Tag may be applied to create tags at the page level as well. For example, a set of words selected by a sequence of clicks can be used as a basis to compute tags for the entire page. Compared to page tagging, paragraph annotation offers a better solution to handle pages that are fairly lengthy and yet only contain a few paragraphs of interest to the user. In addition, paragraph annotation serves as a visual indicator to remind the user what he feels is important on the page, which could be quite useful when the user revisits the page later on.

4.2 Notebook of Annotated Content

An important consideration in designing information foraging systems is to make it easy to store and retrieve the nuggets of information that have been collected by users [24,26]. Inspired by XLibris [23], we create a SparTag.us notebook for each user that automatically captures the results of the user's annotation activities. After the user annotates a paragraph on a web page, the paragraph is automatically extracted from the page and inserted into his notebook.

Figure 5 (see also the Color Page) shows the top portion of the notebook for user *lichan*. On the left side of the notebook, the annotated paragraphs are listed in reverse chronological order. For each paragraph, we show the time when the paragraph was last annotated. The tags and highlights created by the user appear in conjunction with the text of the paragraph. If needed, the user can directly modify his annotation on the paragraph here. At the end of the paragraph, we also include the URLs of pages that the user visited and contain the same paragraph. On the right side of the notebook, we show a tag cloud of the user's tagging keywords.

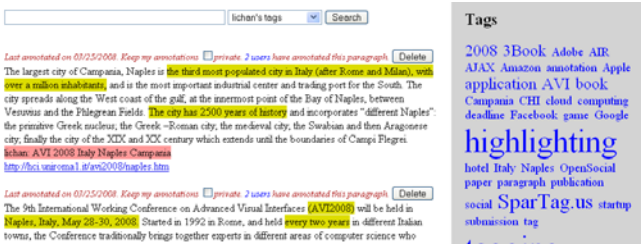


Figure 5. The top portion of the reading notebook that SparTag.us created for user *lichan*.

The search box at the upper-left corner of the notebook provides several search options. The user can search against the paragraph tags, the highlighted snippets, the entire paragraph texts, or the URLs. This makes it easy to retrieve a subset of paragraphs that are related to a certain concept or from a specific web site.

4.3 Architecture

In our implementation, SparTag.us consists of two parts: a client-side browser extension and a server. The client side is currently implemented as a Firefox extension, which includes a browser toolbar. The toolbar, as shown in Figure 6, consists of several shortcuts to key features of SparTag.us, e.g., turning on/off SparTag.us, changing modes, and accessing the notebook. We are currently exploring client extensions for other browsers as well, including IE and Safari.

The client side extension also contains a GreaseMonkey [10] script. GreaseMonkey is a client side tool that inserts our AJAX code into web pages while allowing the user's browser to communicate with multiple web servers. Figure 7 shows an architecture diagram of SparTag.us. As shown here, after a web

Figure 6. Browser toolbar of SparTag.us.

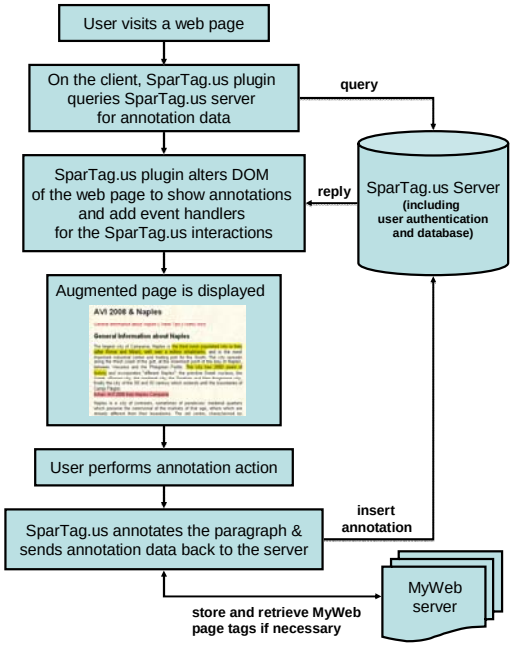


Figure 7. Architecture diagram of SparTag.us.

page is loaded onto Firefox, the GreaseMonkey script extracts the paragraph texts and forwards them to the SparTag.us server. In return, it retrieves annotation data created by the user for the paragraphs. Subsequently, GreaseMonkey alters the DOM of the page to display the annotations and control its event handling. When the user tags or highlights a paragraph, the script submits the annotation data to the SparTag.us server, which is then stored in a database.

The SparTag.us server is an Apache Tomcat server that runs Java servlets and connects to a MySQL database. The database stores the user's paragraph tags and highlights. For example, a tag entry includes things such as what the tags are, which paragraph the tags are attached to, who created the tags and when. The server also offers other services such as user authentication. In the spirit of mashups, we also support page tagging, using Yahoo's MyWeb to store and retrieve page tags. We created this mashup to offer both paragraph and page taggings to the user as well as to take advantage of MyWeb's vast user community and its services.

5. EMPIRICAL STUDIES

In conducting our evaluation of SparTag.us and Click2Tag, we were interested in two issues. First, we wanted to understand whether Click2Tag was indeed a low cost technique, when compared to the traditional way of tagging by typing. Second, we needed to look at the system usability and at how the system is used by real users in everyday practice.

5.1 Evaluation of Click2Tag

We conducted a behavioral experiment to study how Click2Tag compared to type-to-tag or no-tagging. In this experiment, we were interested in the relative costs of Click2Tag and type-to-tag. Our study was part of a more complex experiment that examined comprehension and memory effects. As we shall mention later, a preliminary eye-tracking study showed that processing is quite different in the two conditions, and it led us to also hypothesize that the two tagging techniques would have different effects on comprehension and memory. Due to space constraints, we only present the findings related to encoding costs and just briefly mention the other memory-related results.

Participants. We recruited 27 participants for this study. Each participant was compensated with $20.

Materials. We selected 18 passages from news articles as well as from various web pages on the Internet. The passages reflected a variety of topics (medicine, education, general science, aviation, history, etc). On average, the passages were 267 words long (ranging from 253 to 279).

Procedure. The study was structured as a within-subject study. Participants had to perform 18 trials (each corresponding to one of the 18 passages). In each trial, participants read a passage, selected randomly from the list of 18 passages. Participants were instructed to read at their own paces, but if they spent more than 2 minutes on a trial, they were moved to the next trial. They were also told to try to retain as much content as they could from the text (and they had to remember that content in a different part of the experiment that we do not discuss here). The trial could belong to one of three interface conditions as follows:

(1) **No-tags:** In this condition, no tagging was performed.
(2) **Click2Tag:** Participants had to tag the passage with relevant words by clicking on words from the passage. The tags were displayed in a box under the passage and could not be modified by the participants.
(3) **Type-to-tag:** Participants had to tag the passage with any relevant tags that they could generate, and type those tags in a box under the passage.

Results. We performed ANOVAs with subjects as the random factor using the interface condition (no-tags, Click2Tag, type-to-tag) as an independent variable. Table 3 shows the average reading time and number of inputted tags per condition. There was a significant effect of condition ($F(2,52)=52.72$, $p<0.001$). Contrasts showed that participants spent less time in the no-tags condition than in the Click2Tag condition ($p<0.001$) and in the type-to-tag condition ($p<0.001$), and also participants in the Click2Tag condition were faster than in the type-to-tag condition ($p=0.05$). These results pointed to a time cost associated with the tagging conditions, and confirmed that Click2Tag was a lower-cost interaction than type-to-tag.

People tended to attach more tags ($p<0.001$) in the Click2Tag condition (6.95 on average) than in the type-to-tag condition (3.98 on average), suggesting that they took advantage of the ease to tag in the Click2Tag condition to attach more tags and they achieved this faster than in the type-to-tag condition. These results indicate that, since Click2Tag is faster per tag, people may end up with more tags per passage than in the type-to-tag condition. The implications of this effect on information retrieval and formation of tag folksonomies still remains to be explored.

Table 3. Reading times and number of tags.

	Reading Time (seconds)	Number of Tags
No-tags	81.38	0
Click2Tag	96.01	6.95
Type-to-tag	102.92	3.98

Implications for memory and learning. In the following we briefly review the implications for memory and learning for the two tagging mechanisms. Figure 8 (see also the Color Page) shows hotspot gaze maps for a participant going through a type-to-tag trial versus a Click2Tag trial. As suggested in the Click2Tag condition (see Figure 8b), people fixated more on particular words in the text. This appears to have led them to rehearse the text content more and to perform bottom-up, content-driven tagging. In contrast, in the type-to-tag condition (depicted in Figure 8a), people tended to fixate less on any specific words, possibly because they read the text and then used their background knowledge to generate tags for the text, as opposed to fixating on specific words. Thus, the tagging process was top-down, knowledge driven and entailed more elaboration and connection with prior knowledge.

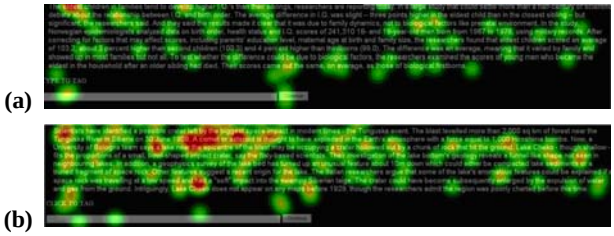


Figure 8. Hotspot gaze maps for two tagging techniques: (a) type-to-tag and (b) Click2Tag.

As a consequence of these two different processing styles, one would expect to obtain better retention of individual facts from text in the Click2Tag condition (since practice is related to recognition memory [22]), but better free recall in the type-to-tag condition (since elaboration improves recall [3]). Our data confirms that Click2Tag led to better recognition memory (both in terms of accuracy and time to recognize the facts) than type-to-tag. However, although there was a tendency for the type-to-tag condition to facilitate recall, this was not significant. In [5] we discuss these effects in more detail and argue that this lack of effect of type-to-tag on recall is due to the high cost of tagging by typing. Because they have to spend time typing rather than studying the text, people lose whatever memory gains they may get by having engaged in elaboration to find tags.

The memory data delivers one more piece of good news for Click2Tag: the recall efficiency of Click2Tag (i.e., number of facts recalled per unit of study time) was significantly better than that of type-to-tag, but not different from the recall efficiency of the no-tags condition. This result suggests that for type-to-tag, the supplemental cost of typing affects recall. While in Click2Tag, due to its low cost, the effects on recall are comparable to the no-tags condition, since most of the time is spent on reading.

5.2 Usability and User Feedback

We also conducted two separate user studies as part of an iterative design process to address how usable SparTag.us was and how it could be improved. The first study was conducted in the lab and consisted of people using SparTag.us to perform specific tasks, and then giving us feedback. In the second study, we had several employees of our company use SparTag.us for an extended period of time and we conducted in-depth interviews with some of these users to collect general feedback.

5.2.1 Usability Study

Participants. The users in our study were 10 company employees, older than 25, who volunteered their participation.

Procedure. Users were asked to complete six tasks. In the first three tasks, they used SparTag.us to find and tag information from the Web (e.g., *find a good pair of headphones under $200)*. The other three tasks required the users to use the SparTag.us notebook to retrieve information that they may have collected or encountered during the first three tasks (e.g., *find good ear bud headphones under $100*). We videotaped all the tasks. At the end of the study, users filled in a 41-item survey including the SUS usability questionnaire [4] and other questions derived from Nielsen's 10 usability heuristics [17], intended to help us understand the usability of SparTag.us. We also asked questions related to system features, and invited open-ended comments.

Results. According to the post-study survey, participants found SparTag.us intuitive and easy to use and felt that they could use the system without having to read the help page. With regard to usability, SparTag.us was rated as a moderately usable system. The average SUS usability score was 64 (ranging from 45 to 82.5, standard deviation 13.13). For comparison, Windows ME has a score of 55.73 [16]. Average scores for SUS are typically between 65 and 70. The following are several main lessons that we learned from the users' feedback:

- Most users considered Click2Tag and the low cost interface as the main benefit of SparTag.us (e.g., *[I liked the] easy addition of tags without extra pop-ups and the easy display of tags overlapped over a page*).
- Paragraph tagging (e.g., *[I liked] annotating in-depth, not stuck with just a label for the entire content, which is a problem of most tagging systems*) as well as easy search and retrieval using the notebook were other mentioned benefits.
- Users appreciated SparTag.us for personal data organization (*[I liked the] ability to capture and organize information. It was easy to learn and intuitive*, and *it appealed to my personal interest).* Indeed, 7 out of 10 users said they would use SparTag.us primarily for the purpose of organizing personal information. Moreover, 9 out of 10 participants said that, when they revisited a page, it was useful to see the annotations they had made on previous visits.
- Due to our initial design, users felt that tagging was easier than highlighting and that highlighting was too hard to perform without making mistakes.
- Even though the highlighting design had problems, most users (70%) felt that the combination of tagging and highlighting was most useful, rather than only tagging or only highlighting.

Improvements made. A close inspection of the videos taken in the study revealed that highlighting provided little feedback to the participants on whether the highlighting interactions were successful. In the next version of SparTag.us, we created separate modes for highlighting, highlight erasing, and text selection, respectively. We also observed that the participants experienced difficulties in making precise mouse clicks to define the highlighting boundaries. We resolved this shortcoming by tracking where the mouse enters and exits a paragraph, and updating the highlight as the user moves the mouse. This mirrors the familiar text selection interactions.

5.2.2 Assessment by Field Trial

Our experience in the laboratory suggested that SparTag.us streamlines a reader's entry of annotations to web content. We wished to examine how this works in people's everyday practices. Consequently, we conducted a 6-week field trial of SparTag.us with 18 people in our own organization outside of the design team. This allowed close support and interaction with users in one social setting, but also exposed the new technology to some range of diversity. Participants ranged in role from summer interns to senior staff and managers, working within four different areas of our organization. The development team stated no constraints or guidance other than feature description and support.

Our initial assessment of use practices is through in-depth interviews with selected users. Based on the observed frequency of use, we chose four participants for discussions conducted at least several weeks after initial deployment. We selected four people who gave the system an extended trial (above average in terms of pages viewed with SparTag.us running). Three of the participants (Interviewees 1, 2 and 3) were above average users of the technology in terms of the amount of annotations made, and one (Interviewee 4) was below average.

Interviews were conducted by a researcher without prior involvement in the project, and were semi-structured, loosely following 26 pre-planned questions, but allowing the interviewee to cover the material as the topics arose in the interview. We designed the questions to walk the person through each of the various components and features of the SparTag.us interface, and phrased them to elicit stories about instances of application use (e.g., "*Could you tell me about the last time you used SparTag.us? When was that? What were you doing? What happened?")*. These questions were then followed up for as many different types of instances as the person could recall. Each interview was conducted in the office of the user and audio-video recorded. Participants were allowed to use the interface to demonstrate actions and look up information in the system to refresh their memories of events.

Reported benefits. People did engage with tagging at the paragraph level (in situ) and utilized the highlighting feature to mark salient points. As Interviewee 3 said while talking about a computer operation web page, "*That part [of the document] was interesting to me, not so much the other parts. But this is something I spend some time looking for and I really want to get a handle. I highlighted that. Oh, yeah, then I put some tags in, dual boot, sort of reminding myself. Remind myself that this is about partitioning and dual booting."*

The focused annotation on specific subsets of a document, and the ease of tagging and capturing paragraph text in the notebook without extra effort was mentioned as what interviewees like best about the system. Interviewee 2 said, "[…]*with ClipClip I need to type, notes, to remind myself, type the tags. I don't like that*"; and "*So I like the notebook page, it gives me at a glance, like the specific content I was interested in as well as the URL. So it serves the same purpose that I use del.icio.us for because it saves the URL, but it gives me more at a glance the interesting content.*"

Reported limitations. All users interviewed reported turning the system on and off (and then forgetting to turn it back on for a while). Much had to do with usability and performance of an early prototype. Of greater interest was that all users reported accidentally tagging things. Each developed the workaround strategy of abandoning the page (and hence not saving the erroneous tags). This points out that, while the Click2Tag implementation was described as easy to invoke, not all people will wish to invoke it in the same way with respect to their own browsing practices (Interviewee 4, "*I had to tell it to stop highlighting things. I would prefer it to be much more in the background and become available when I want it*").

Improvements made. From the findings of our field trial, we improved the performance by reworking the SparTag.us client implementation. We also discovered that our highlighting implementation had led to problems with some pages containing HTML forms. To address this issue, we overloaded the text selection functionality of the browser, using the browser's selection object to define the highlighting region. To enable users to find out the status of SparTag.us, turn it on if not, and repair tagging or highlighting miscues, we developed the SparTag.us toolbar, as described in Section 4.3. With one mouse click, users can now turn on or off SparTag.us and see the effect immediately. The toolbar also shows which mode (highlighting, highlight erasing, or text selection) SparTag.us is currently in, and the mode can be changed interactively. Furthermore, we added additional feedback and correction mechanisms. For example, users can easily see what text is annotatable and undo a tagging or highlighting operation with a mouse click or a shortcut key.

6. CONCLUSION AND FUTURE WORK

We have described a novel tagging system called SparTag.us that introduces a new technique, Click2Tag, to support low cost, in situ tagging. One motivation for the Click2Tag technique came from our analysis of del.icio.us data, which showed that a large number of tagging events involved applying tags that came from the page content. Another motivation was that tagging in current social bookmarking systems has relatively high interaction and attention-switching costs. Our first user study examined how the cost of Click2Tag compared with that of typing to tag. The results suggested that people are more efficient with Click2Tag. They process the text faster and tend to attach more tags.

SparTag.us also allows in situ highlighting and automatically stores all the paragraphs that a user has highlighted or tagged into a notebook. We described two user studies addressing the usability of SparTag.us. These studies have been informative in making interface design decisions. They also reassured us of the potential user acceptance of our ideas on Click2Tag, the combination of tagging and highlighting, and the notebook. Obviously, Click2Tag is designed for web pages whose contents are mainly textual, not for pages consisting of media contents (e.g., images and videos).

There are many issues to be addressed in our future work. First, we need to compare the quality of tags generated by Click2Tag versus typing, as well as their effectiveness in information search and retrieval. In a recent paper [5], we reported the different implications of these two tagging techniques on memory and learning, which shed some light on the potential effects on later retrieval using these tags. Moreover, the folksonomies arising from these two techniques are likely to be different, since, as we discuss in Section 5.1, Click2Tag is content-driven, whereas typing is more knowledge driven.

Second, another question of interest is how in situ tagging and highlighting affect within-page navigation. If you come back to a

page and see your own (or others') annotations, will you be able to retrieve content of interest faster? Generally, we believe that tagging and highlighting support different functions, although there are certainly some overlaps. Tagging seems to help with navigation mostly, whereas highlighting seems to support learning either at encoding time or at reviewing time [7,18,20,25]. Future research is needed to understand whether the combination of tagging and highlighting will offer more support for gist reconstruction than tagging alone.

Third, we plan to develop the social aspect of SparTag.us. In this paper we focused the description on the individual usage of SparTag.us, under the assumption that we first need to make the individual experience as effective as possible in order to be able to attract users to the system. Indeed, despite various burrs to be expected of the early prototypes, users found our system to be as usable as existing well-developed systems. The next step of our research is to explore the design space of social functionalities. We are interested in finding out what kind of features would support social interaction and communication (e.g., seeing your friends' tags or notebooks, being able to share tagged or highlighted content via email or other means, etc).

Lastly, we are interested in exploring the Click2Tag concept on documents other than web pages (e.g., PDF and MS Word documents). Since for most people web browsing is hardly the only means to acquire new knowledge, we plan to investigate how to share annotations across documents of different formats.

7. ACKNOWLEDGMENTS

We thank the user study participants, Lawrence Lee for helpful discussion, and Christiaan Royer for help in experimental setup.

8. REFERENCES

[1] Bush, V. (1945). As We May Think. The Atlantic Monthly, 176(1), 101-108.

[2] Benkler, Y. (2002). Coase's Penguin, or Linux and the Nature of the Firm. Yale Law Journal, 112, 367-445.

[3] Bradshaw, G. and Anderson, J. (1982). Elaborative Encoding as an Explanation of Levels of Processing. Journal of Verbal Learning and Verbal Behavior, 21, 165-174.

[4] Brooke, J. (1996). SUS: A "Quick and Dirty" Usability Scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland (eds.), Usability Evaluation in Industry. Taylor and Francis, London.

[5] Budiu, R., Pirolli, P., and Hong, L. (2007). Remembrance of Things Tagged: How Tagging Affects Human Information Processing. Under submission.

[6] Card, S., Moran, T., and Newell, A. (1983). The Psychology of Human Computer Interaction. Lawrence Erlbaum.

[7] Chi, E., Gumbrecht, M., and Hong, L. (2007). Visual Foraging of Highlighted Text: An Eye-Tracking Study. Proc. HCI International Conference, 589-598.

[8] Churchill, E., Trevor, J., Bly, S., Nelson, L., and Cubranic, D. (2000). Anchored Conversations: Chatting in the Context of a Document. Proc. CHI'00, 454-461.

[9] Golder, S. and Huberman, B. A. (2006). Usage Patterns of Collaborative Tagging Systems. Journal of Information Science, 32(2), 198-208.

[10] GreaseMonkey. DOI=https://addons.mozilla.org/en-US/firefox/addon/748.

[11] Grundin, J. (1994). Groupware and Social Dynamics: Eight Challenges for Developers. Communications of the ACM, 37(1), 92-105.

[12] Lee, K. J. (2006). What Goes Around Comes Around: An Analysis of del.icio.us as Social Space. Proc. CSCW'06, 191-194.

[13] Linderman, M. (2005). Tag Formats: Can't We All Just Get Along? DOI=http://37signals.com/svn/archives2/tag_formats_cant_we_all_just_get_along.php.

[14] Marshall, C. (1997). Annotation: From Paper Books to the Digital Library. Proc. Digital Libraries'97, 131-140.

[15] Marshall, C. and Brush, A. (2004). Exploring the Relationship between Personal and Public Annotations. Proc. Digital Libraries'04, 349-357.

[16] Microsoft. DOI=http://download.microsoft.com/download/d/8/1/d810ce49-d481-4a55-ae63-3fe2800cbabd/ME_Public.doc.

[17] Nielsen, J. (1994). Heuristic Evaluation. In J. Nielsen and R. L. Mack (eds.), Usability Inspection Methods. John Wiley and Sons, New York, NY.

[18] Nist, S. L. and Hogrebe, M. C. (1987). The Role of Underlining and Annotating in Remembering Textual Information. Reading Research and Instruction, 27(1), 12-25.

[19] O'Hara, K. and Sellen, A. (1997). A Comparison of Reading Paper and On-Line Documents. Proc. CHI'97, 335-342.

[20] Peterson, S. E. (1992). The Cognitive Functions of Underlining as a Study Technique. Reading Research and Instruction, 31(2), 49-56.

[21] Pirolli, P. (2007). Information Foraging Theory: Adaptive Interaction with Information. Oxford University Press, New York, NY.

[22] Pirolli, P. and Anderson, J. (1985). The Role of Practice in Fact Retrieval. Journal of Experimental Psychology: Learning, Memory, and Cognition, 11, 136-153.

[23] Schilit, B., Golovchinsky, G., and Price, M. (1998). Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. Proc. CHI'98, 249-256.

[24] Schraefel, M. C., Zhu, Y., Modjeska, D., Wigdor, D., and Zhao, S. (2002). Hunter Gatherer: Interaction Support for the Creation and Management of Within-Web-Page Collections. Proc. WWW '02, 172-181.

[25] Silvers, V. L. and Kreiner, D. S. (1997). The Effects of Pre-Existing Inappropriate Highlighting on Reading Comprehension. Reading Research and Instruction, 36(3), 217-223.

[26] Thomas, J. and Cook, K. (2005). Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE CS Press, Los Alamitos, CA.

Visualization Techniques

Timeline Trees: Visualizing Sequences of Transactions in Information Hierarchies

Michael Burch
University of Trier
burchm@uni-trier.de

Fabian Beck
University of Trier
fabian@fbeck.com

Stephan Diehl
University of Trier
diehl@uni-trier.de

ABSTRACT

In many applications transactions between the elements of an information hierarchy occur over time. For example, the product offers of a department store can be organized into product groups and subgroups to form an information hierarchy. A market basket consisting of the products bought by a customer forms a transaction. Market baskets of one or more customers can be ordered by time into a sequence of transactions. Each item in a transaction is associated with a measure, for example, the amount paid for a product.

In this paper we present a novel method for visualizing sequences of these kinds of transactions in information hierarchies. It uses a tree layout to draw the hierarchy and a timeline to represent progression of transactions in the hierarchy. We have developed several interaction techniques that allow the users to explore the data. Smooth animations help them to track the transitions between views. The usefulness of the approach is illustrated by examples from several very different application domains.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Miscellaneous

General Terms

Algorithms, design, human factors.

Keywords

Visualization, hierarchy, time.

1. INTRODUCTION

The visualization of hierarchical data is at the heart of information visualization. Information hierarchies exist in many application domains: hierarchical organization of companies, news topics and subtopics, file/directory systems, products and product groups of a department store, evolutionary or phylogenetic trees in biology.

AVI '08, 28-30 May , 2008, Napoli, Italy.

There has been a lot of work on visualizing such hierarchies using node-link diagrams [4], radial [7], or space-filling techniques like Treemaps [8], Information Slices [9], or Sunburst [10].

In the application domains mentioned above, there are also relations between elements in the hierarchy. For example, employees are related if they communicate with each other, topics are related if they are covered in the daily newscast, files are related if they are changed simultaneously by the same person, or products are related if they are bought by the same customer at the same time. Through these relations the participating elements together form a transaction.

So far, only few researchers have developed methods to visualize such transactions between elements of a hierarchy [11, 12, 13, 2].

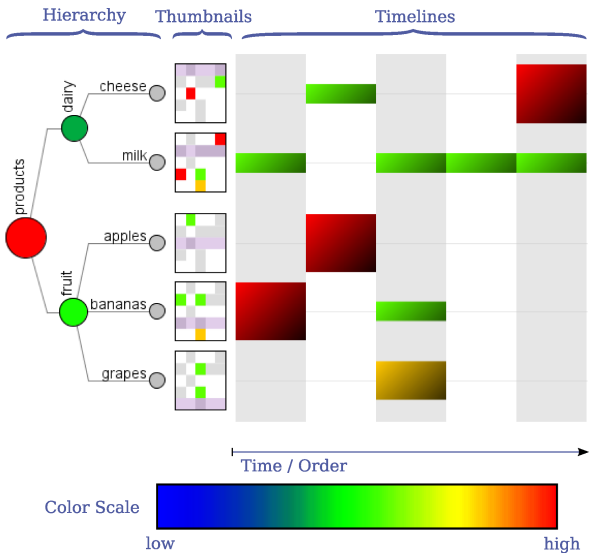


Figure 1: Visualization of the market basket example (see Table 1) as a Timeline Tree and additional explanations.

Often, we are not interested in a single transaction, but in a sequence of transactions that occur over time. Furthermore, the elements can be affected by or involved in a transaction to different extent. To model this we associate a measure with each transaction mapping each element of the transaction to a positive real number. Thus, in the application domains mentioned above the conversational partners, the selection of topics, the files, and the products bought are

the elements of transactions, while the duration of the communication, the extent of coverage of each topic, the size of the modification of each file, the amount paid for a product are the associated time-varying measures.

The Timeline Tree approach presented in this paper supports the visualization of such sequences of transactions in information hierarchies in a single diagram by integrating three views (see Figure 1):

Information Hierarchy: It shows the whole hierarchy to an interactively selectable level. By clicking at a node that is currently displayed as a leaf, it is expanded; by clicking at an intermediate node, the subtree starting at that node is collapsed. Expanding or collapsing subtrees of the hierarchy can help to detect relations at different levels of abstraction.

Timelines: The sequence of transactions is visualized on a timeline drawn as an extension to the interactive tree. The elements of a transaction are represented by boxes, that are colored and sized according to the defined measure. Together with some alternative views and further features, that are introduced further below, the timeline visualization provides an extensive tool to explore and analyze the transactions.

Thumbnails: Small representations of the timeline view at each leaf node or at each collapsed node of the hierarchy enable the user to detect dependencies in which the element(s) represented by that node are involved.

We have developed several interaction techniques that allow the users to explore the data. Smooth animations help them to track the transitions between views.

To the best of our knowledge Timeline Trees is the first approach that allows users to visually explore transactions in information hierarchies. The user can analyze the evolution of transactions, the roles of their member elements, and detect when and how strong elements of the hierarchy are related.

The rest of this paper is organized as follows: In Section 2 we give a formal model of the kind of data that can be visualized by Timeline Trees. Next, we introduce the main features of our technique in Section 3 by looking at a simple example. Then, in Section 4 we illustrate the usefulness of these features by looking at data sets from various application domains. Related work is discussed in Section 5, and finally, Section 6 gives some conclusions.

2. DATA MODEL

For the purposes of this paper we model an information hierarchy as a tree where the leaf nodes represent some pieces of information, that we call items in the rest of this paper. Let $T = (V, E)$ be such a tree where V is a set of nodes, E a set of directed edges and $I \subset V$ the set of leaf nodes.

Furthermore let (μ_n) be a sequence of measures for $n \in \mathbb{N}$ where $\mu_i : 2^I \to \mathbb{R}_0^+$, $1 \leq i \leq n$, is an arbitrary measure defined on the set of items [1]. So we can model a transaction t_i for $1 \leq i \leq n$ as a set of items: $t_i := \{v \in I \mid \mu_i(\{v\}) > 0\}$.

We define a function $\mathit{front} : V \to 2^I$, that maps a node $v \in V$ on the set of its reachable leaf nodes $I' \subset I$, and functions $\hat{\mu}_i(v) := \mu_i(\mathit{front}(v))$, $1 \leq i \leq n$, that extend the measures to arbitrary nodes $v \in V$.

Also note, that Timeline Trees focus on visualizing static hierarchies, nevertheless minor changes in the hierarchy can be simply transformed to fit our data model. For instance, if an item moves in the hierarchy, it will be displayed in both positions and handled like two different items.

3. FEATURES

Table 1: Example: Market baskets.

Day	Market basket and money spent
Monday:	milk \$1, bananas \$3
Tuesday:	cheese \$1, apples \$3
Wednesday:	milk \$1, bananas \$1, grapes \$2
Thursday:	milk \$1
Friday:	milk \$1, cheese \$3

To illustrate the features of our visualization technique we use the small data set given in Table 1. It shows the market baskets of five subsequent days, i.e. the products and prices of each product that a person bought. In our example, we use the price as the measure.

For example, the third transaction corresponding to Wednesday is $t_3 = \{\mathsf{milk}, \mathsf{bananas}, \mathsf{grapes}\}$ where the value of the measure for grapes is $\mu_3(\{\mathsf{grapes}\}) = 2$. Figure 1 shows the market basket example in the Timeline Tree visualization. The visualization is composed of three views: the information hierarchy, the thumbnails, and the timelines. We discuss each of these views below.

3.1 View: Information Hierarchy

We start describing the visualization at the tree diagram of the information hierarchy on the left side. It uses a customary node-link representation where the size and color of a node encode the number of items that are descendants of the node, so one can identify major nodes even if they are collapsed.

Actually, the most important interaction functions of the tree diagram are collapsing and expanding of nodes with smooth transitions. This enables the user to explore larger information hierarchies without loosing focus and to compare data on different levels of abstraction.

The goal of the tree layout is to efficiently display the tree with labeled nodes and to let subtrees visually emerge. The former goal is realized by using more horizontal space with increasing depth of the nodes, so the tree gets more space-filling, and by adapting the label's orientation and size to the available space. The nearly orthogonal layout and smaller vertical distances between siblings help to reach the latter goal. Furthermore tooltips provide detailed information about the nodes.

3.2 View: Timelines

Timeline Trees visualize a sequence of transactions as sets of boxes, that are ordered from left to right on a diagram we refer to as a 'timeline'—in many applications time provides a natural order on the transactions. Each box represents one member element of a transaction and is positioned in the same column as the other members of this transaction and in the row of the according item.

The measure $\mu_i(\{v\})$ of an item v in transaction t_i is redundantly encoded by color and height of the box, whereas its width is fixed. So the size of a box increases linearly with the measure. But the user can also switch to fixed heights because in some applications the 'importance' of an element does not correlate with the measure.

We included numerous predefined color scales for color coding such that the user can select a suitable color scale for the task at hand. Discerning two adjacent, similarly colored boxes might be difficult, so we use a brightness gradient as a kind of cushion effect [14].

So far we discussed how to draw boxes for single items. Next we look at how to handle collapsed nodes. For that we use the function $\hat{\mu}_i$ which is defined for each node as the sum of the measure values of all leaf nodes reachable from this node (see Section 2). In our visualization this sum can be either drawn as a single box or as several vertically stacked boxes, each representing a single item. Both modes can be useful for different applications (see Section 4).

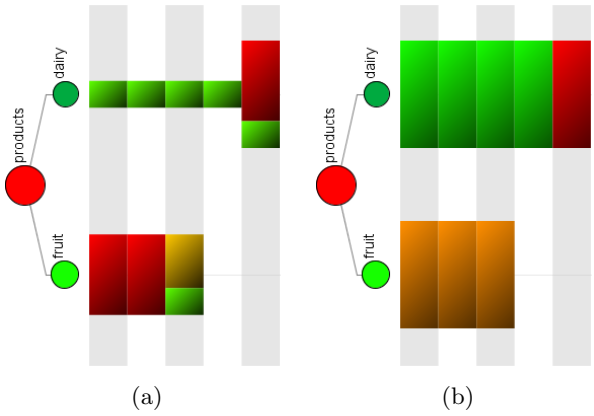


Figure 2: Market basket with collapsed nodes 'dairy' and 'fruit' in different modes: (a) height represents the measure, collapsed items are stacked; (b) unified heights, summed measure values for collapsed items.

During the interactive exploration process of a data set, a good orientation and an easy access to additional information is very important. Our visualization supports these aspects by highlighting the row and column marked by the current position of the mouse cursor and by detailed tooltip texts as shown in Figure 3. Another very useful feature is the masking of transactions. To this end, the user can select some items or collapsed nodes to form a mask set M. Only those transactions $T_M = \{t_i | t_i \cap M = M\}$ that match the mask set will be shown. All transactions that do not contain all nodes of the mask set are faded out. As a result, the user can focus on the relations between the nodes in the mask set.

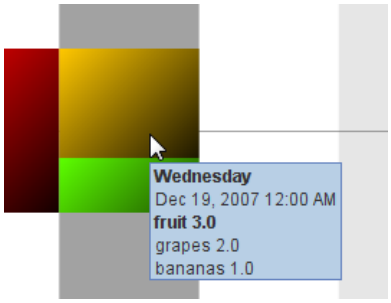


Figure 3: Tooltip for the collapsed 'fruit' node in the Wednesday transaction of the market basket example (see also Figure 2).

3.3 View: Thumbnails

The idea of masking transactions is extended by the thumbnail views of the timeline diagram. These thumbnails are displayed for every item or collapsed node at the right side of the tree diagram. They show the transactions from the perspective of the according node as if this node would be the only element of the mask set. In other words, only those transactions the node is member of are represented in the thumbnail using the selected color code, the remaining transactions are only drawn as gray boxes. As for the general mask set, the thumbnails are a good tool for identifying correlations between nodes, but in contrast to the mask set, the thumbnails are simultaneously shown for each item or collapsed nodes.

To assist orientation in the thumbnails, within a thumbnail the row of the node related to the thumbnail is highlighted as a slightly colored line. Furthermore, to countervail the disadvantage of the thumbnails' relatively small size, we implemented a magnification lens functionality that enlarges parts of the thumbnails when the mouse cursor moves over them.

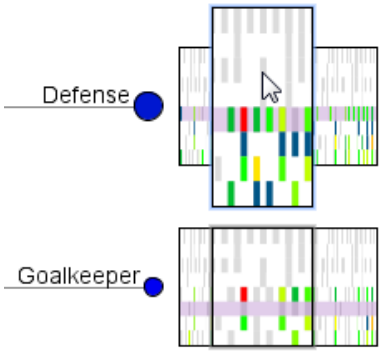


Figure 4: Thumbnail example with lens function whereas the mouse cursor is over the 'Defense' thumbnail (detailed view of the soccer match visualization presented in Section 4.1).

3.4 Alternative Representation: Time Bars

In addition to the visualization discussed above, Timeline Trees include an alternative representation of the transactions which is shown in Figure 5 for the market basket example. Here, the time or order of transactions is encoded using color coding and the measure is represented by the width of the boxes instead of their height. The boxes are drawn from left to right attached to each other, instead of positioning them in separate columns as in the Timeline representation. Thus, boxes related to the same transaction are no longer in the same column, but they have the same color. As the resulting representation is very similar to a bar chart, we call it Time Bars.

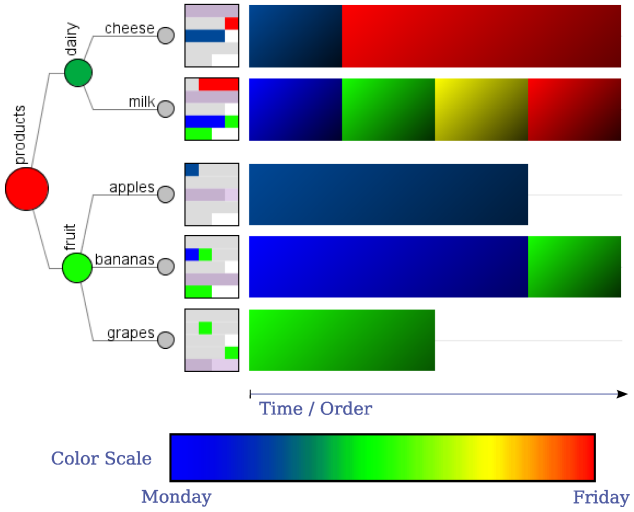


Figure 5: Visualization of the market basket example in Time Bars view.

We use Time Bars only in addition to the default visualization because the color coding of time is not so intuitive, and discerning transactions in time is not accurate enough. But for many analyses it provides the following advantages:

- The Time Bars form a kind of bar diagram that represents the aggregated measures of the items and currently collapsed nodes. So for example, one can easily detect which node is the most 'active' one.
- The shape of the diagram is much more memorable and one establishes a sort of mental map while exploring the data. Thus the orientation in the diagram and especially in the thumbnails is significantly better.
- The distribution of colors gives a more holistic overview of the temporal progress of the transactions: One can detect differences at first sight.

4. APPLICATIONS

To illustrate the features of our visualization system, we apply the system to data sets of very different domains.

4.1 Team Play in a Soccer Match

Soccer teams are hierarchically organized. Eleven players belong to each team and are subdivided into different team parts: the goalkeeper, the defense, the midfield and the offense. Additionally, players have their specific location or area on the soccer ground where they act. The number of contacts with the ball and the different players belonging to a move of the match can be seen as a transaction where each element has a measure namely the number of contacts.

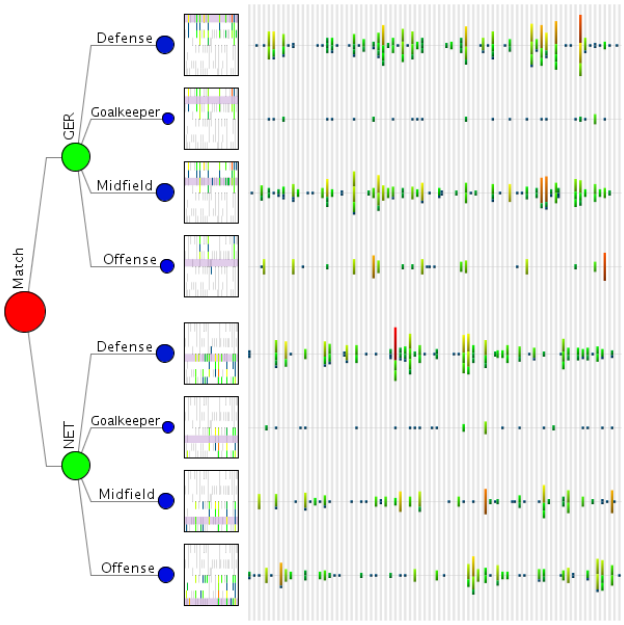


Figure 6: Timeline Trees for the soccer match between Germany and the Netherlands in World Cup Championships 1990 in Italy on team part level.

Figure 6 shows the moves of the first half of a soccer match[1]. In this visualization the organizational structure of a soccer team in terms of offense, defense, midfield and the individual players forms the hierarchy and is represented as a node-link diagram at the left hand side. Players are related to each other, if they take part in the same move which can be observed by the thumbnail view in each of the small boxes. Here, we define move as the time period during which a team has the exclusive ball possession. As the measure of a move we use the number of passes, i.e. how often the ball was passed from one member of the team to another member of the same team. Many ball contacts are indicated by high bars and red color, whereas a green color stands for a little contacts in a move, yellow is a value in between.

In Figure 6 we can also make very interesting observations about the first half of the match. The hierarchy is expanded to the level of team parts. Both defenses are the parts with the most ball contacts. The goalkeepers have only very little contacts, which is an absolutely normal phenomenon. The German offense acts not as much as their counterpart from the Netherlands. But the German midfield takes this part and therefore has much more ball contacts than the one of the Netherlands. A closer look at the lowest timeline in this figure reveals that the offense of the Netherlands increases their number of ball contacts towards the end of this first half.

[1]Germany vs. Netherlands (2:1) at the World Championship 1990 in Italy

In Figure 7 the German midfield and offense as well as the defense of the Netherlands is expanded. The thumbnail view can give us the information that there is one transaction in which one player of each team is involved. Frank Rijkaard and Rudi Völler both received the red card and are ejected from the game. This detail on demand information can be requested by a tooltip when moving to the position of one of the corresponding bars. After this 21st minute of the first half, the following observation can be made. Frank Rijkaard was a defending player and it can be expected that the other players belonging to the defense have to do the work of the missing player. And in fact this is true. The players Adrie van Tiggelen and Ronald Koeman have much more ball contacts than before this 21st minute. Another observation is that the ball contacts of the whole offense part of the Netherlands increase right after this 21st minute and naturally the defense of Germany in the same way.

4.2 Software Evolution

Open Source software systems under version control can be used to gain interesting insights of the development process of the software. One important observation can be which files have been changed together to what extent. Furthermore it can be referred which files have been developed in which period of time. These facts can be very helpful to support software developers during the evolution process of their current project.

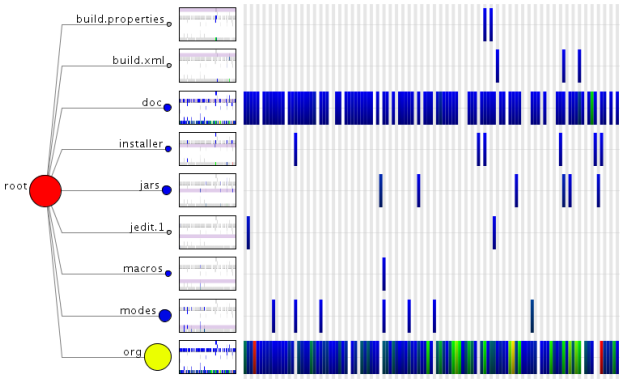


Figure 8: Transactions of a part of the JEDIT Open Source software project.

Figure 8 shows the Timeline Tree visualization for a time period of the development of the JEDIT [5] Open Source software project. In this figure, the two overall blue colored lines indicate that two software artifacts are in the center of the evolution process. The upper one corresponds to the `doc` subdirectory and the one in the lower part represents the whole source code subdirectory of the project.

Most of the transactions contain at least one file of the source code subdirectory. Documentation and source code are changed together very frequently. This can be a hint that developers almost always document their changes immediately.

A closer look at the selection of transactions by the mask set in Figure 9 that contains both documentation files `TODO.txt` and `CHANGES.txt` reveals that in nearly each case when a developer changes the file `CHANGES.txt` he also changes the file `TODO.txt`. The inverse only holds in about 50 percent of the transactions. Our hypothesis is that if someone makes a change to the `CHANGES.txt` file he always has to adjust the `TODO.txt` file because the change solves a problem or implements a feature contained in the to-do list.

4.3 World's Export

Using Time Bars instead of timelines our visualization can be used as an augmented bar chart diagram. The bars are generated by stacking the boxes of each time interval. Additionally to the conventional approach, the single bars are colored with respect to their corresponding time interval. This approach can help to observe in which time interval a bar grows more rapidly than others.

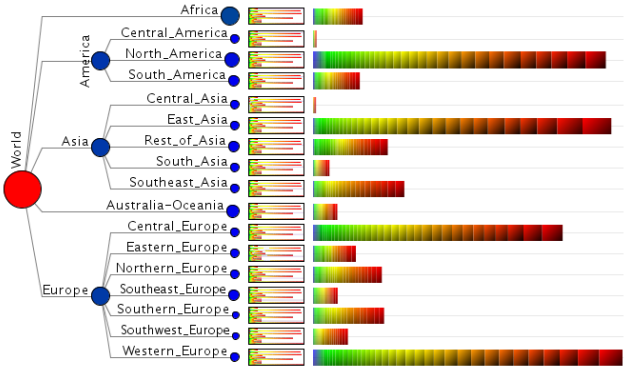


Figure 10: Export data (in Dollar) of the world's regions in Time Bars view from 1948-2005.

Figure 10 shows the yearly export data in terms of dollars for the whole world from 1948 to 2005. The year of a transaction is indicated by color, where blue indicates older transactions and red indicates more recent ones. Green, yellow and orange colors are in between. We can immediately see that Western Europe has the biggest export value for this time interval followed by East Asia, North America and Central Europe. The hierarchy can be expanded to the country level to gain insights about the export data of each country of the world. Another interesting observation is that the whole continent of Africa exports less than Southern Europe for example.

The steady growth of the single boxes from left to right is largely caused by inflation.

4.4 Movies

For this application we looked at the starring actors of movies. For each movie the set of actors is regarded as a transaction, and all movies can be chronologically ordered.

Figure 11 shows the starring actors in movies directed by Martin Scorsese. Actors are hierarchically organized according to the number of their nominations and Academy Awards. Figure 11 is divided into three parts. The left one shows the hierarchy, in the middle part the transactions are represented as well as the thumbnails and the right part shows the Time Bars view. The longest Time Bar is related to Robert de Niro. Many of the nominated actors only starred in one movie directed by Martin Scorsese. Only Scorsese himself and Leonardo Di Caprio appear three times. A closer look at the Time Bars or the timelines reveals that Robert de Niro was a starring actor in the past, whereas

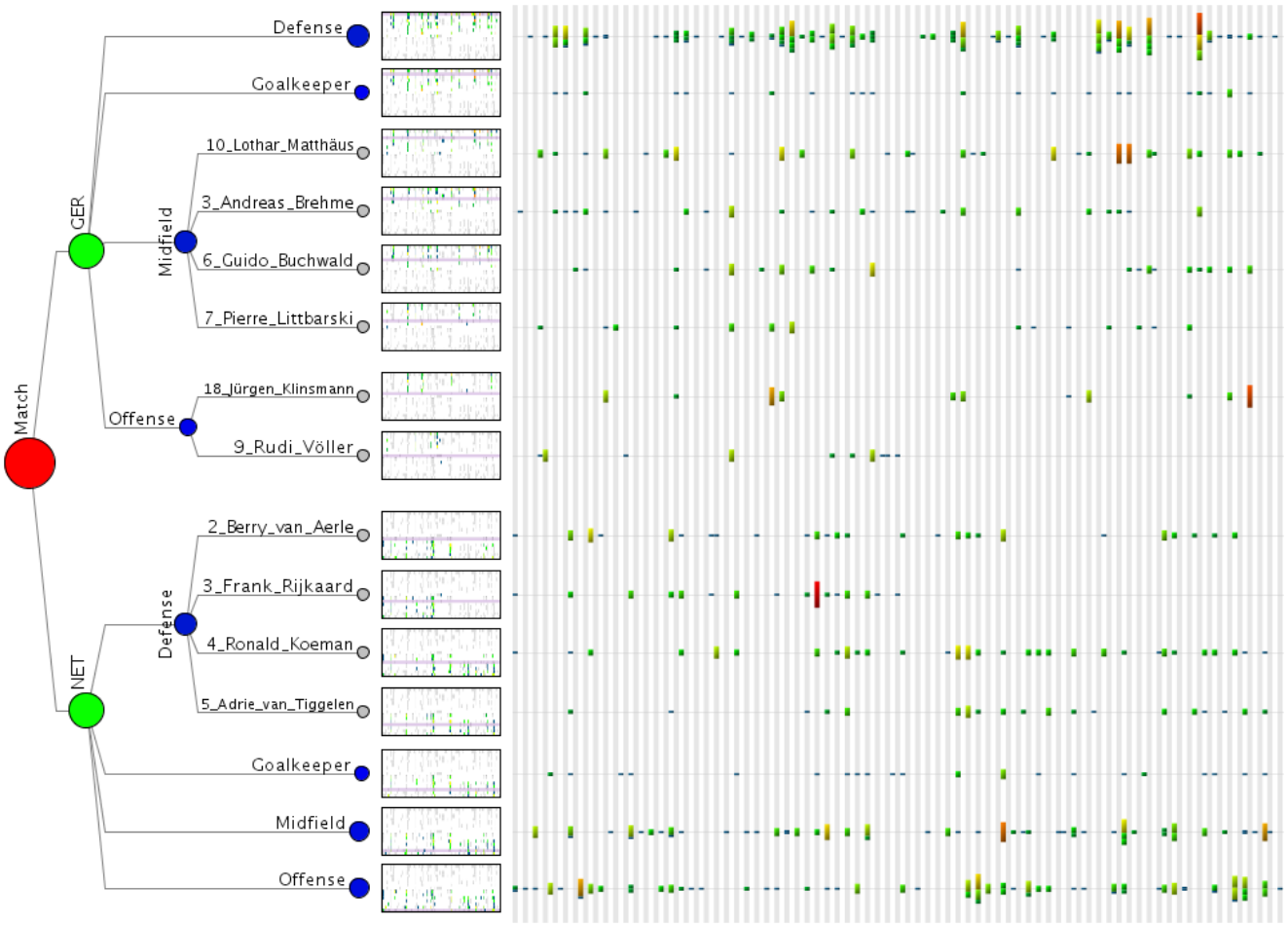


Figure 7: Timeline Trees for the soccer match with expanded team part subhierarchies.

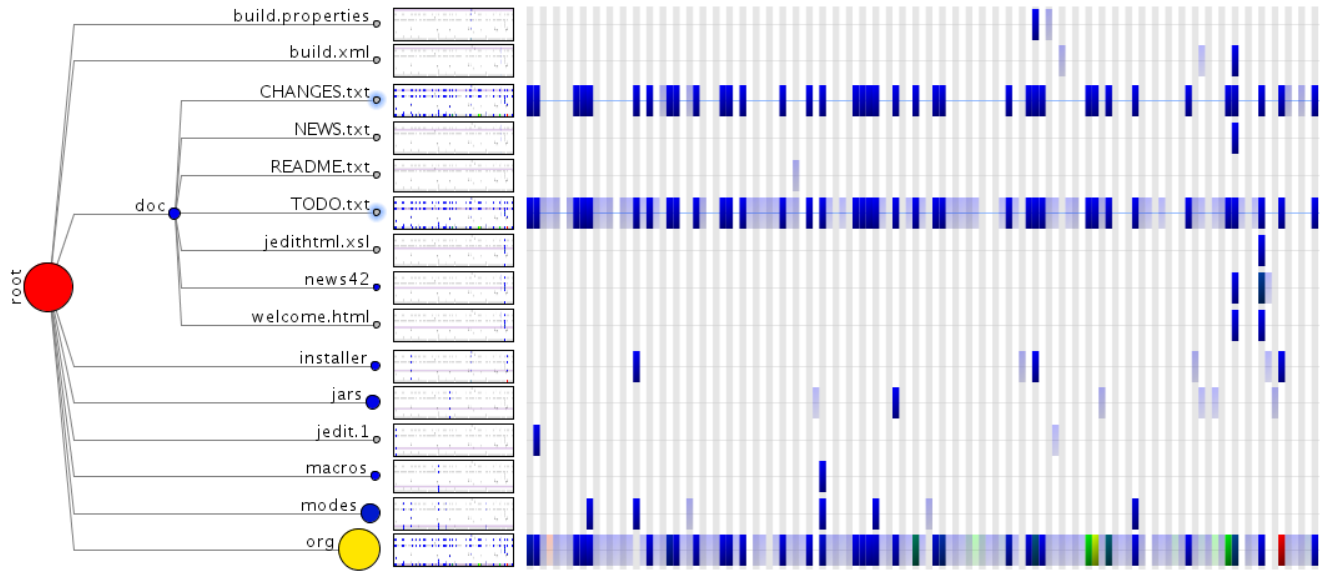


Figure 9: Timeline Trees with two files in the mask set, common transactions of the masked files are highlighted.

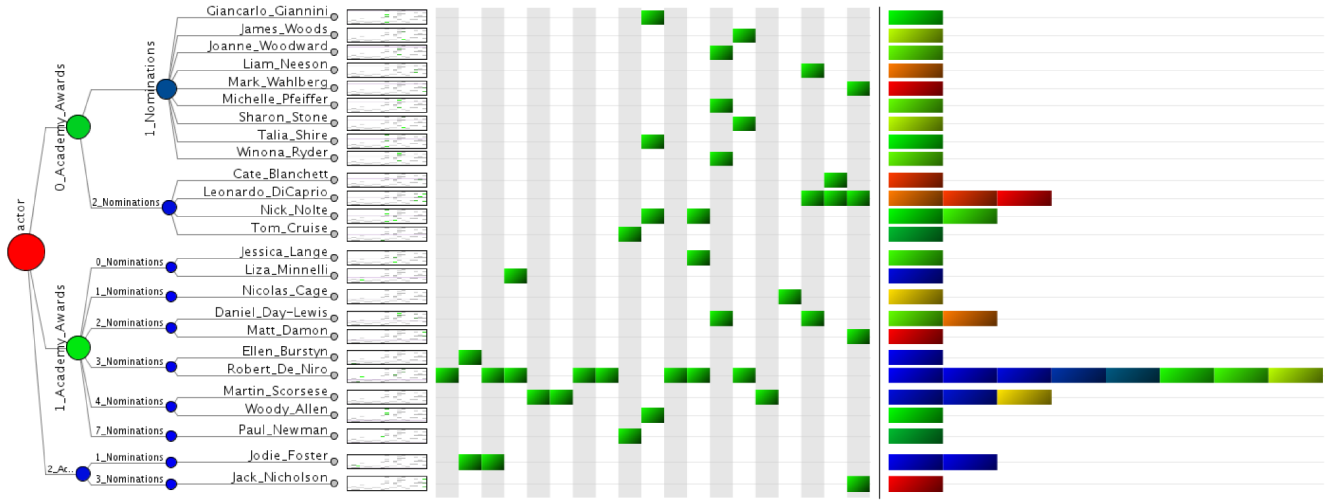


Figure 11: Movies directed by Martin Scorsese (as transactions) and starring actors (as items), that are classified by Academy Award wins and nomimations.

Leonardo di Caprio was a starring actor in the last three movies. The thumbnail view can be used to make observations about costarring actors. Only thumbnails of Nicolas Cage and Martin Scorsese don't have any colored elements. This means they have never been costarring with any other nominees or awardees in movies directed by Martin Scorsese. Furthermore, only actors with at least one Academy Award or no nomination at all played in Scorsese's first eight movies.

5. RELATED WORK

As Timeline Trees integrate views for hierarchies, relations and chronological data, we discuss related work with respect to these.

5.1 Visualization of Hierarchies

Information hierarchies can be seen as a special kind of graphs, namely trees. As a result, information hierarchies can be visualized as node-link diagrams using specialized graph layout algorithms [4], but also space-filling techniques like Treemaps [8], Information Slices [9], or Sunburst [10] have been developed. In Timeline Trees we use a node-link diagram to visualize the hierarchy.

5.2 Visualization of Relations in Hierarchies

Many approaches try to encode existing relations between objects as directed or undirected edges in node-link diagrams. The appearance of edges, for example, their color, shape, orientation, thickness or connection can represent a measure, e.g. the strength of a relation.

There are several approaches that extend the Treemap approach [8] to also show different kinds of relations [12, 13, 2]. For example, ARCTREES [11] is an interactive visualization tool for hierarchical and non-hierarchical relations. It extends the hierarchical view of the Treemap approach with arc diagrams to present relations. In Timeline Trees items are related by transactions. Transactions and the related measure are encoded by the position and color of boxes.

5.3 Visualization of Chronological Data

The ThemeRiver visualization shows the thematic changes of a collection of documents as a set of "rivers" along a time line from left to right [15]. Each river represents a theme and the strength of the theme at a certain point in time is depicted by the width of the river.

Other visualization of events along a time axis [3, 16] focus on the duration of events or when an event has been sent or received. In particular, for the visualization of parallel systems [6] visualization with parallel time axes have been used early on. In Timeline Trees we also use parallel time axes but with a focus on the concurrent participation in transactions.

Furthermore, we have to mention that the notion of Timeline Trees has already been indroduced in [17] but is used for a different concept: There, Timeline Trees describe branching timelines in an interactive multimedia scenario.

6. CONCLUSIONS

We have introduced Timeline Trees as a visualization technique to explore sequences of transactions in information hierarchies. We discussed the various features of the visualization technique, and illustrated their usefulness by applying it to data sets from very different domains. These applications illustrate that Timeline Trees aid users to detect

- global and local trends with respect to the frequency and strength (measure) of the transactions, and
- relations between lower and higher levels of abstraction in the information hierarchy.

7. ACKNOWLEDGMENTS

The authors wish to thank Peter Weißgerber and Felix Bott for providing data sets and constructive comments.

8. REFERENCES

[1] P. R. Halmos. *Measure Theory*. Van Nostrand, New York, NY, 1950.

[2] M. Burch and S. Diehl. Trees in a treemap. In *Proceedings of 13th Conference on Visualization and Data Analysis (VDA 2006)*, San Jose, California, 2006.

[3] G. M. Karam. Visualization using timelines. In *Proceedings of the 1994 ACM SIGSOFT international Symposium on Software Testing and Analysis ISSTA'94*, pages 125–137, New York, NY, USA, 1994. ACM.

[4] E. M. Reingold and J. S. Tilford. Tidier drawing of trees. *IEEE Transactions on Software Engineering*, 7(2):223–228, 1981.

[5] Slava Pestov and Contributors. JEDIT project homepage. `http://www.jedit.org/`.

[6] E. Kraemer and J. T. Stasko. The visualization of parallel systems: An overview. *Journal of Parallel and Distributed Computing*, 18(6):36–46, 1993.

[7] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'01), San Diego, CA, USA*. IEEE, 2001.

[8] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of IEEE Visualization Conference*, pages 284–291, San Diego, CA, 1991.

[9] K. Andrews and H. Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs (late breaking hot topic paper). In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'98)*, pages 9–12, Research Triangle Park, NC, 1998.

[10] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the Symposium on Information Visualization (InfoVis'00), Salt Lake City, UT*, pages 57–65, Washington, DC, 2000. IEEE Computer Society Press.

[11] P. Neumann, S. Schlechtweg, and S. Carpendale. Arctrees: Visualizing relations in hierarchical data. In K. W. Brodlie, D. J. Duke, and K. I. Joy, editors, *Data Visualization 2005, Eurographics/IEEE VGTC Symposium on Visualization Symposium Proceedings*, pages 53–60, Aire-la-Ville, Switzerland, 2005. The Eurographics Association.

[12] J.-D. Fekete, D. Wand, N. Dang, A. Aris, and C. Plaisant. Overlaying graph links on treemaps. In *Poster Compendium of the IEEE Symposium on Information Visualization (INFOVIS'03), Los Alamitos, CA, USA*. IEEE, 2003.

[13] S. Zhao, M. J. McGuffin, and M. H. Chignell. Elastic hierarchies: Combining treemaps and node-link diagrams. In *Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'05), Minneapolis, MN, USA*. IEEE, 2005.

[14] J. J. van Wijk and H. van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *Proceedings of IEEE Symposium on Information Visualization INFOVIS'99*, pages 73–78, San Francisco, 1999.

[15] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[16] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: visualizing personal histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems CHI'96*, New York, NY, USA, 1996. ACM.

[17] N. Hirzalla, B. Falchuk, and A. Karmouch. A temporal model for interactive multimedia scenarios. *IEEE MultiMedia*, 2(3):24–31, 1995.

Visualizing Antenna Design Spaces

Kent Wittenburg, Tom Lanning, Darren Leigh, Kathy Ryall
Mitsubishi Electric Research Laboratories, Inc.
201 Broadway
Cambridge, MA 02139
USA
1-617-621-7500

wittenburg@merl.com, tom.lanning@att.net, leigh@merl.com, ryall@acm.org

ABSTRACT

This paper describes a long-term project exploring advanced visual interfaces for antenna design. MERL developed three successive prototypes that embodied an evolution towards larger scales and more concrete semantics for visualization of large sets of candidate designs and then winnowing them down. We experimented with multidimensional scaling and then collective line graphs before settling on linked scatterplots to visualize performance in a design space of up to 10 million antennas at a time. In the end, the scatterplot solution was most successful at balancing intelligibility with visualization of the space as a whole. The design allows for adding more 1D or 2D linked feature visualizations if needed, and it smoothly transitions to other "details on demand" views for final tweaking.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces - Graphical User Interfaces; I.3.6 [Computer Graphics]: Methodolgy and Techniques - Interaction Techniques; J2 [Computer Applications]: Physical Sciences and Engineering

General Terms

Measurement, Design, Human Factors

Keywords

Antenna Design, Information Visualization, Line Graphs, Human-Guided Search, Multivariate Visualization.

1. INTRODUCTION

This paper describes a long-term project exploring advanced visual interfaces for antenna design. Over five years ago, Mitsubishi Electric Research Laboratories (MERL) began a collaboration with Mitsubishi Electric Corp. Information Technology R&D Center, one of two main corporate R&D centers in Japan, and Mitsubishi Electric Corp. Electronics Systems Division, a business unit that designs antenna systems for commercial and Japanese government applications. End products range from satellites to aircraft communication systems to automobile collision avoidance radar to RFID readers and cell phones. Ongoing to this day, the collaboration was initially triggered by MERL prototypes in the area of Human-Guided Search, where we and others before us had explored how humans can solve a large and complex optimization task through visualization and interaction with an evolving search space [1][3].

This paper will focus on a subset of the prototypes in a discussion of lessons learned as we sought to bring our expertise as visual interface researchers to the problem of antenna design. The individual systems have been described in brief in other publications, mostly in IEEE antenna engineering conferences [12][13][14][17]. Details regarding our contribution to antenna engineering may be found there. Our purpose here is to examine the overall project from the perspective of information visualization and human-computer interaction methods. Our conclusions will be based on qualitative evaluations gleaned from our collaborators and users who carry out the actual tasks involved in antenna design in Japan. The typical pattern of our work on this project was comprised of an annual fiscal year cycle in which initial plans were made in the April timeframe, a prototype was created and shown to our customers and collaborators in the late summer, and then based on feedback from this prototype, a final prototype was created and delivered to Japan by the end of the fiscal year in March. Our systems were intended for internal company use by antenna design engineers, and our most recent tool is in use today.

The project began with an experiment in designing classic Yagi-Uda wire antennas, and over time progressed to more sophisticated types of phased array antennas. (For background information, see, e.g., Wikipedia entries on these topics.) We believe the lessons we have learned about interactive visualization has application not only to antenna design but to multivariate design problems generally. Not surprisingly, we found that it is important to apply different visualization methods to different sized sets to visualize the performance of antenna designs under consideration [2]; the larger task is to select one or perhaps a few candidate designs from what is initially a very large (perhaps infinite) set, trying to maximize certain performance goals while minimizing costs. In the end, we found it most valuable to visualize the performance of very large sets of candidate antenna designs through a pixel-based technique--parallel linked scatterplots of domain spe-

AVI'08, 28-30 May , 2008, Napoli, Italy

cific performance metrics. Through brushing and querying within parallel scatterplots, users can prune the space down to a hundred or so. At that point a table view of numerical performance values is useful in which standard sorting operations are possible. For viewing a single antenna from the table, we provide a visualization of the physical antenna array elements themselves and the corresponding 2D radiation patterns. Individual designs can be cloned and certain physical parameters tweaked and visually compared to the original to arrive at a final design.

On the way to the current system design we experimented with other information visualization techniques that we will discuss here, beginning with dimensionality reduction employed in our first prototype inspired by Human-Guided Search. In our second prototype we focused on line graphs for visualization and (soft) querying. Our third prototype, the main topic of this paper, illustrates the use of scatterplots in groups of three, each of which reveals a face of a cube representing a 3D design space of high-value performance measures.

2. RELATED WORK

The largest body of previous work relating visualization to the antenna design process is characterized by tools that visualize the radiation pattern of a simulated antenna in 3 or perhaps 4 dimensions [5]. Commercial and open source software is available today that in some cases is also compatible with MatLab and with the open source program NEC2 used for simulating the performance of the most common types of antennas (http://www.nec2.org). Visualizing the radiation patterns of individual antennas is undoubtedly important for the final stages in a design process. It is also valuable as a first step when the design process begins with a known example of a successful antenna design and then adapts it to the requirements of a new application.

However, as pointed out in [15], an antenna design process that begins only with known designs is highly limiting. The effects of even small variations on a myriad of parameters can have unexpected results--positive or negative--on the performance of an antenna. There is a huge number of specification variations possible for even basic antennas, and thus finding the best design may require looking into unexpected territory that can be reached only by considering millions or billions of combinations. Not surprisingly, genetic algorithms have thus become an important thread of research in antenna design [7][15]. The art and science of utilizing genetic algorithms for antenna design depends on careful design of an objective function for optimization.

However, as with all fully automatic systems for optimization, expert users may be left with the sense that they would like to look beneath the hood. As articulated by Spence and his collaborators, complex engineering design problems require experience and human judgment [2][19][20]. They have proposed methods to be used in digital circuit design, among other application domains, that make use of dynamic histograms and the "Prosection Matrix," a scatterplot-based visualisation of performance targets against input parameters. Our research shared the goal of providing visualization tools to yield insight in the design process. A goal is to allow the human to make connections between input parameters and performance results in a design space that can be only partially sampled. Perhaps one difference in the domain of antenna design from those that Spence et al. explored is that the performance goals for antenna design are devilishly hard to specify while performance characteristics for common types of antennas are relatively cheap and easy to simulate. The opposite seems to be true in the domain of digital circuit design [20].

3. METHODOLOGY

Conclusions in this paper are based on qualitative evaluations gathered from our collaborators and users of our system--antenna designers and engineers in Japan. We collected input for designs and feedback on prototypes in a number of ways, including attendance at an annual antenna meeting in Japan, e-mail exchanges throughout the year with our collaborators, face-to-face planning meetings both in Japan and the US, and interviews with other visitors to our lab with antenna domain expertise. The email exchanges were most often used for clarifications on points raised in person; visitor interviews usually confirmed feedback given at the annual meetings and during the face-to-face planning meetings. More detail on our annual meeting follows.

3.1 Annual Meetings

Each year one of the authors attended an annual antenna meeting in the late summer, presenting and demonstrating that year's prototype. The audience typically had 50-75 people, all antenna designers and engineers from different parts of our parent company's organization around Japan. After a technical presentation and brief demonstration of the prototype, the audience was free to ask questions about the approach and try out the prototype. Presentation and discussions were conducted mostly in English. As the prototypes evolved, the amount of interest and interaction with the audience grew. Once the third prototype was shown, there was great enthusiasm and requests to use the tool on the spot.

3.2 Observational Study

From the beginning of our collaboration, we had asked if we could meet directly with antenna designers, in particular to observe their current practice. While our direct collaborators were antenna engineers themselves, at this point in their careers, they were primarily scientists and managers and less involved with hands-on design.

After the presentation of our second prototype, we were eventually able arrange a visit to observe a designer in practice. One of the authors spent an afternoon with an antenna designer in his workplace in Japan. We discovered that his process was manually intensive. He would use batch processing to generate a large number of antenna designs and then use COTS or open-source tools to review the results. There was a lot of manual editing of files and typing at the command line to launch multiple instances of the same application. The designer usually reviewed several candidate antenna designs in parallel, opening perhaps twelve graphs at a time in a four-by-three configuration on the screen. He would then pick two at a time and do a more in-depth side-by-side comparison that included looking at statistics in a spreadsheet.

As he explained, the work proceeded with iteratively re-running simulations in batch mode overnight, having tweaked the input parameters based on what he had seen. It became clear that our customers were functionally and numerically oriented. We realized that while advanced visual methods would be helpful to the designers in evaluating candidate antennae, numerical precision had to be maintained, and the tools needed to be compatible with their current practices.

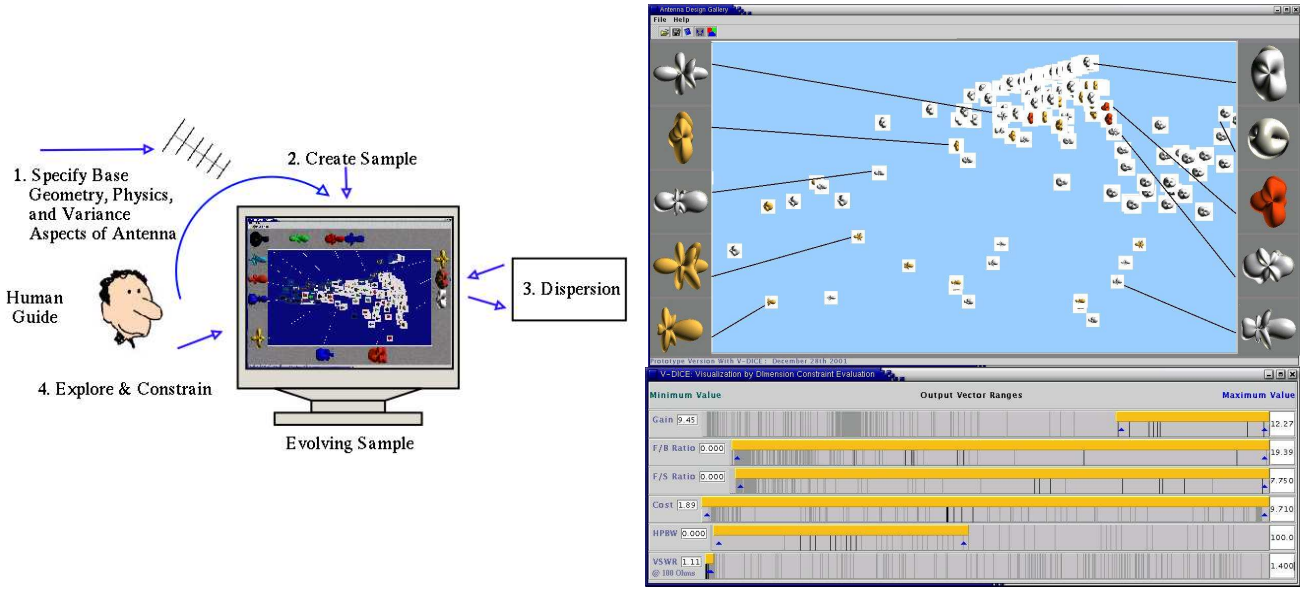


Figure 1: The design cycle shown at left was supported in our first prototype. On the right are the two primary visualization panes for the evolving candidate sample set: the top pane shows multidimensional scaling; the bottom, linear attribute widgets.

4. FIRST PROTOTYPE: ABSTRACT VISUALIZATION OF AN EVOLVING SEARCH SPACE

Our first prototype [17] applied human-guided search [1] [3] to the process of exploring and repeatedly refining a search through a very large space of possible designs for Yagi-Uda antennas. Yagi-Uda antennas are a classic type of directional wire antenna such as one sees with conventional rooftop TV antennas. Among the design parameters are the number, length, and spacing of the elements in the antenna array.

A primary focus of this initial stage of our project was to explore the idea of designing antennas through achieving maximal dispersion in a sampling of a very large design space that could not be enumerated exhaustively. Maximal dispersion here refers to a property of a set such that the set members are maximally distant from one another in a multidimensional space. Such an approach had been applied earlier in a tool for graphics and animation design [16] as well as scheduling [1]. An overview of the design process that the system was designed to support is shown at the left of Figure 1, described in [17]. After the user set some initial ranges over which the design parameters could be varied, the system would simulate a large set of antennas and then select candidates for human inspection based on their maximal dispersion in a multidimensional evaluation space. The multidimensional space was defined through weighted vectors of performance values. As the user honed in on a subset of the design space through interaction with the visualization tool, the system would again try to achieve maximal difference among potential candidates in the chosen subspace. At any time a user could inspect an individual antenna design candidate by visualizing its radiation pattern with an open source 3D visualization module (shown in thumbnails in the margins of the right side of Figure 1) or by inspecting the actual numbers of its performance simulation.

There were two methods deployed for visualizing the collective set of candidate antenna designs. As shown in the right side of Figure 1 in the main area of the top pane, dimensionality reduction was used in a layout in which icons representing individual candidate antennas were projected onto a 2D plane. Their relative distance from one another was intended to reveal distance in the design space, i.e., the Euclidean distance between weighted m-dimensional performance vectors. The layout algorithm, multidimensional scaling [11], attempted to find a positioning such that the distances on the 2D plane best correlated with the relative Euclidean distances in the performance space [16]. The overall goal was to reveal clusters of similar designs as well as outliers through this 2D layout.

The second method, shown in the bottom pane, visualized multidimensional performance attributes on widgets representing linear scales. Each of the parallel widgets revealed the distribution of a performance attribute across the entire set. The attribute values all fit a linear scale whose minimum and maximum corresponded to the actual performance range globally. The numbers were indicated in text boxes at the beginning and end of each attribute widget. Each candidate antenna was visible along each dimension as a vertical line. Selections could be controlled by sliders that would set minimum and maximum values. As subsets of candidates were selected by restricting the value range with one widget, the candidates would be visually highlighted in the other widgets and in the 2D layout above. The attribute widgets are a basic form of one-dimensional data visualization sliders [4] that incorporate brushing techniques [21].

What were the lessons learned from this first prototype? First, multidimensional scaling as a visualization technique was not well received by our users. The reduction of a multidimensional performance space to 2D did not help the designers understand what they were looking at. We were told that they couldn't make sense of this visualization, that it would be better to have a series of 2D views with axes whose semantics were clear. The parallel attribute

visualization sliders were received more positively since they indeed did have an understandable semantics.

Second, our notion of dispersion as a search mechanism got a lukewarm reception, probably because, again, its effects were not transparent to these antenna experts. We came to the conclusion that as the project moved to its next phase, it would be best to decouple the exploration of the design space from the visualization and filtering of the results. We would postpone our original goal of supporting an end-to-end human-guided search tool and instead focus on interactive visualization and winnowing of a large design set. We introduced new modularity in the system that allowed candidate sets to be generated independently by our clients or ourselves. The sets could then be loaded into a tool for visualization and filtering.

5. SECOND PROTOTYPE: VISUALIZATION AND FILTERING THROUGH QUERY LINES

The next phase of our project took on the problem of designing certain types of phased array antennas [6]. Another piece of feedback we had gotten from the first round of the project was that Yagi-Uda antenna design was not enough of a challenge for this group of antenna experts. As with Yagi-Uda antennas, phased array antennas are also directional. They are distinguished by varying the phases of the signals feeding the elements of the array such that a desired radiation pattern is achieved. Our approach was to utilize 2D line graphs both for visualizing the performance of candidate designs and for querying and filtering the candidates [13][18]. The most important of these graphs, examples of which can be seen in Figure 2 (b-e), are the radiation patterns. The x-axis is observation angles (degrees) and the y-axis is the array factor directivity. In general, designers are looking for a gain peak in the center with minimal energy in the off angles. At least for linear arrays, a 2D radiation graph is a good indicator of the primary performance design goal.

As mentioned, in this phase of the project we assumed that a set of candidate designs would be generated independently. For testing purposes we were able to generate an exhaustive set of possible candidates for phased array antennas with some simplifying assumptions. We assumed that the arrays were uniformly linear and looked at variants of phase-only synthesis. The generator took as input the number of elements in the array and a set of quantized values for phase and amplitude coefficients. It then exhaustively enumerated all possible combinations of excitation parameters to compute a set of candidate designs. We were able to load in on the order of 10,000 design variants at a time.

From an information visualization perspective, naively plotting the radiation patterns of all the generated designs in a large candidate set would result in an undifferentiatable blob (Figure 2(b)). Line graphs were not a solution for visualising the space as a whole. However, our hypothesis was that it would be desirable for designers to explore the design space by filtering with queries that could be created directly with and on 2D line plots. Visual querying with 2D line graphs has been tackled before [8], but it is generally not straightforward to specify 2D constraints on line graphs. Approximate matching is even more of a requirement than with conventional Boolean queries since lines are highly unlikely to find exact matches. As with querying generally, result lists characterized only by hard matches do not reveal anything about the set of candidates that almost matched and very little about the space of solutions as a whole. R. Spence has articulated the need for information visualization systems to reveal *sensitivity information* that can help guide users to explore parts of the design space that they hadn't previously considered [19].

The main contribution of this work was in developing a set of approximate 2D graph-based query methods that could reveal sensitivity information. We will touch on only the main features of the system here. Figure 2(a) shows a screenshot of the overall system. Three types of linked performance graphs are shown in the main screen, the most important of which is the radiation pattern, but any of the graphs can be the basis of querying. In the embedded window at the bottom of Figure 2(a) is a set of results of a previous query in which the list on the left represents hard matches for the expressed constraints and the list on the right represents an ordering of soft matches, i.e., matches that are close to the constraints expressed by the query lines but do not fall strictly within them. Figure 2 (c-e) shows examples of different types of query specifications and their resulting matches. Figure 2(c) shows query lines that represent minimum and maximum constraints over the (x,y) plots shown and a set of "hard match" patterns that fall within those constraints. Figure 2(d) shows two soft matches, i.e., patterns that do not fully meet the specifications of the min./max query lines but are nevertheless close to the constraints. Figure 2(e) shows a different type of query line--a goal or preference. The contribution of a goal or preference query line to the results returned by the system is to sort the hard and soft matches on the basis of similarity to this 2D preference pattern. If no other constraints are given, a goal query line amounts to a query by example and all results are soft matches.

From one perspective, the second round of prototyping for antenna design represented an extension of the 1-D attribute visualization widgets of the first round to a 2D approach. Selecting and filtering with 1D attributes is straightforward. Selecting and filtering with 2D line patterns is a harder problem that required inventing these new approximate matching methods and interactions.

The second prototype drew more interest than the previous one -- both because of the more realistic problem domain and the more intuitive visualization techniques; there were many requests to try different queries using the prototype. While there was some indication that the query mechanism itself might be too complicated for engineers to use, the bigger question (and excitement) was whether the approach would scale to more complicated antennas and problems of larger size. Given the success with this small (constrained, yet realistic) problem specification, we were again directed to move onto a more challenging and larger problem task. It seemed clear to us that a different approach to visualizing larger sets globally was needed.

6. ROUND THREE: A PIXEL-BASED APPROACH WITH LINKED SCATTERPLOTS

In our next round [14], we were given a particular optimization problem to focus on, which was as follows. Our goal was to maximize performance of sparse linear array antennas with uniform excitation, i.e., the assumption of linear spacing of the phased array elements in our previous round was relaxed but the constraint of uniform signal excitation across the array elements was fixed. The number of array elements and their spacing were the variables to explore. With a supercomputer cluster, we were able

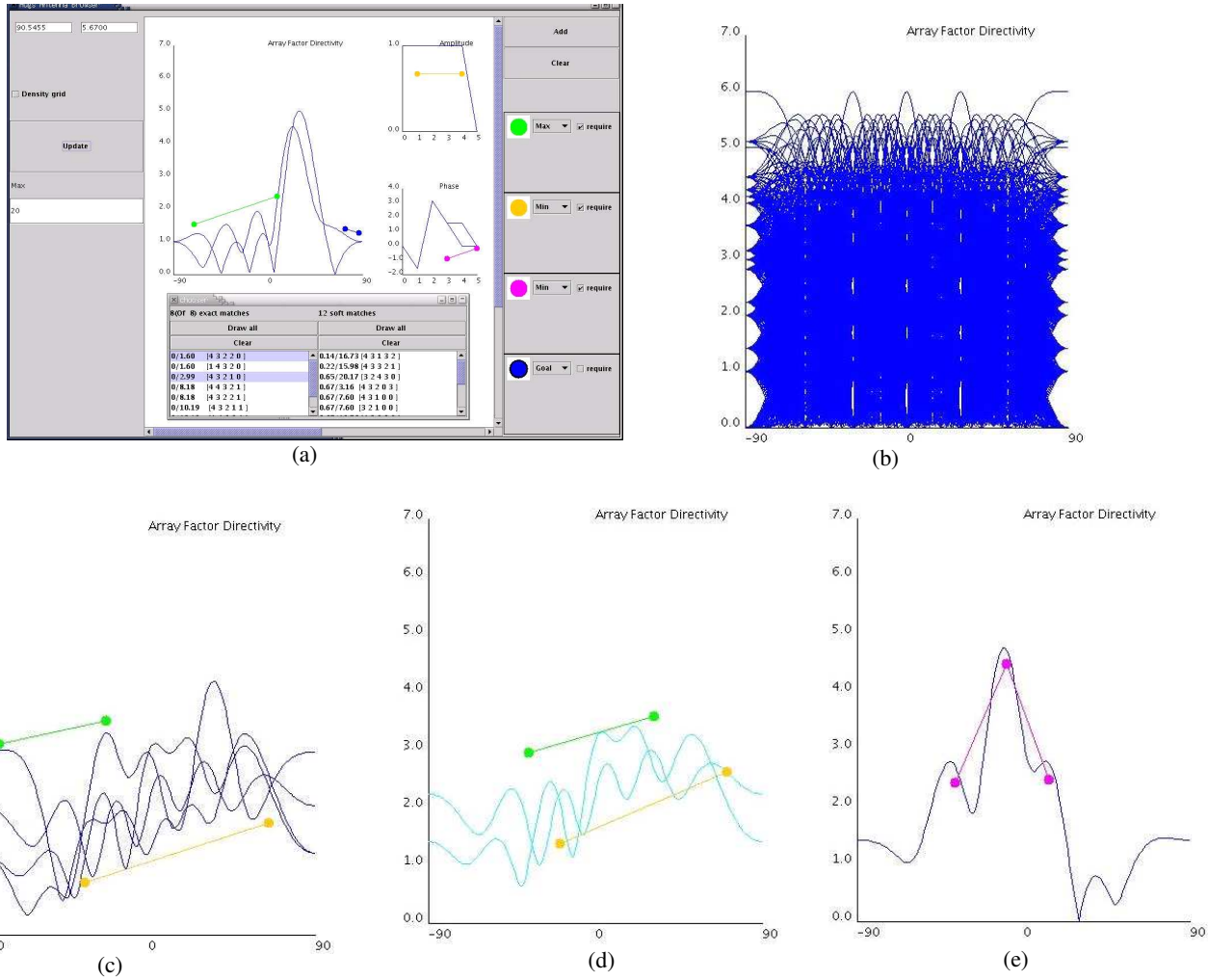


Figure 2: The QueryLines System: (a) a snapshot of the overall system, (b) a large set of graphs, (c) min and max hard constraints and several matching results, (d) two soft matches that do not fall within the hard constraints, (e) a goal query and a result that is the closest match.

to generate and simulate on the order of 1 billion variants of such antennas.

Our goal for the visualization tool was to handle up to 1 million candidates at a time and to develop interaction methods for exploring and filtering that would respond almost instantaneously. It should come as no surprise that our solution for visualization of such a large set of elements utilized pixel-based techniques [9]. Although the most important antenna performance design goal is best visualized as a line graph (or 3D plot) of the radiation pattern; neither of these techniques are appropriate for viewing in large sets. It was more efficient for machine computation and human interaction to use simple performance numbers that would characterize the radiation pattern indirectly. These were as follows:

• The width of the main lobe (full-width half maximum).

• The gain of the highest side lobe.

• The angle of the highest side lobe.

These performance measures are suitable for visualization with standard 1-D widgets, but we explored a variation in which these three dimensions defined the dimensions of a cube. We could then plot three faces of the cube as linked scatterplots; 1 million design variants are shown in Figure 3. The main interaction method is to sweep out selections of pixels in any of the scatterplots, which will be painted in all of the scatterplots. At any time a user can reduce the set (zoom in) by filtering to the current selection.

In order to meet the requirement for quick response time, we came up with a code design that utilized the resolution on the screen to organize the data. Each time there is a screen resize, a one-time process sorts the data into bins, one for each pixel. When rendering, the pixel is lit if it contains any data. The result is that a rendering of a scatterplot with one unit per pixel, can happen within a second on most standard desktop or laptop computers.

The striking striations visible in Figure 3 are an example of unexpected results that may be revealed by a visualization tool such as ours. The antenna experts we consulted are not sure why these

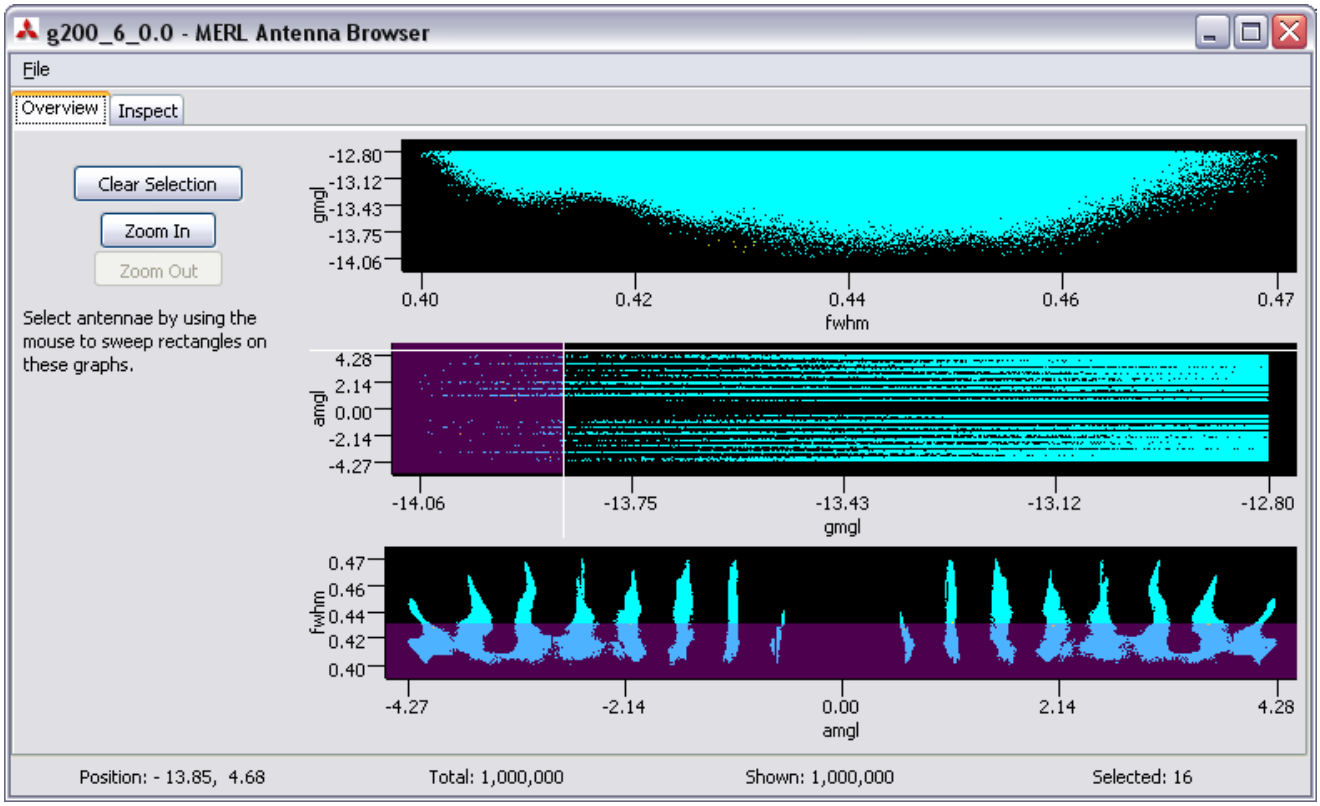


Figure 3: The overview pane of the latest version of our visualization tool (shown in small window size for readability). Three linked scatterplots are views into a 3D space of performance measures. The current selections are updated in all views.

patterns emerged, indicating a non-uniform distribution of the shape of the main lobe across the angles of the highest side lobes. All of the antenna variants in this particular example set had the same number of antenna elements in the array. But clearly the possible positioning of these elements left gaps in the distribution of main lobe energy with respect to highest side lobe angles.

An example based on observing how an antenna expert used the tool follows. The expert first swept out the lower region of the bottom scatterplot, the results of which are (subtly) visible in Figure 3. These antennas would have the narrowest main lobes (indicated on the y-axis), a measure of high directionality irrespective of the angle of the highest side lobe (indicated by the x-axis). The expert interactively played with the maximum setting on the y-axis in order that some selections appeared at the left side of the middle scatterplot, an area of sparse distribution. This area contained antennas whose highest side lobe has low gain, irrespective of its angle. Again, in general, designers are looking for high energy in the main lobe with minimal energy in the side lobes. Then the expert swept out the rectangle in the left area in this middle pane. This further constrained the selection set to those antennas with the desired properties. From a million antennas, the expert was able quickly to narrow down the set to a size of 16 or so, which he then looked at more closely in the Inspect pane, where line graphs of radiation patterns and the performance numbers themselves were visible.

A screen shot of the Inspect pane is shown in Figure 4. The upper part of the pane contains a table of the selected antenna design candidates. The columns contain the numbers for the position of each included element as well as the three performance measures mentioned above. The table rows may be sorted on the basis of any of the columns in the usual way. Such a table method is useful and usable when there are no more than a few hundred design variants under consideration. A common interaction pattern we noted is for users to sort the antenna units along one column and then hold down an arrow key to traverse the list from top to bottom, causing a rapid serial visual presentation (RSVP) of the 2D gain pattern [19].

For viewing details of an individual design, a user may select a row in the table, an example of which is visible in the lower part of Figure 4. The graphic in the middle of the pane represents the physical position of the array elements and the line graph at the bottom is the radiation pattern. An individual antenna (shown in blue) may be copied (shown in yellow). The position of certain array elements in the copy may be interactively moved and the radiation pattern of the copy compared graphically to the original.

This visualization tool contributed to the finding published in [14] that it was possible to achieve essentially the optimal performance of uniformly spaced arrays with fewer elements (thus less cost) spaced non-uniformly in certain configurations. The tool can of course be used to explore other kinds of design issues as long as the data basically conforms to the patterns shown here.

With this third prototype we were pleasantly surprised to have several audience members at our annual design meeting ask if they could try the tool on the spot; at previous meetings they most

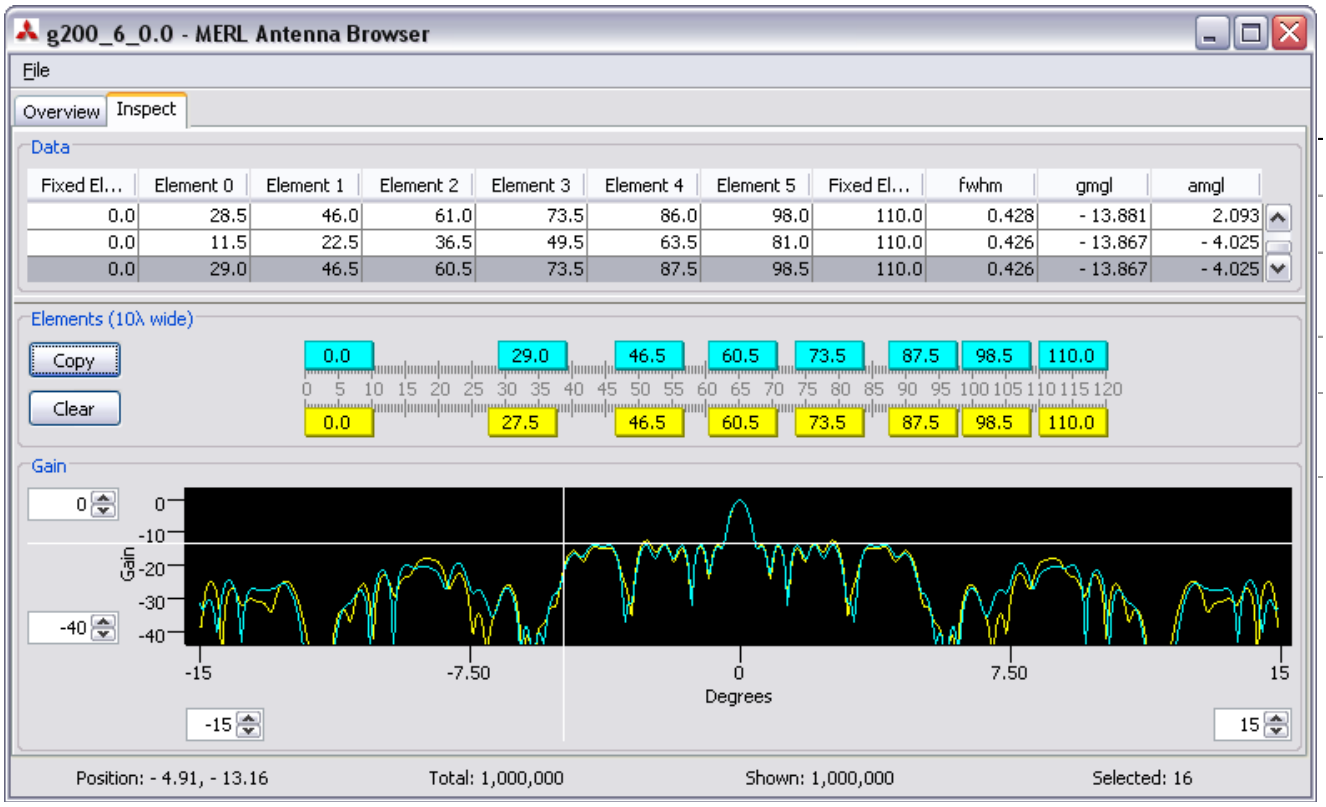


Figure 4: The inspect pane of the latest version of our antenna visualization tool (shown in small window size for readability). The table at top shows performance numbers for the selected antennas; the graphic in the middle shows individual elements and their positioning; the bottom shows a radiation pattern and its copy, which may be tweaked.

often made their request verbally and the presenter (one of the authors) would run a query in the prototype. There was also much discussion and side conversation in Japanese. This prototype was extended and deployed later that year and is in use today.

7. LIMITATIONS OF THIS STUDY

We see two main scientific limitations to our study. First, due to our clients' request we changed the problem specification (task) every year. As the antenna types shifted over the course of the study we were not easily able to compare prototypes head-to-head on the same set of antenna designs. Thus it was difficult to tease apart which changes in our design were due to the change in task and which were a result of improvements to the interaction and UI design. In a more controlled study we would have fixed the task, or perhaps re-run some of the later prototypes on the earlier data sets. Unfortunately, neither of these approaches was feasible logistically. We do believe that the visualizations used in all three prototypes would be appropriate for all antenna types with the exception of the antenna element visualization in prototype three, which would require modification for Yagi-Uda antennas.

A second limitation of our study is that our evaluation methodology was flawed. In a real-world setting it is difficult to do a comprehensive user study. In a more controlled setting, it is easier to design and evaluate systems and recruit participants. We faced some additional challenges due to geographical distance as well as differences in language and domain expertise (Antenna Engineers vs. Computer Scientists). In the end, we believe it is the latter that had the largest impact. It was sometimes difficult to get buy-in on our design process and requests for feedback. For example, convincing our colleagues that we would benefit from directly meeting with and observing practicing antenna designers was a lengthy process. In a perfect world, we would increase the scientific rigor of our work by introducing surveys, increasing sample sizes, and retesting across the prototypes with a constant dataset.

8. CONCLUSION

In this paper, we have presented a longitudinal design study that resulted in what we believe to be a successful interactive visualization tool for antenna designers in a Japanese industrial setting. The primary visualization methods employed were parallel scatterplots, sortable tables, and 2d line graphs--not novel in themselves, but we believe novel in their application to this domain.

Dimensionality reduction, a popular technique among visualization researchers, was not successful in the eyes of our users. In order for a global visualization of a very large set to make sense, it is important that the semantics of the visualization be concrete. One can see from the example of use described in Section 6 that an expert would know how to filter a large space and understand the patterns if the data is presented with dimensions easily related to the task at hand. Dimensionality reduction, useful in many ways, may not be so useful if the first concern is concrete performance numbers.

Our experience with QueryLines showed that, although it is attractive to consider querying directly with line graphs representing radiation plots, there is complexity in specifying such queries as a set of constraints. Our conclusion was also that line graphs would not scale well in an overview mode. In retrospect, we should note that the methods in Line Graph Explorer [10], which we were not aware of at the time of this work, do provide some ability to scale

up to larger sets. However, in order to achieve the scale of, say, 10 million designs in a single view, each graph displayed using Line Graph Explorer would have to represent an aggregation on the order of 10,000 radiation patterns since one line of the display is needed per graph and there are order 1000 horizontal pixel rows available on desktop displays. We don't know whether the computational demands of such an approach would be able to offer suitably rapid response or how it might be received by antenna designers, but it may be worth a look.

9. FUTURE WORK

Since the development of the visualization tool described here, the project has returned to the problem of algorithms for enumerating the search space [12]. We have also extended the Inspect pane of the visualization tool to handle circular antenna arrays. In the future, we imagine that it may be useful to consider radiation plots in 3D rather than 2D as more complex types of antennas come within the scope of the tool. In such cases, we suppose that again we will need to come up with numerical functions that can be visualized with scatterplots and other types of easily understandable parallel widgets to handle large design spaces.

Also, one question we are left with is how our use of three 2D scatterplots to represent a 3D design space might compare to actually rendering the scatterplot itself in 3D. (See, e.g., [21].) We will have to leave the answer to that question to future research.

10. ACKNOWLEDGMENTS

A large number of people have contributed to the work described in this paper. They include, among our Japan colleagues, I. Chiba, Y. Hara, K. Hirata, Y. Konishi, S. Makino, H. Miyashita, and T. Sakura. On the MERL side, besides the authors, contributors include C. Lee, N. Lesh, J. Marks, and A. Quigley.

11. REFERENCES

[1] Anderson, D., Anderson, E., Lesh, N.B., Marks, J.W., Mirtich, B., Ratajczack, D., and Ryall, K. 2000. Human-Guided Simple Search. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence (August 2000), 209-216.

[2] Apperley, M., Spence, R., and Wittenburg, K. 2001. Selecting One from Many: The Development of a Scalable Visualization Tool. In Proceedings of IEEE Symposium on Human-Centric Computing Languages and Environments (Stresa, Italy, Sept. 5-7, 2001). HCC '01. IEEE Computer Society. 366-372.

[3] Colgan, C., Spence, R., and Rankin, P. 1995. The cockpit metaphor. Behavior and Information Technology 14, 4, 251-263.

[4] Eick, S. Data Visualization Sliders. 1994. In Proceedings of ACM Symposium on User Interface Software and Technology (Marina Del Rey, California, USA, Nov. 2-4, 1994). UIST '94. ACM Press, New York, NY, 119-120.

[5] Chakrabarti, S., Wong, J. C., Gogineni, S. and Cho, S. 1990. Visualizing Radiation Patterns of Antennas. IEEE Computer Graphics and Applications (January 1990), 41-49.

[6] Hansen, R. C. ed., 1985. Microwave Scanning Antennas. Peninsula Publishing, Los Altos, CA, 1985.

[7] Haupt, R. L., and Werner, D. H. 2007. Genetic Algorithms in Electromagnetics. Wiley-IEEE Press, Hoboken, NJ, 2007.

[8] Hochheiser, H., Shneiderman, B. 2004. Dynamic Query Tools for Time Series Data Sets, Timebox Widgets for Interactive Exploration. Information Visualization 3, 1 (March 2004), 1-18.

[9] Keim, D. A. 2000. Designing Pixel-Oriented Visualization Techniques: Theory and Applications. IEEE Transactions on Visualization and Computer Graphics 6, 1 (January-March 2000), 1-20.

[10] Kincaid, R., and Lam, H. 2006. Line Graph Explorer: Scalable Display of Line Graphs Using Focus+Context. In Proceedings of Advanced Visual Interfaces (Venice, Italy, May 23-26, 2006). AVI 06. ACM, 404-411.

[11] Kruskal, J. B., and Wish, M. 1978. Multidimensional Scaling. Sage Publications, 1978.

[12] Lee, C., Leigh, D., Ryall, K., Miyashita, H., and Hirata, K. 2006. Very Fast Subarray Position Calculation for Minimizing Sidelobes in Sparse Linear Phased Arrays. In Proceedings of European Conference on Antennas and Propagation (Nice, France, November 6-10, 2006). EuCAP 2006. ESA Publications Division, Noordwijk, The Netherlands.

[13] Leigh, D., Lanning, T., Lesh, N., and Ryall, K. 2004. Exhaustive Generation and Visual Browsing for Radiation Patterns of Linear Array Antennas. In Proceedings of International Symposium on Antennas and Propagation (Sendai, Japan, August 17-24, 2004). ISAP.

[14] Leigh, D., Ryall, K., Lanning, T., Lesh, N., Miyashita, H., Hirata, K., Hara, Y., and Sakura, T. 2005. Sidelobe Minimization of Uniformly-Excited Sparse Linear Arrays using Exhaustive Search and Visual Browsing. In Proceedings of IEEE Antennas and Propagation Society International Symposium (Washington, D.C., July 3-8, 2005), Vol. 1B, IEEE, 763-766.

[15] D. S. Linden, D. S., and E. E. Altshuler,E. E. 1999. Evolving Wire Antennas Using Genetic Algorithms: A Review. In Proceedings of the First NASA/DoD Workshop on Evolvable Hardware(Pasadena, CA, USA, July 19-21, 1999). IEEE, 225-232.

[16] Marks, J., et al. 1997. Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. In Proceedings of International Conference on Computer Graphics and Interactive Techniques (Los Angeles, CA, USA, August 1997). SIGGRAPH '97. ACM, 389-400.

[17] Quigley, A., Leigh, D. L., Lesh, N. B., Marks, J. W., Ryall, K., and Wittenburg, K. B. 2002. Semi-Automatic Antenna Design via Sampling and Visualization. In Proceedings of IEEE Antennas and Propagation Society International Symposium (San Antonio, TX, USA, June 16-21, 2002), Vol. 2, IEEE, 342-345.

[18] Ryall, K., N. Lesh, N., T. Lanning, T., D. Leigh, D. H. Miyashita, H., S. Makino, S. 2005. QueryLines: Approximate Query for Visual Browsing. In ACM Conference on Human Factors in Computing Systems (Portland, OR, USA, April 2-7, 2005). CHI 2005. Extended Abstracts, ACM, 1765-1768.

[19] Spence, R. 2007. Information Visualization: Design for Interaction, Second Edition, ACM Press, 2007.

[20] Spence, R. 1999. The Facilitation of Insight for Analog Design. IEEE Transactions on Circuits and Systems--II: Analog and Digital Signal Processing, 46, 5 (May 1999), 540-548.

[21] Swayne, D. F., Cook, D., and Buja, A. 1998. XGobi: Interactive Dynamic Data Visualization in the X Window System. Journal of Computational and Graphical Statistics, 7, 1 (March 1998), 113-130.

The In-Context Slider: A Fluid Interface Component for Visualization and Adjustment of Values while Authoring

Andrew Webb and Andruid Kerne

Interface Ecology Lab, Computer Science Dept.,
Texas A&M University, College Station, TX 77843,USA

awebb, andruid@cs.tamu.edu

ABSTRACT

As information environments grow in complexity, we yearn for simple interfaces that streamline human cognition and effort. Users need to perform complex operations on thousands of objects. Human attention and available screen real estate are constrained. We develop a new fluid interface component for the visualization and adjustment of values while authoring, the *In-Context Slider*, which reduces physical effort and demand on attention by using fluid mouse gestures and in-context interaction. We hypothesize that such an interface will make adjusting values easier for the user. We evaluated the In-Context Slider as an affordance for adjusting values of interest in text and images, compared with a more typical interface. Participants performed faster with the In-Context Slider. They found the new interface easier to use and more natural for expressing interest. We then integrated the In-Context Slider in the information composition platform, combinFormation. Participants experienced the In-Context Slider as easier to use while developing collections to answer open-ended information discovery questions. This research is relevant for many applications in which users provide ratings, such as recommender systems, as well as for others in which users' adjustment of values on concurrently displayed objects is integrated with extensive interactive functionality.

Categories and Subject Descriptors

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General Terms

Design**,** Human Factors, Experimentation

Keywords

In-Context Slider, interest expression, in-context interface, fluid gestures, interaction design

1. INTRODUCTION

As information environments grow in complexity, we yearn for simple interfaces that streamline human cognition and effort. Interactive spaces contain thousands of objects. Users need to perform complex operations on individual objects and subsets. The limits of human attention and available screen real estate constrain the design solution space. We need to discover new interface components, which recognize and take into account the context of the user's situated task. In-context interfaces address these design issues by providing affordances in-place. Activation is *transitory*, that is, they only appear when necessary and requested. Clear mappings are based on fluid gestures. Activation rules are based on the user's context.

The present research is concerned with contextualized visualization and adjustment of a one-dimensional value. The task context integrates authoring and getting recommendations. It is conducted either in a space of many graphical objects, or in a text editor. With each graphical object or word, a value is associated. The set of these values constitutes the profile of user interests. Eliciting the user's input on ratings is sufficiently difficult that it proves to be a barrier of participation in many recommender systems [3, 11].

We redefine providing ratings in a human-centered way, as "expressing interest." We develop a fluid in-context interface for interest expression, which can be tightly integrated into other user tasks, such as authoring and editing of textual and visual information. Our hypothesis is that expressive interaction will be increased by reducing user effort and increasing feedback.

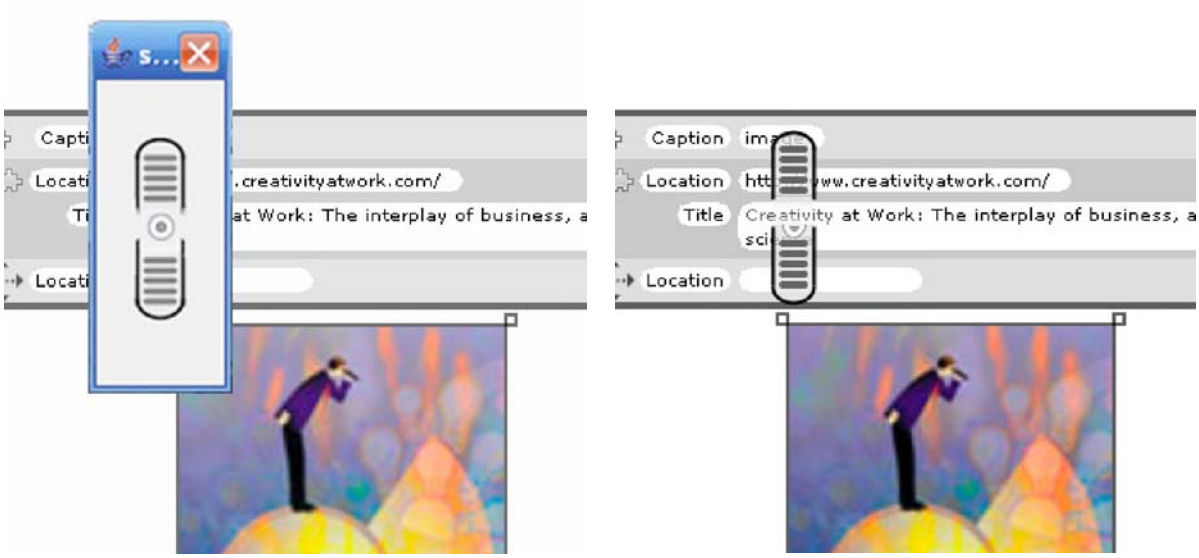
A typical interface design for adjusting a value associated with a graphic object or word is to display an input interface (e.g. slider or text field) inside a pop-up window that is often activated by selecting from a menu or sidebar. The pop-up can occlude the user's context (see Figure 1) or appear outside the current point of focus. Some alternative interface design methods dedicate real estate. Others require the user to press hot keys that lack visibility.

We develop the In-Context Slider, a fluid transitory affordance for visualization and adjustment of values. We describe the role of the In-Context Slider in the integration of interest expression with authoring. We present an evaluation based on text-editing and image ranking tasks. We introduce the combinFormation mixed-initiative information composition platform [9], and how the In-Context Slider fulfills interest expression needs within that platform. The platform plays a key role in information discovery tasks performed by 1000 undergraduate students annually. The student users are novices, with no particular computing background. We present user experiences of expressing interest to represent collections with composition. We then review related work and conclude by deriving implications of this research.

2. THE IN-CONTEXT SLIDER

The In-Context Slider is a user interface component that recognizes aspects of the user's situated task to provide transitory affordances in proximity to the focus object to support the adjustment of a value through fluid movements. We arrived at this solution through a human-centered iterative design process. The goal of the design process was to create a better interface for

AVI'08, 28-30 May , 2008, Napoli, Italy



(a) **(b)**

Figure 1: Popup vs. In-context Interfaces

interest expression in combinFormation while not disrupting the user's experience of authoring compositions.

2.1 Layered Activation

What makes an in-context interface fluid is the ability to activate layers of the interface at the point of focus, in the midst of an interactive space, through simple gestures. Clear affordances are required to cue the user about how to trigger each successive layer. We call these affordances *activators*. An activator provides fluid transitions between the layers of interaction. Activation affordances must be designed so that their presence minimally disrupts other constituent functionalities of the context. Through layered activation, the affordance's capability and screen presence grow gradually, with the user's attention. The possibilities for interaction are always visually clear. The affordance for each successive layer of activation is positioned in-context, relative to the positions of the preceding activators. In order to prevent unwanted activations, a delay may be necessary before visualizing each layered activator.

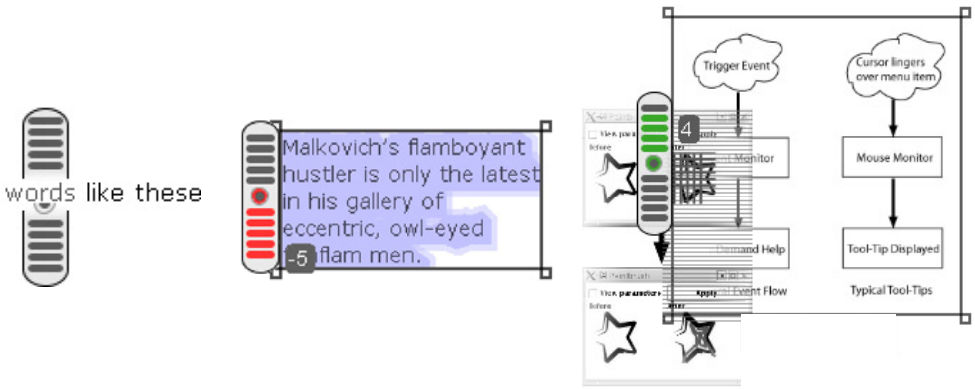
An In-Context Slider has three layers of activation. Each layer is activated by the mouse-over gesture. The activator in the initial layer, layer 0, is an object already in the interactive space, whose functionality is augmented by the In-Context Slider. As an activator, this object receives new functionality as an affordance for accessing the next layer of activation. In the present research, a layer 0 activator is an image, a word in a passage of text, or a whole passage of text (see Figure 3). While a value is visualized, disruption of context is minimized. Thus, editable text remains editable, while each word is augmented to enable interest expression. Since a layer 0 activator has other pre-existing functionality, mousing over it does not necessarily mean the user desires to activate an In-Context Slider. The user could be simply passing over the activator to interact with something else. To confirm the user's intention to interact, a small adjustable delay (defaulted to 550ms) is applied before visualizing the layer 1 activator. Interaction with the pre-existing functionality of a layer 0 activator, such as clicking to type a character amidst text, or click and drag to highlight, results in the immediate removal of the activator. Pulling the mouse off the layer 0 activator, away from the layer 1 activator, also removes both activators.

In the In-Context Slider, the layer 1 activator is an affordance called the *navel*. The navel is a small circular object that is designed to be differentiable from, yet not disruptive of its surroundings (see Figure 2b), and to form the center of the subsequent layer 2 In-Context Slider body (see Figure 2c). The location for the navel places it in close proximity to layer 0, while avoiding occlusion of visual features that are otherwise necessary for legibility and usability of the context. The navel comes in two distinct visual forms to accommodate different layer 0 activators. For images and passages of text, the navel is a full circle (see Figure 3b,c). For text, the navel is the bottom half of the full circle version (see Figure 3a). The horizontal edge forming the top of the half circle navel fits visually with the base line of text. As well, text is normally formed by a horizontal sequence of words across vertical arrangements of lines. The gap between the lines provides an appropriate unused space to place the navel. To avoid interference between text editing and activation of navels for text passages, the navel for a text passage is placed directly to the left side of the text (see Figure 3b).

Layer 2 is visualized by the body of the In-Context Slider, which expands vertically outward from the navel. The slider body contains a set of vertically stacked horizontal bars representing the possible values for the slider. The horizontal bars are split across the navel, so that bars representing positive values appear above the navel and bars representing negative values appear below the navel (see Figure 2c). As few as 3 pixels can be used for each bar and in between space. The total number of bars can

(a) **(b)** **(c)**

Figure 2: Example of Activation Layers, (a) layer 0, (b) layer 0 and 1, (c) all layers



(a) (b) (c)

Figure 3: Examples with all three layers of activation: (a) single word, (b) passage of text, (c) image.

be adjusted. The default number is ten, five positive and five negative. A slight translucence is applied to the slider body in the area surrounding the bars. This translucence allows visual objects occluded by the slider body to remain partially visible. As an in-context interface designed to minimize the cognitive effort on the user, keeping the focal point of the interactive space optimally visible is an important task. The translucence also gives the slider body a lighter than air quality, which is representative of its transitory nature as a layer of activation. Mousing off the slider body but onto the layer 0 activator removes the slider body and leaves the navel. Mousing off the slider body and off the layer 0 activator in the process removes both the slider body and the navel.

2.2 Visualizing Values

The present research applies Norman's prescription, to "make things visible" [13]. The current value of an In-Context Slider is visualized by highlighting, with hue, the navel and the bars in the slider body that represent the value (see Figure 4b, c). Color is a pre-attentive visual feature [12]. In our vision, hue is processed early and in parallel requiring no attention. This cognitive property of color makes it well-suited for visualizing value in an In-Context Slider. With the In-Context Slider body, positive values are represented in green by default, with negative values in red. The colors were chosen based on the stop light metaphor. The neutral value is represented by gray. Since gray is an entirely unsaturated color, the saturation of the color is used to represent the intensity (distance from zero) of the value. In other words, a positive value of five has a much higher saturation than a positive value of one. A value of five will appear greener than a value of one. The same applies to negative values with the color red. To handle physiological (e.g. color blindness) and cultural issues, the hues for positive and negative can be reconfigured.

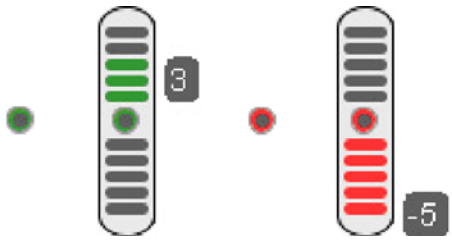
The navel and layer 0 activator provide mechanisms for visualizing the value of an In-Context Slider even when the slider is not activated to the third level. Inside the navel is a light gray ring that changes color to match the current value (see Figure 4a, c). This allows the In-Context Slider, while not fully expanded, to visualize whether the current value is positive, negative, or neutral and provide some indication of the intensity of that value. The layer 0 activator of an In-Context slider can also have its appearance adjusted to reflect the current value. For example if an activator is a textual word, the color of the word will change to match the color for its assigned value. This provides quick feedback in context to the user about the currently assigned value.

2.3 Interacting to Change a Value

To change the value of an In-Context Slider, the user moves the mouse cursor up or down over the layer 2 slider body. We chose move instead of drag to minimize effort. All bars from the navel (center) to the current mouse position are highlighted with the appropriate color (see Figure 4). A small popup textbox with the current visualized value appears to the side of the slider vertically matching the current mouse position. Once the desired value is visualized, the user clicks the left mouse button to set the value, and, depending on whether the mouse cursor is currently over the activator or not, the In-Context Slider either reverts to the collapsed navel-only form or disappears entirely. The user can choose not to change the value by simply moving the mouse off the In-Context Slider without clicking. If, after moving the mouse off, the mouse cursor is still positioned over the layer 0 activator, the In-Context Slider layer 1 remains in collapsed navel-only form. If the mouse cursor ends off the layer 0 activator, the In-Context Slider is fully deactivated, removing it entirely (both layer 1 and layer 2) from the screen.

2.4 Activating Multiple Sliders

In the iterative design process, it was discovered that a user might wish to assign the same value to multiple objects at one time. To accommodate this action, multiple layer 0 activators can be activated at once. Layer 1 navels remain visible for each activated object through the course of the activation sequence. The process of multi-activate is similar to that of marking a route on a map through a set of waypoints. The waypoints are the navels. The user enacts multi-activate by holding the left mouse button down while over the navel and dragging the mouse cursor. A fuchsia-colored line is drawn from the center of the navel to the current



(a) (b) (c) (d)

Figure 4: Visualizing an In-Context Slider value: (a) collapsed positive value, (b) expanded positive vaue, (c) collapsed negative value, (d) expanded negative value.

mouse cursor position. While dragging, the user can mouse over another layer 0 activator, causing another navel to appear. In this case, there is no delay for showing the navel, since the intention to activate additional In-Context Sliders is clear from context. If the user ends drag by releasing the left mouse button while over the new navel, the fuchsia line disappears and a persistent gray line is drawn connecting the center of the two navels, just as a connecting line marks a route segment between two waypoints on a map. Since the user is now over a navel, the slider body is activated. The user can continue activating In-Context Sliders by repeating the same process from one navel to the next. After activating the desired sliders, the user changes the value of the last activated slider. This changes the value for the others. Multi-activation is cleared when the user either sets a value or deactivates an activated In-Context Slider.

When activating multiple sliders, it is not required to end the mouse drag on a navel. If the mouse drag is ended anywhere on the layer 0 activator, the slider will be activated, moving the mouse cursor to the navel center, displaying the slider body, and drawing the persistent gray line between the navels. Multiple-activation doesn't have to start at the navel. It can also start from the slider body. The process is the same as when starting from the navel (hold left mouse button and drag). The difference is that when activating another slider (by ending drag), the current value for the newly activated slider is set to the value of the previously activated slider. In other words, by starting multi-activation in a slider body, the current value is propagated to each slider activated afterwards in the activation sequence. This multi-activation sequence provides flexibility in assigning the same value to multiple objects. If at any point in the process the user decides a different value is appropriate, that value can easily be assigned from the current slider, and the sequence can continue.

3. EVALUATION

3.1 Participants

Forty-three student volunteers participated in the experiment. Undergraduate members of the "psychology subjects pool" fulfilled a requirement of their introductory psychology course by participating. Concurrently offered sections of the course had a total enrollment of more than 1000 students. The subjects represent a spectrum of undergraduates, with no focus on computer or information science majors. The experimenters were not personally familiar with the participants.

3.2 Method

Two tasks were designed to evaluate the In-Context Slider in comparison to a Typical Dialog Box Slider interface for interest expression. The Typical Dialog Box Slider interface consisted of a draggable slider with a knob inside a dialog box with OK and Cancel buttons. The dialog box was activated through a right-click popup menu. Before completing each task, an instructional video was shown explaining the task and how to use each interface to complete it. Participants were given a brief practice session before using both interfaces.

In Task 1, participants were asked to rate a collection of images of automobiles according to their personal taste, using the two different interfaces, the In-Context Slider and the Typical Dialog Box Slider. Images were displayed four at a time, each labeled above with a single letter.

Task 2 was different from Task 1 in that rather than rating images, participants were asked to rate single words in a text editor. The context was as if one wished to express interest in particular words in the context of an editing task. The two rating interfaces, the In-Context Slider and the Typical Dialog Box Slider were the same as before. The layer 0 activators were words, instead of images. Further, in this task, instead of spontaneously and personally rating words, participants were provided with a value

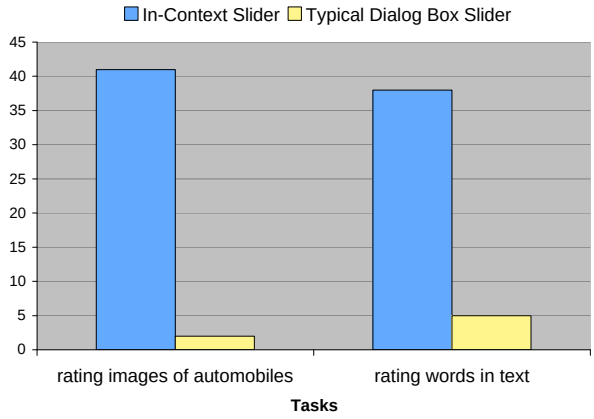


Figure 5: Time performance: with which interface were participants faster?

to assign to each word. This value was located in the text, in parentheses, following the word, to maintain contextual continuity. Words that required rating were presented in bold face to distinguish them from the other words.

The experiment was a 2x2 within-subjects design where the independent variable was the interface used for the task. All participants completed Task 1 first and Task 2 second. The independent variable conditions were counterbalanced, so that an equal number of participants used each interface first or second on each task.

The mouse interactions of participants for both interfaces in both tasks were logged. This enabled us to compute statistics about the times and answers for each condition.

3.3 Results – Quantitative

We measured how long it took participants to perform each task with each interface. Of the 43 participants, 41 (95%) [X^2 (1) = 35.372, $p < 0.0001$] for Task 1 and 38 (85%) [X^2 (1) = 25.326, $p < 0.0001$] for Task 2 were faster at rating with the In-Context Slider (see Figure 5). Average completion time for Task 1 with the In-Context Slider was 72.39 seconds, while that of the Typical Dialog Box Slider was 122.68 seconds. The difference was statistically significant [$F(1,42) = -13.263$, $p < 0.0001$]. Average completion times for Task 2 were 82.04 seconds with the In-Context Slider and 107.21 seconds with the Dialog Box Slider, and the difference between these is statistically significant [$F(1,42) = -4.535$, $p < 0.0001$] . The accuracy measures for Task 2 for the two interfaces were not significantly different.

We asked each participant which interface was easier to use. The possible responses were In-Context Slider, Dialog Box Slider, or both the same. For Task 1, 37 (86%) of the participants said the In-Context Slider was easier to use, and the results were statistically significant [X2 (2) = 54.326, p < 0.0001] (see Figure 6). For Task 2, 40 (90%) participants said the In-Context Slider was easiest to use [X2 (1) = 28.488, p < 0.0001]. Only one participant felt the Typical Dialog Box Slider was easier to use for Task 1.

Participants were also asked which interface was more natural for expressing interest. Again, both the same was the third possible choice. From the 43 participants, 33 (76.7%) for Task 1 [X^2 (2) =

37.023, $p < 0.0001$] and 32 (74.4%) for Task 2 [X^2 (2) = 32.977, $p < 0.0001$] found the In-Context Slider to be a more natural interface for expressing interest (see Figure 7).

3.4 Results - Qualitative

The participants answered open-ended questions about their experiences, from which we obtained qualitative data. Many of the participants that found the In-Context Slider to be the easiest to use noted that the In-Context Slider required less effort to use in terms of mouse clicks.

"The traditional slider was just more cumbersome to use. Having to right click then select your choice. The in context just seemed easier."

Several of participants recognized that the In-Context Slider's representation of values for interest level with red for negative and green for positive promoted comprehension.

"It was just easier. The red and green helped identify the levels easier."

The colors also provided some participants with a realization of the affect of interest expression. To them, the experience of using the In-Context Slider was tied with emotional expressivity.

"The colors made it easier to know how you felt. The pop-up was just setting a value while the in-context was almost setting an emotion."

The handful of participants that found the Typical Dialog Box Slider easier said that it was a more familiar interface. IThey were accustomed to it, and had used before. The In-Context Slider was a completely new and somewhat daunting.

4. INTEGRATING INTEREST EXPRESSION WITH AUTHORING

Providing ratings of image and text surrogates, which visually represent documents and their constituent ideas, is an important part of the user interaction in combinFormation. combinFormation (cF) is a creativity support tool that uses composition of images and text to represent collections of information resources [9]. The user directly manipulates the composition and the collection process through a set of design tools within the software. The agent semi-automatically collects, and arranges within the composition space, image and text surrogates extracted from online resources. A model of information semantics and the user's interests forms the basis for the agent's semi-automatic actions. In character with the human-centered design of cF, the user's act of providing feedback, which shapes the model, is called "expressing interest," instead of "providing ratings." The user can express interest in an information object at any time, but never has to. Prior versions of cF provided a modal toolbar-based interface for interest expression. Among the problems with this interface was the need to look away from the focus object, to the toolbar, in order to express interest. The In-Context Slider replaces the toolbar creating a fluid interface that maximizes expressivity and minimizes cognitive load and task disruption through layered activation.

Authoring tasks with combinFormation involve conceptualizing, finding, editing, designing, and composing collections of information resources [9]. The user's information needs may evolve in the course of a session, in response to the spontaneous stimulus of found information. We call tasks in which the user's goal is to have ideas while searching, browsing, and collecting, *information discovery tasks* [8]. combinFormation supports the user by using an agent to assist in the collection of information resources. However, the agent needs direction in order to effectively work in service to the user's information needs. Image and text clippings from documents in the composition space serve as affordances for interest expression, in addition to functioning as surrogates for the documents they come from.

4.1 Using the In-Context Slider for Interest Expression

As the combinFormation agent collects images and text surrogates, it also gathers metadata about each surrogate, such as

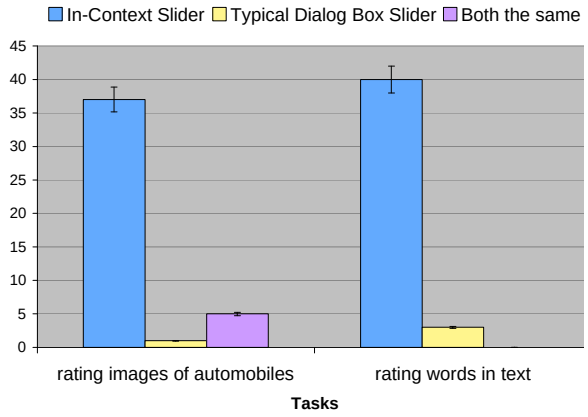


Figure 6: Participants experience reports: which interface was easier to use?

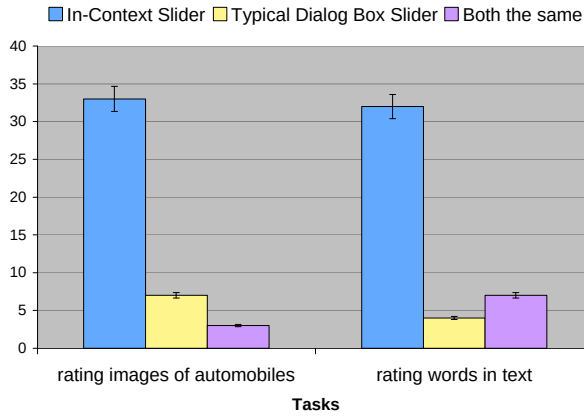


Figure 7: Participants experience reports: which interface was more natural for expressing interest?

the caption for an image, the title of the document the surrogate represents, and additional semantic fields, when available, such as author and keywords [9]. The terms from this metadata, and also the terms from within text surrogates, are used by the agent through the interest model to determine what new surrogates to bring into the composition, and which links to crawl. These information retrieval components [1] of the interest model store interest values for each term. As the interest values in terms change, the agent looks to obtain surrogates whose metadata contains terms with positive values. Users have asked for finer grained control of interest expression. The In-Context Slider gives the user the ability to directly affect the interest model on a per term basis, as well as on a per surrogate basis, in order to obtain the most relevant and interesting results from the agent.

Figure 8: Composition of surrogates created by a study participant for the summer internships information discovery question. An In-Context Slider can be activated for each surrogate and each word.

5. USER EXPERIENCES: EXPRESSING INTEREST TO DEVELOP COLLECTIONS AS COMPOSITIONS

5.1 Participants

Twenty-two subjects participated in a user experience trial. Qualitative experience reports were elicited. Once again, the subjects were students from an introductory psychology course. This was a different set of subjects than those who participated in the experiment reported above.

5.2 Method

Participants were asked to complete two information discovery tasks [8] using combinFormation. They used the in-context interface for one task, and the modal toolbar interface for the other. The interfaces were counterbalanced across participants, so that half the participants used the in-context interface first while the other half used the traditional interface first. The two information discovery tasks were:

- Your department adviser has suggested participating in a summer internship. What would you enjoy doing for a summer job? Where would you work?
- If you could spend a semester studying anywhere in the world, where would you choose to go? What would you study while there?

The two tasks were selected because of their similar levels of personal interest for the undergraduate student participants.

Prior to doing each information discovery task, participants were shown an instructional video explaining how to use combinFormation with a given interface. The video for the second task contained only an explanation of the changes between the two interfaces. The participants were given a brief warm-up session to gain familiarity with combinFormation and the interface. The participants were given 22 minutes to complete each task. The final compositions were logged for each participant on both tasks. After completing both tasks, the participants were asked to describe their experiences with the two interfaces.

5.3 Results

We collected qualitative data regarding the participants' experiences. Figure 8 depicts an example composition created by one participant. The composition shows the participant is interested in obtaining a summer internship in journalism, possibly as a news reporter. Many of the images depict news broadcasts. Several of the textual elements point to reporter jobs.

An interest in international affairs, particularly relating to Africa, is also expressed.

We collected comments about the experience through open-ended questions:

"[The In-Context Slider interface] was easy to express interest with because you could do it on the fly without having to go back and choose your interest each time."

"I could easily rate the picture I selected because the [navel] would immediately open instead of a tool bar where I had to click elsewhere and a few more times."

6. RELATED WORK

This research is related to prior work regarding recommender systems and fluid interfaces.

6.1 Ratings in Recommender Systems

Recommender systems are agent-based tools that work to find documents relevant to a user's interests. Providing ratings is a quintessential component of these systems. Recommender systems use the ratings, and techniques such as collaborative filtering [11] and information retrieval models [2] to make choices about what information resources from a larger collection to retrieve for a user. Providing ratings is personal and contemplative, requiring focus and attention. The process necessitates that the user make decisions about how interesting things are. The user must assign a valence, a positive or negative value, regarding relevance.

Despite the benefits of ranking recommendationss, the extra effort required may discourage users. McNee et al. researched differences between a user-controlled and a system-controlled recommender system [11]. By user-controlled, they mean a system in which the user decides when to make recommendations. They discovered that while the user-controlled system increased user burden, this system also provided users with more relevant results. While the user-controlled system required more time to use, some users did not seem to notice, due to a sense of increased engagement. However, the greater effort required by the user-controlled system resulted in fewer users completing the assigned tasks. combinFormation is also a user-controlled system in this sense. The present research reduces the effort of interest expression, to more easily engage users.

Others describe similar problems with getting users to provide ratings. Fab is a hybrid recommendation system using two types of recommendation methods as a way to obtain equivalent or better results with fewer ratings required by the user [3].

6.2 Fluid and Contextual Interfaces

FlowMenu is a marking menu designed for a display surface with a pen input device and allows for in-context execution of commands by making gestures with the pen device [7]. FlowMenu applies several of the same interaction principals designed for the In-Context Slider. FlowMenu uses motions that are natural and intuitive to the user to improve performance.

FaST sliders combine marking menus and the typical slider to create a new slider interface component with three stages [10]. In the first stage, a marking menu [e.g. 7] selects the value to be adjusted. The marking menu is activated by holding the control key while selecting an object and clicking the mouse. The second stage adjusts the value. The third stage allows the use of additional controls to affect the value. The FaST slider was designed for use by expert users. This mitigates issues in the lack of visibility of the activation mechanism. The In-Context Slider was designed to be used by first-year undergraduate students, many of whom lack a technical background. The In-Context Slider was also developed to integrate smoothly with authoring tasks such as text editing. In the case of text editing, mouse gestures used by the FaST slider such as click and drag are already used for positioning a text cursor and selecting text, respectively. The FaST slider requires the user to first position the slider, and then adjust the value using extra mouse click and mouse drag actions. These mouse interactions, as noted by the authors, can lead to setting the wrong value if the user moves the mouse while ending drag or releases the mouse button too soon. It also requires more effort than the layer 0 mouse over, and layer 2 mouse move motions used to adjust a value with the In-Context Slider.

Fluid links are a mechanism in hypertext for displaying information about a hyperlink in-context to help the user decide which hyperlinks to follow [e.g. 11]. When a user mouses over a fluid link, the visual layout of the hypertext document is modified by the addition of new information about the link placed on the line below the link, moving all lines below down a few lines, or in a margin to the right or left of the fluid link. Fluid links are similar to the proposed in-context interface in that layers of activation are engaged when the user mouses over a fluid link.

Side Views is a user interface component that provides on-demand details along with persistent and dynamic previews for a given command [15]. Side Views supports open-ended tasks in which case it is unclear the sequence of steps required to reach the desired final solution. Side Views provides in-context visualization by displaying previews directly next to the point a command is selected and executed (e.g. a menu item from a drop-down menu).

Local Tools is an alternative to tool palettes and arguably the antithesis of the In-Context Slider [3]. Local Tools provides the user with tools that can be picked up, used, and then dropped anywhere on the screen. This idea differs from the standard tool palette in that tools are not fixed to single location allowing placement of tools near the point of interaction. The In-Context Slider addresses a problem that Local Tools inherited from the standard tool palette: the user must still shift focus to select the tool.

Data Visualization Sliders use information visualization techniques to enhance sliders [6]. Data Visualization Sliders use a slider's screen real estate to visualize information in the form of graphs with both continuous and discrete values. Each graph shows information related to the data value adjusted by the slider.

See-Through tools are translucent tools located on a plane above the interactive space [5]. The user interacts with objects through these tools to apply the tools' effects to the objects below. The tools can be moved around the screen, between applications, and layered on top of each other. The In-Context Slider is not a See-Through tool; it shares the translucence quality. The layers of activation, although serving different functionality, are similar in concept to See-Through tools' layering capabilities.

7. CONCLUSION

New interaction modalities require new integration of functionalities. Providing different kinds of interactivity in context, so that, for example, the user can fluidly switch from authoring to rating and back without visually context switching, is an interaction design challenge. The In-Context Slider meets this challenge by integrating its visual representation with that of

surrounding content, and minimizing the cognitive and physical effort of activation.

Many of the current parameter value adjustment interfaces require extra effort and attention on the part of the user. These interfaces are often activated through a series of menus or keyboard commands and located in a popup window or a side bar that is not always located near the object of interest. Some use dedicated web-based forms with slow responses. Some waste screen real estate with non-transitory affordances [1]. Others use invisible control characters for activation, which novices may not recall. Thus, the user may fail to use the interest expression interface. Fluid in-context interfaces seem appropriately suited for interest expression mechanisms. The minimal effort required to use these interfaces can overcome the reluctance of users to express interest. A user's decision about the relevance of information occurs while that information is in the user's focus. Having an interest expression mechanism appear in-context allows the user to express interest immediately and directly. Integration with authoring enables the user to focus attention on more primary tasks, and perform interest expression spontaneously when it feels worthwhile.

The quantitative and qualitative results show that the In-Context Slider is quicker and easier to use than the Typical Dialog Box Slider. The In-Context Slider, through its fluid layers of activation, allowed the participants to more rapidly express interest with minimal distraction. The In-Context Slider's layer 0 and layer 1 activators provide less disruption of the interactive space than the typical right-click popup menu. The sleek, precisely positioned, and translucent In-Context Slider layer 2 body is likewise designed to blend with and contribute to the participant's focus of attention within the interactive space, in contrast with bulky opaque dialog boxes that obscure context.

More than three fourths of the participants found the In-Context Slider to be a more natural interface for expressing interest than the Typical Dialog Box Slider interface. This result points out a problem with many of the standard interfaces for rating. These interfaces were designed primarily to obtain data for agent software, rather than to support human users. A human-centered design approach changes the experience.

The results are striking, considering that the In-Context Slider is a new interface, with which the participants had no prior experience. This was borne out by the qualitative data, in which the few participants who preferred the typical interface told us that they preferred it because it was familiar. This discrepancy, though not large, would be reduced in a realistic usage scenario longer than a 60 minute laboratory experiment. The performance and ease of use findings are particularly significant since participants were not users with a particular background in interactive systems.

Shneiderman and Bederson proposed three strategies to help better maintain user attention: reduce short-term and working memory load, provide information abundant interfaces, and increase automaticity [14]. By automaticity, they were referring to designing command sequences such as keyboard shortcuts that reduce the interactive steps required to complete tasks. With the In-Context Slider, as a fluid in-context interface, we instead increase automaticity through visual design. By designing simple, distinguishable visual affordances such as the navel, the user is able to quickly recognize interaction possibilities.

The navel is a small, simple and clear affordance providing visual continuity between un-activated and activated states, and visualization of a value with minimal disruption of context. With the navel located in the center of an In-Context Slider, it places the mouse cursor at the center of interaction. The navel functions as a focal point for interacting with an In-Context Slider. It helps the user learn what the slider does and how it works, forming a recognizable affordance, that when seen again, a user will understand its function. While we used timeout for activation of the navel by novice users, control click can be used by experts.

User engagement in laboratory information discovery tasks using combinFormation with the In-Context Slider proved meaningful for personal growth and development. After viewing compositions that participants created, it became clear that some participants, such as the creator of Figure 8, went through a thought provoking process in which they obtained information and synthesized ideas that may actually affect future decisions in their lives.

The In-Context Slider was designed to minimize physical effort. This minimization should reduce any occurrences of Occupational Overuse Syndrome in comparison to other interfaces, which require more mouse clicks and a greater range of mouse movement.

A primary design concern when developing the In-Context Slider was screen real estate. In an instance where minimal screen real estate is not a problem, the In-Context Slider is not necessarily the best solution. In this particular case, a normal slider can be displayed in-context at all times; therefore, negating a need for a transitory interface like the In-Context Slider.

Authoring is an iterative process of creating, collecting, refining, and composing ideas. The process involves emphasizing certain ideas and discarding others. Expression is an important part of this process. When authoring with systems like combinFormation that use agents, expressing interest in relevant information is beneficial. Yet, it can take attention away from other task components. Thus, an interface for interest expression needs to minimize the demand on a user's attention, allowing action to be accomplished easily, as if expressed through the body, and not through a disembodied interface. The full set of design choices for the slider: color, fluidity, translucence, integration, fluid gesture, and lack of saccadic movements produce an embodied sense of affect that promotes expression.

8. REFERENCES

1. Apple, iTunes, http://www.apple.com/itunes/
2. Baeza-Yates, R. and Ribeiro-Neto, B. Modern Information Retrieval, New York: Addison Wesley, 1999.
3. Balabanović, M. and Shoham, Y. Fab: Content-Based, Collaborative Recommendation, *Communications of the ACM*, v.40 n.3, 66-72, March 1997.
4. Bederson, B.B., Hollan, J.D., Druin, A., Steward. J., Rogers, D., and Proft, D. Local tools: An alternative to tool palettes. *Proc ACM UIST 1996.*
5. Bier. E.A., Stone, M.C., Fishkin. K., Buxton, W., and Baudel, T. A taxonomy of see-through tools. In Proceedings of *ACM CHI'94*, 358–365, 1994.
6. Eick, S. Data Visualization Sliders. In Proceedings of *ACM UIST'94*, 119-120, 1994.
7. Guimbretiere, F. and Winograd, T. Flowmenu: Combining command, text, and data entry. *Proc UIST'00*, 213–217, 2000.
8. Kerne A. and Smith, S.M. The Information Discovery Framework. *Proc DIS 2004*, 357-360, 2004.

9. Kerne, A., Koh, E., Dworaczyk, B., Mistrot, J.M., Smith, S.M., Graeber, R., Caruso, D., Choi, H., Webb, A., and Joshi, P. A Mixed-Initiative System for Representing Collections as Compositions of Image and Text Surrogates, *Proc JCDL 2006*.
10. McGuffin, M., Burtnyk, N., and Kurtenbach, G. FaST Sliders: Integrating marking menus and the adjustment of continuous values. *Graphics Interface*, 2002.
11. McNee, S.M., Lam, S.K., Konstan, J.A., and Riedl, J. Interfaces for Eliciting New User Preferences in Recommender Systems. *Proc User Modeling 2003*.
12. Nagy, A.L. and Sanchez, R.R. Critical color differences determined with a visual search task, *Journal of the Optical Society of America*, A 7, 1209–1217, 1990.
13. Norman, D. The Design of Everyday Things, New York: Basic Books, 1988.
14. Shneiderman, B. and Bederson, B.B. Maintaining Concentration to Achieve Task Completion. *Proc DUX'05*.
15. Terry, M. and Mynatt, E.D. Side views: Persistent, on-demand previews for open-ended tasks. *Proc ACM UIST'02*, 71–81, 2002.
16. Zellweger, P.T., Chang, B.W., and Mackinlay, J.D. Fluid links for informed and incremental link transitions. *Proc ACM HT'98*, 50–57, 1998.

User Experience

Exploring the Feasibility of Video Mail for Illiterate Users

Archana Prasad[1], Indrani Medhi[1], Kentaro Toyama[1], Ravin Balakrishnan[2]

[1]Microsoft Research India
196/36, Scientia, 2nd Main Road, Sadashiv Nagar
Bangalore, India

[2]Department of Computer Science
University of Toronto
Toronto, Ontario, Canada

arcna@hotmail.com, indranim@microsoft.com, kentoy@microsoft.com, ravin@dgp.toronto.edu

ABSTRACT

We present work that explores whether the asynchronous peer-to-peer communication capabilities of email can be made accessible to illiterate populations in the developing world. Building on metaphors from traditional communication systems such as postal mail, and relevant design principles established by previous research into text-free interfaces, we designed and evaluated a prototype asynchronous communication application built on standard email protocols. We considered different message formats – text, freeform ink, audio, and video + audio – and via iterative usage and design sessions, determined that video + audio was the most viable. Design alternatives for authentication processes were also explored. Our prototype was refined over three usability iterations, and the final version evaluated in a two-stage study with 20 illiterate users from an urban slum in Bangalore, India. Our results are mixed: On the one hand, the results show that users can understand the concept of video mail. They were able to successfully complete tasks ranging from account setup to login to viewing and creating mail, but required assistance from an online audio assistant. On the other hand, there were some surprising challenges such as a consistent difficulty understanding the notion of asynchronicity. The latter suggests that more work on the paradigm is required before the benefits of email can be brought to illiterate users.

Categories and Subject Descriptors

H.5.2 Information interfaces and presentation: User interfaces

General Terms

Design, Experimentation, Human Factors.

Keywords

ICT for development, video mail, illiterate users.

1. INTRODUCTION

Information and communication technology for development, focuses on computing applications for socio-economic development of underserved communities [12, 13, 14, 15, 16, 19, 20, 20, 22, 23, 24, 26]. One common characteristic of these underserved communities is illiteracy. Even conservative estimates of illiteracy suggest that there are over one billion illiterate people in the world [11]. As such, there is value in computing application intended to aid and be used by people in exploring how computing can be made accessible to illiterate users. This poses a significant design challenge, as the sheer abundance of text in standard interfaces suggests that significant retooling of the interface or completely new interaction styles would be required to ensure usability by illiterate populations.

Conference AVI '08, May28-30, 2008, Napoli, Italy.

Previous work in designing user interfaces for illiterate and semi-literate populations focuses on broad principles, recommending features such as the use of graphical icons [6, 7, 12, 13, 14, 19, 20, 20], minimal use of text [2, 7, 12, 13], voice annotation [12, 13, 14, 20], easy navigability [2, 7, 12, 13, 17, 20, 20] and the use of numbers for people who may be illiterate but not innumerate [12, 13, 19, 20, 20]. These principles have been applied to applications in the areas of job search [12, 13, 15], healthcare [6, 14], map navigation [13] and microfinance [19, 20, 20], but has not yet been significantly applied to computer-mediated communication.

The majority of communication applications targeted towards illiterate users are in the area of agriculture and dedicated to query-based communications between an illiterate person and a literate agricultural expert [22, 24, 26]. These applications, and of course the increasingly ubiquitous mobile phone, currently only provide illiterate users with voice-based communication that is typically synchronous. To the best of our knowledge, there are no applications yet developed for asynchronous computer-mediated communication dedicated to illiterate users. In particular, the most common asynchronous communication tool, email, which has had a profound impact on the lives of the world's literate population, is essentially inaccessible to illiterate people.

In this paper, we explore the question of whether and how the benefits of an asynchronous communication tool like email might be made accessible to populations with little to no literacy. Our ultimate goal is to create a communication experience built on standard email protocols – thus enabling simple inter-operability with other systems used by the literate world – that do not require literacy to enable effective asynchronous communication. As a first step towards realizing this goal, we explore the use of video rather than text as the communication medium. We present the design and evaluation of a prototype video-mail application that uses a combination of graphics, animation and voice assistance to empower illiterate users to be completely self-reliant right from setting up accounts through communicating using it.

In the following sections, we further discuss the motivation and challenges of this problem space, previous research from which we draw design guidance, the user community we worked with, the iterative design and implementation of a fully functional prototype video mail system, and a two-stage user study that evaluated this functional prototype with twenty illiterate users from the urban slums of Bangalore, India. We conclude with a discussion of the insights gained from both our iterative design process and user study, and design recommendations for future instantiations of the concept of video mail for illiterate users.

2. MOTIVATION and CHALLENGES

There are over 1.2 billion email users in 2007, and this number is expected to rise to 1.6 billion by 2011 worldwide [25]. Thus, email is clearly a dominant asynchronous communication tool amongst literate people worldwide. However, users are expected to have several pre-requisite skills, including:

- Literacy or at least semi-literacy.
- The ability to recall login information that is a combination of text, numerals and symbols.
- The ability to set-up their account by understanding the standard settings of the service provider.
- Dealing with email client applications that are normally text-intensive.
- An understanding of and ability to use a navigational system which is heavily text-based.
- An understanding of hierarchical structuring of information (folders, etc).
- Constant decision-making from multiple existing choices to achieve a task.

These skill requirements of current email systems make them essentially inaccessible to the world's one billion illiterate people. Now, one could argue that this population might be better served with technologies such as mobile phones which have a lower usability barrier. However, the synchronous communication capabilities provided by mobile phones are clearly not suitable for all communication scenarios. More asynchronous forms of communication are also desirable, as the numerous "professional letter writers" who ply their services in the towns and villages of the developing world (particularly in India) vividly illustrate. While the age-old physical letter continues to serve its intended purpose, as these populations become increasingly mobile, traveling farther afield from their home villages in seek of work, the need for efficient communication with their now geographically distant relatives and friend becomes more acute. Thus, more efficient alternatives to the postal mail service merit serious investigation. Given email's established track-record as the asynchronous communication tool of choice amongst the technologically-literate peoples of the world, it is arguably reasonable to see if we can somehow morph this technology to also serve the communication needs of the illiterate.

Doing so requires significantly ameliorating the skill requirements of current email systems as enumerated above. The primary challenge is in providing both an interface and communication medium that does not rely on text. The difficulties inherent in text as the medium of communication could potentially be solved by using video and audio exclusively. The arguably more daunting remaining challenge is in designing a user interface that would allow asynchronous "email-like" video and audio communication in a facile manner for people who are not only illiterate but also completely novice computer users. In approaching this challenge, we draw from previous research interfaces used in other application domains.

3. RELATED WORK

There are three areas of related work which are particularly relevant to our research. First is the work on user-interface design for illiterate users. Second is the area of web-based asynchronous communication applications focused on novice but literate users. Third is the work on pictorial passwords for authentication.

3.1 Interfaces for Illiterate Users

Early research in this area placed emphasis on the need for contextual design methods to explore this problem, as illiterate users are very different from the target user imagined by most interface designers [3]. We follow this lead, and have spent literally hundreds of hours in the field working with non-literate people on a variety of projects. Most previous work with non-literate users focuses on the mechanics of the interface. In particular, researchers recognized the value of imagery in place of text, and extensive use of graphics is advocated by most of this work [6, 7, 12, 13, 14, 19, 20]. Some also explored the value of voice instructions and annotations. Much of the interesting work in this area focuses on the subtle interplay between graphics and audio to generate a compelling interface. Some authors note that it might be plausible to include numerals, as illiterate users are often numerate [12, 13, 19, 20, 20], while others have focused on ulta-simple navigation as a design goal [6].

These principles have been applied to create text-free user interfaces [12, 13, 14, 15] for similar user groups for other application domains such as a job-information system for illiterate domestic helpers [12, 13, 15], a health information dissemination system for illiterate patients [14], map-navigation [13] and microfinance [19, 20, 20]. One high-level goal of these systems was to create interfaces that an illiterate person can use, on first contact with a computer, to immediately perform useful tasks with minimal additional human assistance. Since our interface goals are clearly very similar, it is worth identifying some design principles that arise from this prior art. In particular, we draw upon earlier work by members of our research team [13], where design principles for text-free interfaces were elicited through an ethnographic study involving over 300 hours and 200 people from urban slums in Bangalore, India. They are as follows:

Liberal use of graphics and imagery; use of static hand-drawn representations with voice annotations: While the use of graphical imagery rather than text is an obvious feature, the exact nature of the graphics can make a huge difference. A comparative study [14] between different audio-visual media with 200 participants showed that static hand-drawn representations with voice annotations were best understood.

No use of text, but numbers might be acceptable: While less text makes sense for people who cannot read, it was discovered that illiterate people could often easily recognize numerals (0, 1, … 9), hence it might be reasonable to use numerals in the interface, at least for some subset of the illiterate user population.

Voice feedback and help throughout: Prerecorded human speech segments as a way for the interface to "converse" with the user was found to be extremely valuable. As such, it is recommended that voice feedback and help be provided throughout the interface.

Consistent "help": Easily accessible and always available auditory help allows an application to be more autonomously used, even for novice users. Additionally, an on-screen character could help users relate the voice to a visual representation.

3.2 Asynchronous Communication Applications for Novice Users

Researchers have looked at communication systems between novice users and domain experts through a query-based approach using voice [22, 24, 26] and video [26]. Both these examples are in the agriculture domain. While these do use voice and video like we intend to, they do not address the peer-to-peer personal communication domain that is the focus of our work.

In the personal communication domain, research on Chinese migrant workers and the interaction with their children has been studied [17, 18]. While these studies explore research questions similar to those we are interested in, they were essentially initial feasibility explorations that recommend voice-based communication augmented with video. No hi-fidelity working prototype was developed and as such no significant usability studies were conducted using a working system [17, 18].

3.3 Pictorial Passwords for Authentication

Researchers have explored the use of pictorial passwords for user authentication that provides better recall and usability [3, 5, 8]. However, most of this work aims at providing insights and principles for literate computer users. In a series of experiments, Katre [8] studied the use of pictorial passwords for illiterate populations and found that while users could easily recall previously seen pictures, they were not necessarily able to recall a set of pictures in a particular sequence as would be required for a typical password authentication scheme. Katre proposes various schemes that might mitigate the sequence recall problem, but these have yet to be tested in a real application.

4. TARGET USER COMMUNITY

We conducted our research with people in three urban slum communities in Bangalore, India. To gain access into these communities, we worked with a non-governmental organization (NGO) called Stree Jagruti Samiti (SJS), which has had an established presence in these three communities for 15 years. SJS works primarily with the women and children in the slums. All of the people we worked with had three common background traits: (1) functional illiteracy or semi-literacy (typically somewhat numerate); (2) low levels of formal education (highest education attained being schooling up to the fourth grade); and (3) no experience whatsoever in using a computer. These traits make them an ideal user population with which to explore our ideas with regards to creating a voice and video based email surrogate suited for illiterate populations.

These communities we work with have their own unique characteristics, and these should be kept in mind when attempting to generalize the results we present later. For example, populations differ in terms of their attitudes toward illiteracy. The people we worked with were very frank with respect to illiteracy, attaching no shame to the inability to read; this is unlike illiterate individuals in developed countries who often hide this inability. Also, our users held strong positive associations of the English language (which they did not speak for the most part) with wealth and prestige – both a holdover from colonial British rule, as well as a modern-day fact due to the economic opportunities available to English speakers. These characteristics might have had an impact on our results, in a manner that could well have been different had our users been people from other locations and cultures. The state of illiteracy, poor education, and ignorance of computer technology are factors common across all our users and are arguably those that impact our designs the most. Other demographic characteristics of this population likely had less influence on our particular problem space; however, we list them here for the sake of full disclosure: About half the people were female household workers who clean private homes, wash dishes, and so forth. The other half were males who are typically daily wage laborers like plumbers, carpenters, construction workers, mechanics, or fruit and vegetable vendors. Their primary language of communication is Kannada, but many speak additional languages such as Hindi, Tamil, or Telegu. The average household income was INR 800 - INR 3000 (approximately USD 18 – USD 67) per month, in line with general market statistics on wages in India [1]. A few had television sets, music players and gas burners, but these were not owned by all households. Some had seen computers in the houses of their employers, but due to class- and caste-based discrimination, were generally prohibited from touching the computer (even for the purposes of cleaning!).

5. PROTOTYPE VIDEO MAIL SYSTEM

While some aspects of our explorations into the feasibility of video mail as a synchronous communication technology for illiterate users could be accomplished using primarily low-fidelity methods, we chose to iteratively develop a higher-fidelity working prototype system instead. Since a working system will allow us to conduct true usability tests in the field, we believe it will allow us to obtain a more ecologically valid and deeper understanding into the issues surrounding the potential use of this technology. The prototype was designed and refined over three stages:

5.1 Initial Designs and Mental Models

In the first stage of our design process, we used principles suggested by prior research in text-free user interfaces [12, 13, 14, 15] to generate the initial user interface. In addition, we strove to promote a strong mental association with existing systems of communication as a method to ease novice users into this technology. Accordingly, we chose the postal system as a metaphor that would be easily understood by our target community. The user interface's "look" had the essence of a familiar Indian postcard (Figure 1), and the functions and controls simulated the process of creating and posting it. A visually salient audio assistant was designed as a post-person who explained creating, sending and receiving video mail in terms of the postal system's process that was familiar to this user population.

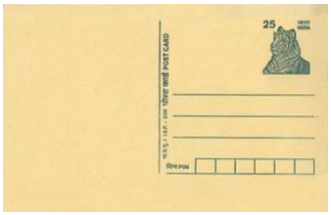


Figure 1. Indian postcard used as a familiar metaphor.

Our initial prototype design is illustrated in Figure 2. On the left of the screen is an image of a woman dressed as a typical Indian post-person that represents the audio assistant who provides context specific voice help when the user hovers the mouse cursor over it. This audio assistant remained on screen at all times as it was our intent to provide a consistent place where users could turn to for help at any time. Figure 2a shows the login screen, which consists of numbers for the user ID, and pictures for the password. Figure 2b illustrates the inbox of messages, consisting primarily of a photograph of the sender with color coding to indicate new

and old messages, and a time of day icon (moon and sun), and the area used for creating messages in four possible formats: freeform ink which we thought might be viable as a means of spontaneous expression, audio only messages, audio+video, and text – included for backward compatibility for the literate user. Clicking on the relevant message format icon immediately started the recording. We chose to support these three non-text formats rather than video alone as at this initial stage we did not want to presume that video was necessarily the best format for asynchronous communication by illiterate users.

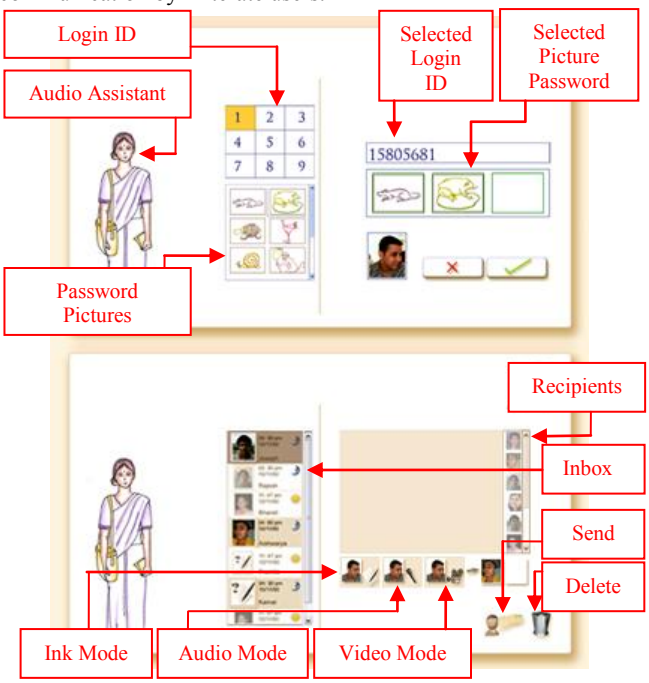


Figure 2. Initial prototype. (top) Login. (bottom) Inbox & mail creation. Red annotations are for illustration only.

Using this initial prototype, we sought feedback via informal user testing and interviews with three people representing illiterate, semi-literate and seasoned computer user groups. Note that we deliberately included a semi-literate and seasoned computer user because we wanted to get a sense of how the perceptions and usability of our designs might differ depending on the user population. These sessions led to several key observations:

a) Login has to be simplified. User name as a combination of numerals and the password as a combination of images required recall on two separate parameters, and this proved difficult for the users.

b) Vertical scrollbars were not understood by the illiterate and semi-literate users, as the arrows simply went unnoticed.

c) Color coding for old and new messages was not understood.

d) Actions should occur only when the user conducts a specific function and not automatically. For example, the prototype started recording immediately upon clicking the relevant message format button, rather than using additional "start/stop recoding" buttons, resulting in some confusion.

e) Animation is needed when transitioning from one screen to the next. Abrupt switching as is typically done in regular applications resulted in confusion due to a loss of continuity.

5.2 Free-Form Study of Revised Prototype

In response to the key observations in the previous stage, we made several modifications to the prototype in turn:

a) Changed the login mechanism to use photos of users as the login ID instead of the numeric ID. The pictorial password mechanism used in the first stage was retained. (Figure 3a)

b) Replaced the vertical scrollbars with much larger up/down arrow icons at the top/bottom of lists as required (Figure 3b).

c) Removed the color coding of old and new messages. Unfortunately, we were not able to devise an alternate mechanism for distinguishing between old and new messages that we felt would work for this population of users.

d) Added explicit start, stop, and play buttons to control the playback and creation of mail messages (Figure 3b).

e) Added smooth animated transitions between screens.

f) Added an account creation phase for first time users.

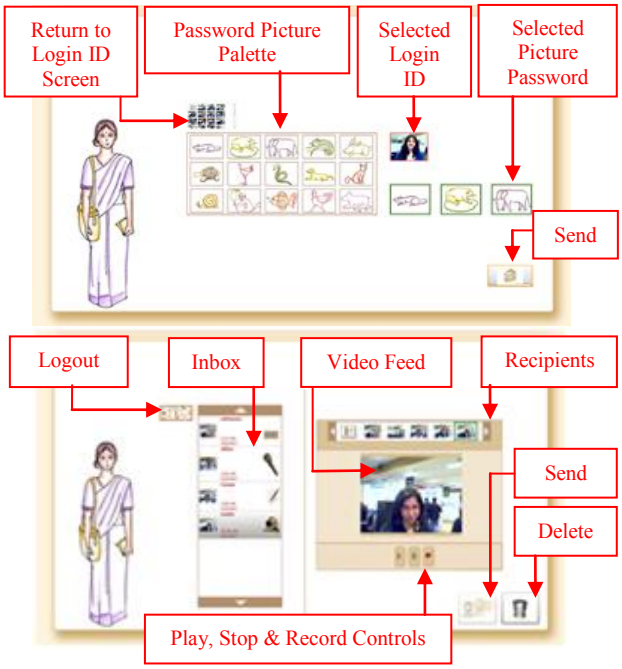


Figure 3. Revised prototype. (top) Login. (bottom) Inbox & mail creation.

In order to further evaluate this revised prototype, we installed the software on an unattended computer accessed by a group of thirty people whose background was comparable to our ultimate target user community. These users were employed as cleaning and facility maintenance staff at our corporate office in Bangalore. The software ran on this computer for a six-week test period. We chose to do this phase of testing in this setting rather than with the ultimate target users in the slum communities because we wanted to refine our prototype as much as possible within our controlled facilities before taking it out into the field.

The task required the participants to set-up their own mail account on the system, login and send a mail. We observed whether the application could be used without human assistance, if in fact it would be used at all, and the hurdles faced while using it.

Attention was paid to understanding which features were most used and why. This stage resulted in the following observations:

a) Users struggled with setting-up their account as we had not completely eliminated the requirement for text input in this phase (*i.e.*, names had to be typed in as text). This was not unexpected, and we were concurrently working on a text-free account setup design to be implemented in the final prototype.

b) The application sometimes offered more than one way for users to accomplish their task. In particular, when creating a new mail, users had to choose between video, audio, ink, and text. This choice was found to be confusing by some users.

c) Video mail was the most used. Users occasionally used ink mail (drawn images as mail), rarely used voice mail, and predictably never used text mail.

d) Interface elements that were not to be used and were grayed-out perplexed them.

e) Users found innovative uses for video mail such as creating a song chain or reading out the new headlines (if they were semi-literate) for their illiterate co-workers.

f) Often they would help each other with setting-up and using the application. Superfluous visual elements that were used to enhance the post-card metaphor proved to cause confusion.

g) Some users faced difficulty in remembering the order of the graphical passwords.

5.3 Final Prototype

Based on the feedback from the previous stage the application was further revised in several ways.

First, we redesigned the account setup interaction to avoid any text input, relying instead on taking a photo of the user and using it as the login ID with a unique computer generated identifier assigned automatically to that photo. Further, given the difficulties faced by users in remembering the order of the graphical passwords – a finding similar to that found by Katre [8], we replaced the pictorial passwords with numeric ones instead. While we do not know *a priori* whether or not this would work better than pictorial passwords, we felt it was worth exploring as pictorial passwords were clearly not feasible. Figure 4 illustrates.

Second, we significantly simplified the rest of the interface, in particular retaining only the video mail feature and removing the ink, audio, and text mail formats as choosing between multiple formats was found to be confusing by users in the previous stage. Audio assistance was enhanced, and retained the postal worker metaphor. Figure 5 illustrates.

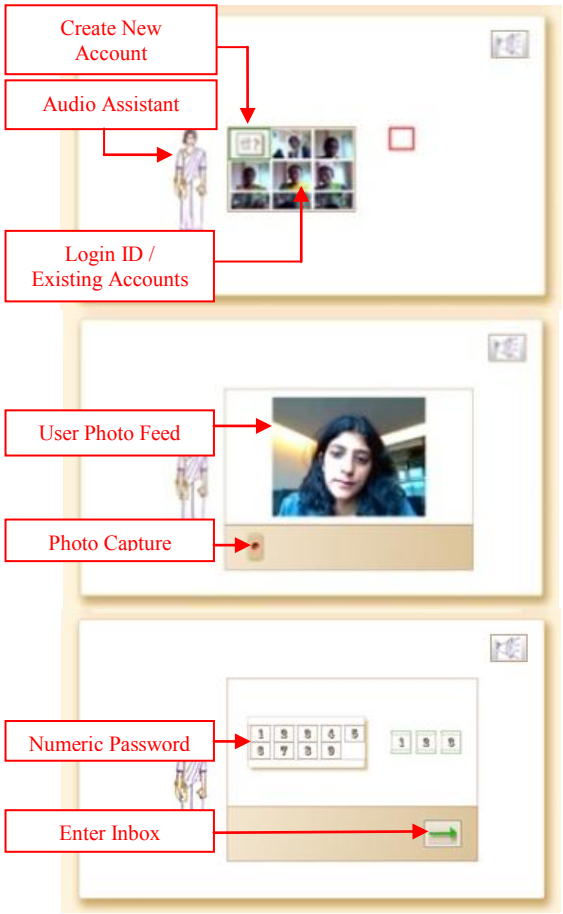


Figure 4. Final prototype: account creation. (top) Opening page showing existing accounts and a "create new account" icon. (middle) Capturing a photo for a new user ID. (bottom) Selecting numeric password.

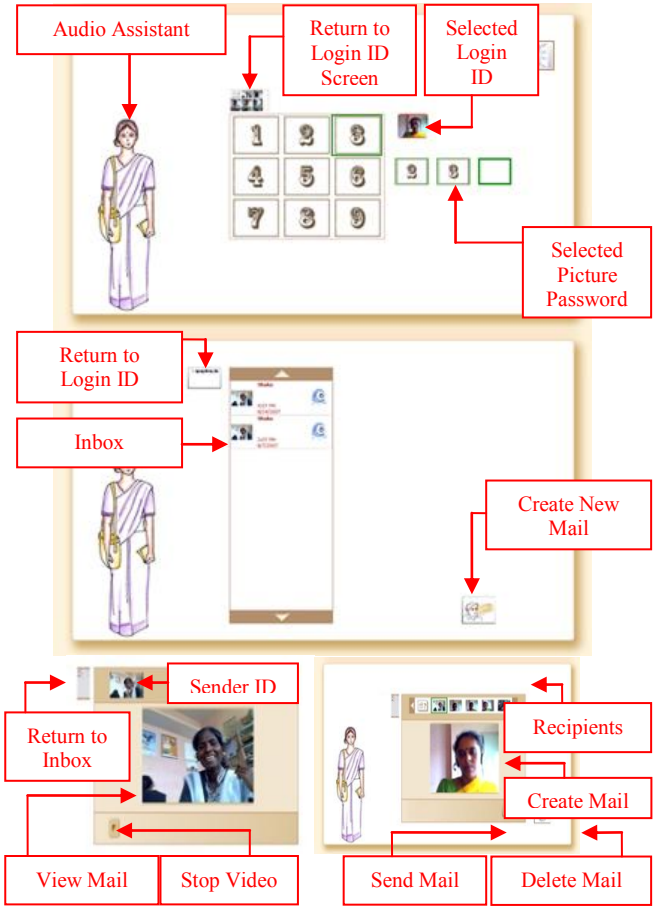


Figure 5. Final prototype. (top) Login screen. (middle) Inbox. (bottom left) View video mail. (bottom right) Create video mail.

6. USER STUDY

6.1 Goals and Design

We evaluated the final prototype with 20 participants (10 female, 10 male) from the target user community described in section 4. Participants ranged in age from 25 to 45 years.

The study was intended to determine if our target users would be able to understand the overall concept of video mail and perform the actions required to setup an account, login, and send/receive mail. The study was conducted in the homes of the participants, and we used a tablet PC and a pen as we felt it more closely resembled the paper-and-pen letter-writing metaphors that participants were used to. A representative of the NGO we work with acted as a primary contact person with whom the participants would communicate via the video mail system. This primary contact person was well known to all participants, and pre-recorded a welcome email message that was shown in the inbox of all new accounts. We conducted the study in two stages:

Stage 1:

Using a pre-authored script, participants were walked through the system by the experimenter (the first author). The overall concept of asynchronous email communication was explained to them using analogies to the postal mail system that participants were already familiar with. Participants were shown how to use the tablet PC and pen and told that they can seek assistance from the audio assistant at any time.

They were then asked to perform a set of tasks as follows:

a) Set up their own video-mail account. This involved taking a user photo of themselves and selecting a 3 digit numeral-based password as shown in Figure 4.

b) Login to their new account (Figure 5a).

c) Retrieve a welcome message in their inbox created by the primary contact person

d) Compose and send a response to the welcome message

e) Logout

Stage 2 (10 days later):

The primary contact person looked at and replied to all the mails sent to her from the previous stage before the same set of participants began this stage. We deliberately conducted this second stage 10 days after the first stage was completed as we wanted to see if after some time away from the system, participants could recall their passwords and how to use the system without the help of the experimenter. Unlike in the first stage, in this second stage the experimenter did not do an initial walkthrough of the system, but was available to assist if participants got completely stuck and were unable to proceed otherwise.

Participants were asked to perform a set of tasks as follows:

a) Log into their video mail account

b) Retrieve new mail from the main contact

c) Compose and send a response

d) Log out

6.2 Results and Discussion

Overall, each participant took about 5 to 20 minutes to complete the task in stage 1, and 5 to 10 minutes in stage 2. One female user did not show up for the second stage of the study, but the remaining users all eventually completed the task in both stages. Overall, we found that the male users completed the task faster and were more at ease with the technology than the female users.

We used four techniques for data collection: detailed notes taken by the experimenter in-situ while the participants were performing the tasks, continuous screen captures using a software tool (Community Clips) to record all on-screen activity, a video camera that recorded participants actions from an "over the shoulder" vantage point, and a software logger that recorded all mouse and keyboard inputs within the application.

The following are key observations gleaned from careful analysis of the detailed notes taken by the experimenter, and manual coding of the many hours of screen captures and video camera data. We ended up not using the data from the software logger that recorded mouse and keyboard inputs, as we found that data at such low level of detail was not necessary to determine the essential usability issues.

A key aspect of our overall design was the availability of the audio assistant at all times, and we were very interested in how this style of assistance fared with our target users. We found that:

a) Users were unable to follow multiple linear audio instructions, and most often just followed the first or last in the series. For example if the audio assistant says "Please click on the Play button to play the video mail, the Stop button to stop the video mail and the Record button to record a video mail", users will disregard the fact that they have a choice and simply follow through with either Play or Record.

b) Unless prompted by the experimenter, users did not use the audio assistant in stage one of the study. However, during the second stage of the study it was observed that the users were significantly more confident and needed less prompting from the experimenter to use the audio assistant. As Figure 6 illustrates, the audio assistant was used extensively during both stages of the study, despite the relatively simple nature of the tasks performed by the users. This indicates that having continuously available help is crucial for these users.

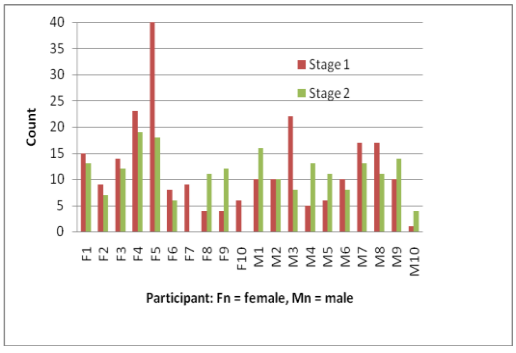


Figure 6. Number of invocations of audio assistant by each user during both stages of the study.

c) Audio instructions that relied on color may be misleading even when used in combination with another parameter. For example, if the audio assistant instructs the user to click on a green arrow, the user is likely to try and click on anything

green, including green parts in the background of a video-still. This suggests that color coding should either be avoided completely or used only when the rest of the screen does not contain the same color.

d) Users tended to hold their mouse over the audio assistant through a whole message, continually hovering the cursor over the icon without clicking.

e) Congratulatory audio messages seem to produce excitement and encouragement. For example, after going through the login process of selecting the login photo and correct password, an audio congratulatory message informing the user in a congratulatory tone that they had successfully entered their inbox and that they could now retrieve mail or create a new one produced a lot of positive excitement.

With regards to the concept of video mail and receiving/creating new mail, the data shows that receiving personalized video mail was clearly seen as an exciting event. It was very interesting to observe that many participants did not understand that the welcome video mail was pre-recorded. They attempted to have conversations with the video mail, even when it was re-played several times over! A possible hypothesis is that their mental model of the synchronous telecommunication system overrides our intended asynchronous postal-system mental model. The moment users are faced with a video of a person talking, they immediately respond as they assume that this is similar to a telephone conversation – that this is a video phone. This finding was a bit disappointing, as we had put in significant effort in designing the interface to project an asynchronous model. Clearly further work is required to get this aspect of the system right.

With regards to the authentication scheme, we found that the combination of photograph as the login ID and numeric passwords worked reasonably well. However, we observed that users sometimes had difficult deciphering all numerals and confused the numbers 2, 3 and 5. From interviewing users on this issue, we discerned that many tended to remember the numbers by their placement within the on-screen number pad rather than by actual recognition of the numerals per se. This indicates that one should keep the on-screen visuals of the numerals consistent across versions of software, and potentially use a similar login screen for multiple applications for this population in order to reduce any recognition confusion. Regardless of the mechanisms by which they recalled the numeric passwords, in stage one, all the male users were able to create their passwords in the account setup stage and immediately thereafter reenter their password on the login screen with no errors. Six of the ten female users made mistakes when reentering their password on the login screen, even though they had just created the password moments ago in the account setup stage. We attribute this to differences in numeracy between the males and females in our user population. Interestingly, however, in the second stage of the study, male and female users were equally adept at recalling the passwords they had created ten days earlier, with just three men and three women making mistakes on the first attempt at password entry.

In the rare case that two users share the same password and one of them logs in with the other user's photo, it was observed that they were unable to decipher that they were in the wrong inbox. This leads us to believe that a personalized welcome message when a user enters their inbox is required.

The data also revealed several other key issues with regards to various interface elements. While these were observed within our video mail application, many are general issues that would apply to any application for illiterate populations and hence potentially have implications beyond the present work:

a) Linear progressions are not conceptually understood. Users did not understand that they were being taken from one screen to the next. Thus icons that show the previous screen as a means to get to the previous screen may be moot.

b) Users had some difficulty identifying 2D thumbnail photos of themselves. This could be due to poor quality of images taken by webcam and the small thumbnail size and poor eyesight that some users might have. A potential design solution might be to increase the image size on mouse-over.

c) Users did not seem to realize that they need to click on the 'Stop' button to stop the action of recording a message. This was in spite of clear audio instructions to do so. We suspect this might be due to users thinking of the recording as a synchronous open communication channel where there is no explicit end. We continue to seek better ways of reinforcing the notion of asynchronicity.

d) Similar to the 'stop' issue above, users did not seem to see the need to "exit" the application on completing their tasks. The notion of an application that had to be started and stopped is clearly foreign to this population. Perhaps a kiosk-style "always on" appliance might be more appropriate.

e) Several users clicked on the lower (older) mail in the inbox when attempting to access their new mail. As noted in the earlier phases of our work, standard grouping techniques such as color coding of new and old messages were ineffective. We intend to explore other techniques such as putting old messages in a completely separate space on screen.

7. CONCLUSIONS

Our work suggests that providing a personal asynchronous communication system for illiterate users could be viable. Our user study showed that users were able to grasp the basics of the application and complete the given tasks. Most importantly, they were able to do so even after a ten-day break from the initial demonstration by the experimenter. While users clearly required help throughout, they were able to get this help mostly from the onscreen audio assistant, which indicates that such systems will likely not require a human expert attendant beyond the initial demonstration. To the best of our knowledge, this is the first articulation of the viability of video mail for illiterate users. Further, the design insights gained in our work also contributes to the growing literature on designing interfaces for this population.

While the use of text-free graphical interfaces for applications focused on illiterate users is not new, our work expands upon the literature by applying these principles into the previously largely unexplored domain of asynchronous personal peer-to-peer communications. Our experience in designing and evaluating this prototype video-mail system clearly showed that there remains much to be learnt in the area of designing interfaces for this population of users, as simple application of previous design principles did not immediately result in a usable system. In particular, it is important to note that although the tasks in both stages of our study using the fully functional final prototype are almost trivially simple from the perspective of seasoned literate

computer users, they are anything but trivial for illiterate users who have never previously used a computer. Despite our best efforts in earlier phases of the work to reduce interface complexity, our study revealed various highly nuanced issues that remain to be solved. Many of these issues would not have manifest themselves in a more literate population, indicating that significant challenges need to be surmounted in order to make even the simplest applications accessible to illiterate users.

Next steps in this work include a limited deployment of a working system to determine if it will actually be used by, and be useful to, the community over an extended period in the field.

8. VIDEO

A video demonstrating the system and aspects of the user study can be found at www.youtube.com/videomailapp

9. ACKNOWLEDGMENTS

We thank all who participated in our studies.

10. REFERENCES

1. Bery, S. (2006). National Council of Applied Economic Research.**Error! Hyperlink reference not valid.**
2. Chand, A. (2002). Designing for the Indian rural population: Interaction design challenges. *Development by Design Conference.*
3. Cooper, A. and Reimann, R. (2003). *About Face 2.0, The Essentials of Interaction Design.* Wiley Publishing Inc. USA.
4. Davis, D., Monrose, F. and Reiter., M. (2004). On user choice in graphical password schemes. *13th USENIX Security Symposium*. p. 151–164.
5. De Angeli, A., Coventry, L., Johnson, G. and Renaud, K. (2005). Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems. *International Journal of Human-Computer Studies,* 63(1-2), p. 128-152.
6. Grisedale, S., Graves, M. and Grunsteidl, A. (1997). Designing a graphical user interface for healthcare workers in rural India. *ACM CHI Conference.* p. 471-478.
7. Huenerfauth, M. (2002). Developing design recommendations for computer interfaces accessible to non-literate users. *Master's thesis, University College Dublin*.
8. Jermyn, I., Mayer, A., Monrose, F., Reiter, M. and Rubin, A. (1999). The design and analysis of graphical passwords. *USENIX Security Symposium*. p. 1-14.
9. Katre, D. (2004). Using mnemonic techniques as part of pictorial interface for self identification of illiterate villagers. *International Conference on Human Computer Interaction, Bangalore, India*.
10. Kumar, R. (2004). eChoupals: A study on the financial sustainability of village internet centers in rural Madhya Pradesh. *Information Technology and International Development.* 2:1. p. 45-73.
11. Lourie, S. (1990). *World literacy: where we stand today - One billion non-literates*. Editorial, UNESCO Courier. July 1990.
12. Medhi, I. Pitti B. and Toyama K. (2005). Text-free UI for employment search. Proc. of *Asian Applied Computing Conference, Nepal*.
13. Medhi, I., Sagar, A. and Toyama K. (2006). Text-free user interfaces for illiterate and semi-literate users. *International Conference on Information and Communication Technologies and Development, Berkeley, USA.* p. 72-82.
14. Medhi, I., Prasad, A. and Toyama K. (2007). Optimal audio-visual representations for illiterate users. *International World Wide Web Conference*. p. 873-882.
15. Medhi, I. and Kuriyan R. (2007). Text-free UI: Prospects for social inclusion. Proc. of *International Conference on Social Implications of Computers in Developing countries, Brazil*.
16. Mitra, S. (2005). Self organizing systems for mass computer literacy: Findings from the hole in the wall experiments. *International Journal of Development Issues,* 4(1). p. 71-81.
17. Moraveji, N., Ho, R., Huynh, D., and Zhang, L. (2005). An exploration in interface design for the Chinese migrant worker population. *Designing for the User Experience, San Francisco, CA.*
18. Moraveji, N., Ho, R., Huynh, D., and Zhang, L. (2005). Communication gap: Designing an interface for Chinese migrant workers. *Usability Professionals' Association, User Experience Magazine*, 5(2).
19. Parikh, T., Ghosh, K. and Chavan, A. (2003). Design considerations for a financial management system for rural, semi-literate users. *ACM CHI Conference.* p. 824-825.
20. Parikh, T., Ghosh, K. and Chavan, A. (2003). Design Studies for a financial management system for micro-credit groups in rural India. *ACM Conference on Universal Usability.* p. 15-22.
21. Parikh, T. (2002). HISAAB: An experiment in numerical interfaces, Media Lab Asia Panel Discussion, *Baramati Initiative on ICT and Development*.
22. Plauche M. and Prabaker M. (2005). Tamil market: a spoken dialog system for rural Indi. *Extended Abstracts of the ACM CHI Conference.* p. 1619-1624.
23. Ratnam, B. V., Reddy P.K., and Reddy, G. S. (2005). eSagu: An IT based personalized agricultural extension system prototype – analysis of 51 Farmers' case studies. *International Journal of Education and Development using Information and Communication Technology*, 2(1). p. 79-94.
24. Ramamritham, K., Bahuman, A., Duttagupta, S., Bahuman, C., and Balasundram, S. (2006). Innovative ICT tools for information provision in agricultural extension. *International Conference on Information and Communication Technologies and Development, Berkeley, USA.*
25. Radicati Group Q3 2007 Market Update: www.radicati.com/uploaded_files/news/Q32007_PR.pdf
26. Ramamritham K., Bahuman A. and Duttagupta S. (2006). aAqua: a database-backended multilingual, multimedia community forum. *ACM SIGMOD Conference.* p. 784-786.
27. Wiedenbeck, S., Waters, J., Birget, J.-C., Brodskiy, A., and Memon, N. (2005). Authentication using graphical passwords: Basic results. *Human-Computer Interaction International Conference.* p. 1-12.

The Inspection of Very Large Images by Eye-gaze Control

Nicholas Adams*, Mark Witkowski and Robert Spence
Department of Electrical and Electronic Engineering
Imperial College London
Exhibition Road
London SW7 2BT
+44 (0)20 7594 6259
*n8vision@gmail.com; {m.witkowski, r.spence}@imperial.ac.uk

ABSTRACT

The increasing availability and accuracy of eye gaze detection equipment has encouraged its use for both investigation and control. In this paper we present novel methods for navigating and inspecting extremely large images solely or primarily using eye gaze control. We investigate the relative advantages and comparative properties of four related methods: Stare-to-Zoom (STZ), in which control of the image position and resolution level is determined solely by the user's gaze position on the screen; Head-to-Zoom (HTZ) and Dual-to-Zoom (DTZ), in which gaze control is augmented by head or mouse actions; and Mouse-to-Zoom (MTZ), using conventional mouse input as an experimental control.

The need to inspect large images occurs in many disciplines, such as mapping, medicine, astronomy and surveillance. Here we consider the inspection of very large aerial images, of which Google Earth is both an example and the one employed in our study. We perform comparative search and navigation tasks with each of the methods described, and record user opinions using the Swedish User-Viewer Presence Questionnaire. We conclude that, while gaze methods are effective for image navigation, they, as yet, lag behind more conventional methods and interaction designers may well consider combining these techniques for greatest effect.

KEYWORDS: *User interaction studies, Visual interaction, Eye-gaze control, Image space navigation.*

1. INTRODUCTION

Recent advances in technology ([6]) have improved our ability to detect and record a user's eye-gaze behaviour, and especially to do so with diminishing discomfort to the user. As a consequence the range and number of applications of this technology have increased rapidly. Two classes of application for eye-gaze detection can be identified. One, which has been of interest for many decades, exploits its investigative potential. Pirolli *et al* [19], for example, employed gaze detection to identify the manner in which users examine web pages, and Cooper *et al* [5] have been able to associate image recognition and user preferences to the nature of eye-gaze trajectories. Other examples include the study of advanced interface design ([11]) and the manner in which visual search is conducted ([4]; [24]).

The other class of application addresses the potential of eye-gaze to control. Many schemes have been proposed, for example, in which the use of eye-gaze replaces or augments human motor processes in circumstances where the use of eye-gaze, alone or with augmentation, can go some way to ameliorating limitations on motor processes experienced by people with disabilities. Gaze control has been established for both disabled and able-bodied users for data input (e.g. [15], [9]), display inspection (e.g. [22]) and spatial navigation (e.g. [3]). In this paper we address another potential application for gaze control – the inspection of large images where gaze controls both zoom and pan.

1.1 Large Images

There are many situations in which very large images must be viewed in the execution of a variety of tasks, at levels of granularity ranging from an overview mode to a study of fine detail. They include the viewing of medical images to identify pathological anomalies (e.g. [16]; [23]), the inspection of large maps such as Google Earth[1] and NASA's World Wind[2] for purposes such as search and aerial surveillance, or the search of astronomical images for a variety of phenomena. Our study is directly concerned with earth images, specifically Google Earth, though we expect our results and conclusions to be relevant to large image inspection in general.

Figure 1: The gaze control system in use

AVI'08, May 28-30, 2008, Napoli, Italy.

[1] http://earth.google.com/

[2] http://worldwind.arc.nasa.gov/

Traditionally, the navigation (i.e., pan and zoom) of large images has been achieved by well-established means of interaction such as mouse or tracker ball control. In this paper we explore the use of eye-gaze – alone and in conjunction with other forms of interaction – to control the actions of panning and zooming in the context of exploring a large image in pursuit of a variety of tasks.

1.2 Related work

Most investigations addressing the use of eye-gaze to zoom into an image have been concerned with the activity known as gaze contingent zooming, and for data-rate reduction [7]. The automatic gaze-controlled expansion of a localised area of a display can help to overcome inaccuracies in gaze detection as well as to enhance the readability of small areas of text on a crowded display ([12]). An interesting extension of these investigations considers local stretching of an area identified by gaze, using a technique known as the bifocal display ([21]). Stretching can either be discrete ([8]) or continuous ([2], [19]). A word of caution, however, was sounded by Gutwin [10] who pointed out that continuous magnification actually slows down focus window targeting: it does so because "*the magnification lens makes windows appear to move in the direction opposite to pointer movement*". Apart from these and similar studies, attention appears to have been confined to situations in which part of a display is at one or the other of two zoom levels, with the possible modification that a localised pre-determined stretching can take place.

By contrast, our study addresses the potential for gaze to control movement of an image through multiple zoom levels ranging, for example, from a view of the entire Earth to a view of a London street. Another word of caution was expressed by Zhai *et al* [25] and is, in a sense, fully acknowledged by our study. Zhai *et al* remarked that "*to load the visual perception channel with a motor control task seems fundamentally at odds with users' natural mental model in which the eye searches for and takes in information and the hand produces output that manipulates external objects….*". The danger to which Zhai draws attention prompts any investigation of gaze control to compare the use of eye gaze alone with eye-gaze employed to augment other interaction modalities.

1.3 Google Earth

The size of some images that must be inspected can be enormous. For example, were the Earth to be imaged at one square metre over its entire surface, the resulting image would have the equivalent of about $5x10^{14}$ pixels. The Google Earth "image" is not yet at this resolution, but is still an impressive size and, moreover, as it is freely available on demand, it was chosen as the image for study. An additional advantage is that the data is convenient, it being both familiar and potentially intuitively navigated by anyone with a rudimentary knowledge of geography.

1.4 Goal of the investigation

Rather than study in some detail one selected means of eye gaze controlled inspection we elected to devise what we felt were a number of promising approaches and then compare them with methods in which eye gaze control was either not used or served to augment another interaction modality. In this sense the investigation was exploratory with the primary aim of providing useful guidance to designers considering similar applications.

2. METHODS OF GAZE CONTROL

In common with other investigators (e.g., [8]), the study we report offers a comparison of eye-gaze on its own (STZ) with two augmented systems (HTZ and DTZ) and with a solely mouse-based system (MTZ) as control.

2.1 Equipment

The system design and investigations described here used LC Technologies (www.eyegaze.com) eye-gaze position monitoring equipment. Gaze position on screen is determined by comparison of the larger retinal ("pupil") reflection, and small corneal reflection (figure 2, centre) from an axially mounted infra-red source on the eye-imaging camera mounted beneath the screen (figure 1 and figure 2, left). The eye image is available in a relatively small volume (approx. 100 mm^3) centred about 70 cm from the screen. The user's eye must remain within this volume for the system to operate. Movements towards or away from the screen cause de-focusing (figure 2, right), which is detected by the system and may, within limits, be used to calculate screen to eye distance. The system requires a brief calibration procedure prior to use by each new user. Accuracy is quoted as 1º (about 15 screen pixels); gaze position readings are made 60 times a second.

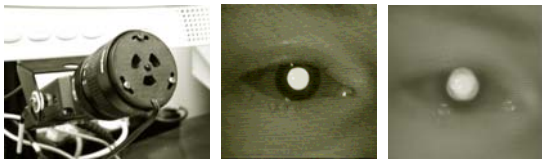


Figure 2: Eye camera (detail) and eye-images

2.2 Stare-to-Zoom (STZ)

In the STZ method all control of pan and zoom is by gaze position and timing. The overall strategy is illustrated in figure 3. The screen is divided into a central zoom region surrounded by a pan region. The extent of the pan region (100 pixels top and bottom, 150 pixels left and right, on a 1024x768 screen) was established empirically, allowing the user sufficient screen space to achieve uninterrupted panning. No zooming takes place while gaze is in the pan region.

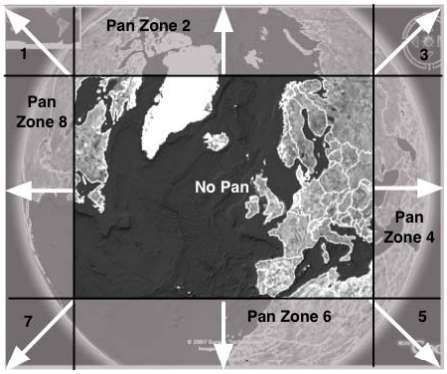


Figure 3: Screen panning regions

Sustained gaze in the central zoom region causes the image to zoom inwards. Normal saccades and fixations in the central region do not cause zooming, so the image may be

inspected in the usual way. However, extended stationary gaze (>420 ms) within the central region causes the image to zoom inwards at a comfortable rate. Zooming continues while the point of gaze remains stationary, as determined by a running calculation of the standard deviation of the screen gaze position. If gaze is fixed within the zoom region but offset from the centre, zooming is also accompanied by panning towards the screen centre. Once the feature of interest is at the centre of the screen, and while gaze is sustained on that feature, zooming continues uninterrupted until maximum resolution is obtained. Zooming outwards is achieved by glancing directly at the camera fixed to the base of the screen (Figure 1).

2.3 Head-to-Zoom (HTZ)

Figure 4 shows the essence of the HTZ technique. HTZ mixes eye-gaze controlled panning with head movement initiated zooming. It was suggested by the intuitive action of leaning forward to examine detail and leaning back to gain an overview (*c.f.* [14]). Small movements (about ±40 mm) of the user's head, detected by the eye-gaze equipment employed, control zoom direction and rate. Pan is controlled by eye gaze fixation: movement of gaze away from the centre of the display causes movement of the image in the appropriate direction, as previously described for STZ.

Zooming is initiated by the system's calculation of eye to screen distance based on de-focusing (section 2.1). To assist the user, and provide fine control of the zooming rate, a non-linear transfer function was adopted, following comments from users during a pilot study [1] prior to the main evaluation reported here. This is illustrated in the insert (figure 4, top). A narrow "dead-band" where no zooming takes place allows for some positioning error by the user, after which slow zooming occurs. This becomes more rapid as the head moves further from the central point (trace III). Unfortunately, the system returns erratic distance estimates once the focus limits are reached and the user must learn to keep within the operating space.

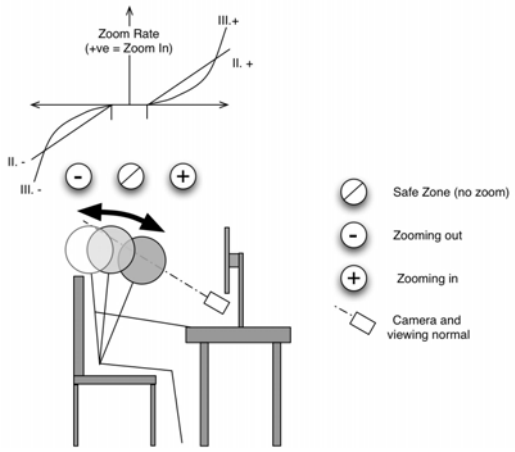


Figure 4: The HTZ zooming mechanism

2.4 Dual-to-Zoom (DTZ)

Following comments as to the care required to use the HTZ method during our pilot investigations, we additionally implemented a Dual-to-Zoom (DTZ) system, which combines gaze position panning input (as STZ and HTZ) with manual zooming using a mouse. The user clicks the left mouse button to zoom in, and the right button to zoom out.

2.5 Mouse-to-Zoom (MTZ)

To provide some means of comparison with which to establish the benefit of gaze control a fourth method of controlling pan and zoom was implemented that did not use gaze control. As with dual control, left and right mouse buttons were used to initiate zooming in and out respectively, and mouse position to control image panning. It should be noted that this method is distinct from the standard mouse control of Google Earth, and it uses the same program control strategies as the other methods.

2.6 Design Issues

The eye-gaze software (supplied) and Google Earth run on a single computer. Control of Google Earth is achieved by a combination of the Google Earth COM API[3] and emulation of keyboard-strokes and mouse clicks; direct view control through the API having been found to be too slow for this type of real time application.

The use of gaze control for cursor movement in this manner necessitates a filter to remove natural eye "jitter", which is highly disruptive to the viewer experience. We created a hybrid filter with a moving average component to stabilise high frequency movements during fixations and intentional looking, and a high-pass component to maintain responsiveness during rapid saccadic movements. Because Google Earth has an inherent centring motion during zooming, we also devised a tracking method that compensated for apparent movement across the screen during extended zooming operations in STZ mode (note also [17] and [18]).

An eye "icon" can be displayed on screen (top-right corner) to assist the user with their head positioning relative to the camera, although the system, by and large, provides its own feedback in terms of pan and zoom. In the trials described here this icon appeared only when tracking was lost, in conjunction with an audible warning, to assist the user to rapidly regain control of the interface.

3. EVALUATION METHOD

Having developed the four methods of pan and zoom control (HTZ, STZ, DTZ and MTZ) described previously, we performed a number of evaluations using Google Earth to assess and compare these methods. In the first evaluation, a search task, we were concerned to evaluate how effectively a user might use each of the methods to locate and identify known targets embedded within the larger image space. The search test also serves to confirm that each of the methods is capable of supporting this important class of browse activity, but also that each of the methods does not cause inadvertent gaze control actions (i.e. panning or zooming) that might interfere with the task. The metric for this task would be how many distinctive objects (in this case London buses) could be located within a test period of 90 seconds. We also monitored zooming activity during this task.

In the second evaluation, a navigation (or tracking control) task, users were asked to zoom directly into a specific location on the earth's surface starting from a "global" view. The purpose of this task was to evaluate and compare how effectively each of the four methods

[3] http://earth.google.com/comapi/

provided a fine degree of control. The primary metric for this task was taken as the deviation of the gaze or control point from the optimal ("shortest" or "direct") path between starting and ending screen images.

In order to perform these tests effectively, users were given a period of time with each of the methods they were to use to practice and browse the earth image freely (the "browse" period). Users were allowed to take the time they required for this familiarisation period. The time they took was noted. Each subject used one of the three gaze-controlled methods (STZ, HTZ or DTZ) and, as a control comparison, the mouse-only method (MTZ). Every participant completed a "subjective" questionnaire about their experiences using each method.

3.1 Familiarisation Activity

Each participant was allowed to use the selected method for an unspecified time ("browsing") to gain familiarity with the control method, being asked to activate a stop key when they were ready to continue with other aspects of the experiment. Actually the time period was interrupted after 180 seconds, but the experimenter restarted the period if requested. The length time the participant elected to use the method was noted as an indication of the time required to become confident with the current method of control (figure 8).

3.2 Navigation/tracking Task

In this task users were asked to perform a straight line navigation from an image of the whole earth to a specific point on the earth's surface by continuous zooming and, if required, corrective panning and zooming actions. Each subject observed an "ideal" pre-programmed navigation from a fully zoomed out view of the Earth to a specific geographical location. Following a repeat of this demonstration the subject was asked to undertake the same navigation task using the selected method. The task was repeated three times to detect any learning or improvement through practice.

Figure 5 shows the starting and required end locations for the task. Two easily recognised end locations were selected; the southern tip of the island of Sicily at the toe of Italy, and the northern most point of Madagascar, off the eastern coast of continental Africa. The starting location varied, but was always within 10º of the target location, which was therefore fully visible. To ensure that the desired target was maintained, and to reduce the cognitive load required to remember and identify the target, a small (constant sized) red dot was drawn over the target location at all times during the test.

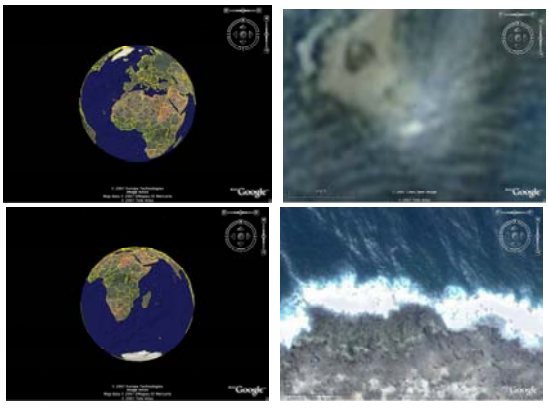


Figure 5: Start and end conditions for navigate task

Data was captured regarding eye gaze and the trajectory followed during execution of the task. The primary measure of performance is taken as the offset between gaze (or mouse) controlled cursor and the position on screen of the target. This is recorded at each time step (16.667 ms, 60 Hz) and measured in screen pixels (0.27 mm). An ideal control strategy would overlap the cursor and target to achieve optimal rates of zooming without the need to pan. Representative offset traces for each of the methods are shown in figure 11. Figure 10 shows the radial offset for a single instance. The time take to complete each instance of the task was recorded (figure 9).

3.3 Search Task

Another method of evaluating the potential of gaze for navigation involved search for a specific type of target. Participants were asked to search for as many London buses as possible within a given time constraint (90 seconds). Two alternate starting locations in central London were selected; a part of Regent Street (figure 6, left), and a part of the Strand just east of Charing Cross railway station (figure 6, right). Each starting image contains five buses, and there are many others in the surrounding area of both starting images (although they are not evenly distributed). Participants were able to zoom in and out or pan around to locate new buses, and were asked to press a clearly indicated key each time they identified a new bus. Participants had all lived in London for a significant time, but were reminded that some buses have white roofs! Again, appropriate data was captured, including the number of buses located, and the amount of zooming in relation to other activities. As with the navigation task, each subject used one of the gaze-controlled methods and, for comparison, MTZ.

Figure 6: The two search task start screens

3.4 Subjective Feedback

A third evaluation elicited the opinions of subjects regarding usability and acceptability. Each of the four methods was evaluated with a "subjective" questionnaire (figure 7), designed to give the experimenters insight into how the users found the experience of using the different methods of control. The 13 questions were designed to provide insight into four aspects of the methods' usability: (a) "presence" (Q1, Q2, Q4, and Q5), the degree to which the user considered themselves immersed in the task; (b) "enjoyment" (Q3, Q6); (c) "sickness" (Q8, Q9, Q10, Q11, Q12), the degree to which they experienced adverse or nauseous sensations while using the system; and (d) "external awareness" (Q7 and Q13), the degree to which the users focussed their attention on the task in hand. Answers, given on a Likert-like numeric scale (shown 1 – 10 in figure 7), were combined in each category and analysed as a whole. The questionnaire used is derived from the Swedish User-Viewer Presence Questionnaire ([13]). Results were analysed with three non-parametric Wilcoxon tests.

Q1: To what extent did you think that the things you did and saw happened naturally and without much mental effort?

not at all 1 2 3 4 5 6 7 8 9 10 extremely

Q2: How natural was the interaction with Google Earth?

not at all natural 1 2 3 4 5 6 7 8 9 10 extremely natural

Q3: To what extent did you find Google Earth fascinating?

not at all fascinating 1 2 3 4 5 6 7 8 9 10 extremely fascinating

Q4: To what extent did you feel you were present in Google Earth?

not at all present 1 2 3 4 5 6 7 8 9 10 extremely present

Q5: How involved were you in the experience?

not at all involved 1 2 3 4 5 6 7 8 9 10 extremely involved

Q6: To what extent did you think it was enjoyable to interact in Google Earth?

not at all enjoyable 1 2 3 4 5 6 7 8 9 10 extremely enjoyable

Q7: To what extent did you focus your attention on the situation, rather than on other things?

not at all 1 2 3 4 5 6 7 8 9 10 extremely

Q8: I felt nauseous

not at all nauseous 1 2 3 4 5 6 7 8 9 10 extremely nauseous

Q9: My eyes felt strained

not at all strained 1 2 3 4 5 6 7 8 9 10 extremely strained

Q10: I had a headache

not at all 1 2 3 4 5 6 7 8 9 10 extremely

Q11: I had problems concentrating

not at all 1 2 3 4 5 6 7 8 9 10 extremely

Q12: I felt unpleasant

not at all 1 2 3 4 5 6 7 8 9 10 extremely

Q13: To what extent were you aware of things happening around you, outside Google Earth?

not at all aware 1 2 3 4 5 6 7 8 9 10 extremely aware

Figure 7: The subjective evaluation questionnaire

4. EXPERIMENTAL PROCEDURE

We asked 32 volunteer participants primarily drawn from the student population (9 female, 23 male, avg. age 24.6 years) to conduct a familiarising browse session, a navigation task and a search task using one of the three gaze control strategies (STZ, HTZ or DTZ), and an equivalent control session using the MTZ method. Participants were also asked to complete the questionnaire relating to their subjective experiences directly after using each of the two methods.

Each experimental session was conducted according to a pre-prepared script to ensure that the conditions under which the measurements were made were as constant as possible, although the experimenter responded to participant questions as necessary. The interaction method (MTZ vs. gaze method) and the two tasks (navigation and search) were counterbalanced. The schedule of activities is as follows:

1) **Introduction:** The experimenter settles the participant, obtains consent, and explains the reasons for the experiment. The experimenter introduces Google Earth and briefly explains that two methods of control will be used, one following the other.

2) **Set-up:** For the STZ, HTZ or DTZ methods the eye-gaze system requires calibration (section 2.1). MTZ does not require calibration. The participant is asked to complete the calibration routine, in which the gaze follows a dot through five screen locations. At this point the participant is asked to keep their head still for the duration of the experiment due to the limited operating volume of the equipment, and the role of the eye indicator (section 2.6) is explained.

3) **Browse:** Using the selected method, the participant is invited to browse with the system until they are ready to continue the experiment. The participant might try to locate the University site, or their home. An automatic timeout sounded at 180 seconds, but the experimenter would continue this activity if requested.

4a) **Search:** The search task was performed once using the selected starting point for a period of 90 seconds, at which time the system ceased operating.

4b) **Navigation:** The selected navigation task is demonstrated twice and then the participant is asked to "navigate as quickly as you can to the point you just saw, when you are done, please say 'OK'". Data recording is automatic.

The order of steps 4a and 4b are determined by the order in which participants used the mouse or eye-gaze method, according to the experiment schedule.

5) **Questionnaire:** The participant is handed the printed questionnaire sheet and asked to choose a value for each of the questions relative to the method they have just used, which the experimenter records.

Steps 2-5 are repeated with the second method. Next, the participant is asked to select the method they preferred.

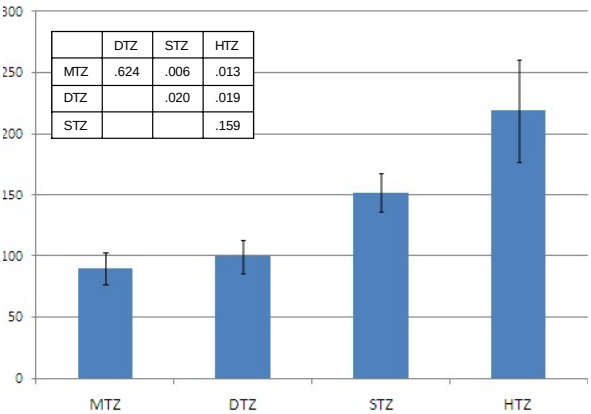
At the conclusion of the experiment the participant is asked to compare the methods used and to make any additional comments they wished, which were recorded by the experimenter. Finally the participants are asked not to share details of the experiment with others and thanked for their help. No reward was made. The complete procedure took approximately 25 minutes per participant and the sessions were conducted over a period of three consecutive days.

5. RESULTS

This section presents the results of the investigations together with some preliminary analysis. Every participant undertook the MTZ method and sufficient results were obtained to have at least 10 instances each of the STZ, HTZ and DTZ methods. In two cases it proved impossible to achieve calibration with the eye gaze equipment, and these results were discarded.

5.1 Familiarisation Activity

Figure 8 summarises the elective time taken by the participant for each of the four methods.



	DTZ	STZ	HTZ
MTZ	.624	.006	.013
DTZ		.020	.019
STZ			.159

Figure 8: Mean browse time (s)

We note that users spent substantially less time to familiarise themselves with the MTZ and DTZ methods than with STZ and HTZ. We surmise that this is due to the

greater familiarity these technically aware participants will have for the mouse based methods, and, in particular, the novelty value associated with the HTZ method, which appears to take more time to get used to. This effect is apparent in later results also. Analysis of variance between means (inset, figure 8, and also figures 9, 10, 13 and 14) using a heteroscedastic *t*-test indicates, within the limits and applicability of this analysis, that there is no significant difference (at the 95% level, two tail) between MTZ-DTZ (p = .624) and between STZ-HTZ (p = .159), but significant differences between the other combinations.

5.2 Navigation Task

Data from the navigation task is analysed both in terms of the overall time taken to complete the task (figure 9), and in terms of the overall offset between the target point and the gaze/mouse controlled cursor place (figures 10, 11 and 12). Mean time to complete indicates that MTZ, DTZ and STZ are closely matched[4], and show little variation between successive trials. MTZ, in particular, demonstrates little variability between participants. However, HTZ shows both a marked increase in mean time to complete and variability in standard deviation (error bars) between trials, but indicates considerable decrease in time between successive trials. It is clear that users found this method less intuitive than the others, and it is tempting to surmise that the rapid improvement is a reflection that users were able to quickly adapt to the requirement to hold the head still in the correct place to achieve smooth and constant zooming along the desired path. Analysis (as section 5.1) shows no significant variation between any of the final attempt times to complete, except between MTZ-DTZ (p = .004).

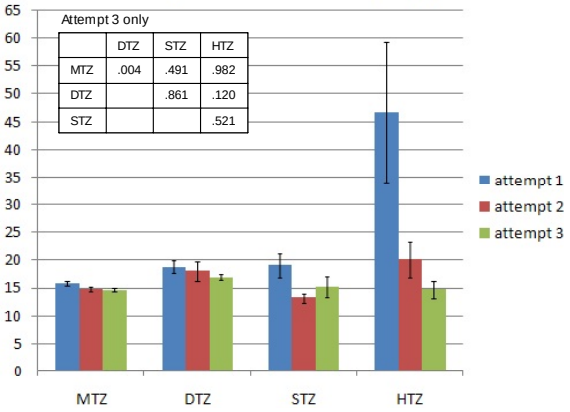


Figure 9: Navigation – Mean time to complete (s)

Figure 10 shows the mean offset from target for each of the four methods. The measure is Euclidean distance (in screen pixels) between target and controlled cursor point. Readings were taken 60 times a second. It may be seen that the MTZ method allows for precise control through positioning of the pointer manually with the mouse. The DTZ and STZ methods are broadly comparable, but less effective. The HTZ method again shows a substantial improvement over the three attempts, although again the final attempt is comparable to DTZ and STZ.

Figure 11 shows four individual traces for each of the tasks. These thumbnails are only intended to convey an impression of the effect, but they are selected to be representative of their type. In the typical MTZ offset trace (figure 11, top left) there is a small initial spike, corrected as the user quickly corrects the initial offset. Tracking is good under manual control until the very last stage when the image becomes highly magnified and offset to one side, which is immediately corrected. In the DTZ trace (top, right) note several peaks during the second attempt. These appear to be due to the user's gaze falling to one side, necessitating a corrective action to rotate the earth image, involving an element of overshoot. Figure 12 shows a radial plot (i.e. target at centre, gaze as offset) of the same trace, apparently confirming the overshoot hypothesis. The STZ plot (figure 11 bottom, left) indicates continued good control, whereas the HTZ (bottom, right) clearly illustrates the difficulty in control of the first attempt, and the rapid improvement in the second and third attempts in both time and accuracy. Note the MTZ third attempt mean is significantly (DTZ) or marginally different (STZ, HTZ) from the others.

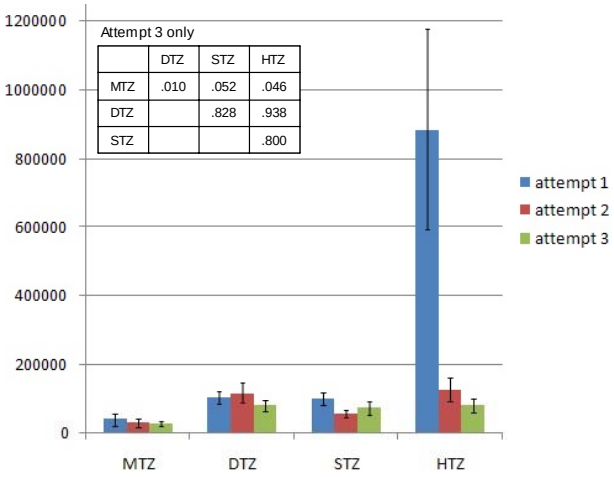


Figure 10: Navigation – Mean total offset from target (pixel distance)

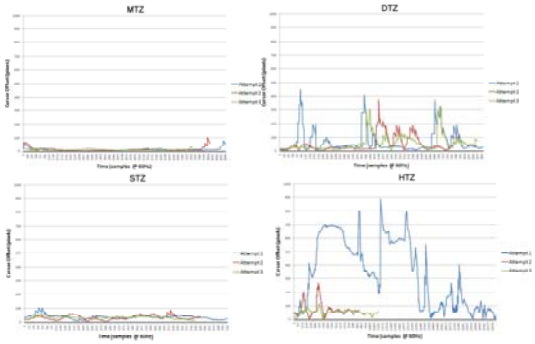


Figure 11: Navigation - Gaze point offset traces

5.3 Search Task

Figure 13 shows the average number of London bus targets found during the search task. The ordering of this result appears to confirm the previous findings, that MTZ offers the highest level of control, followed by DTZ, then STZ, with HTZ proving to be the least effective.

It is perhaps interesting to note from figure 14 that users consistently minimised their use of zooming (expressed as

[4] The MTZ and DTZ times may be overestimated. Some users reported that the zoom rate appeared slower with MTZ and DTZ, and this was later confirmed to be so.

a percentage of total time) when using the STZ method, perhaps indicating that the zoom strategy inherent in this method was less effective than the panning component.

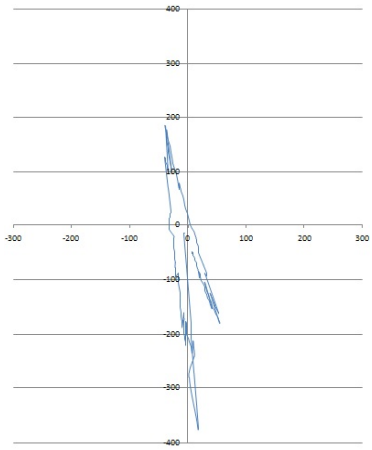
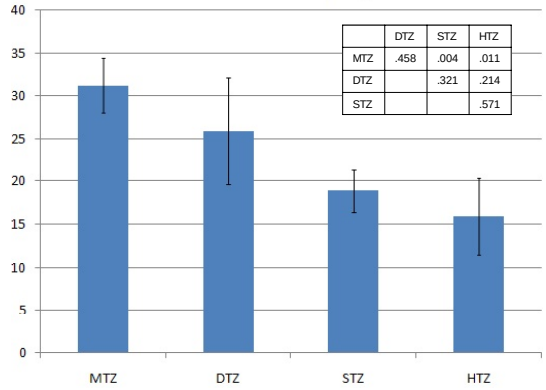


Figure 12: Navigation – Gaze point offset trace



	DTZ	STZ	HTZ
MTZ	.458	.004	.011
DTZ		.321	.214
STZ			.571

Figure 13: Search – Mean number of targets found

	DTZ	STZ	HTZ
MTZ	.304	.000	.616
DTZ		.013	.211
STZ			.002

Figure 14: Search – Mean 'non-zooming' times (% total)

5.4 Subjective Results

When completing the Swedish User-Viewer Presence Questionnaire, participants reported on four measures: presence, enjoyment, external awareness and sickness. The questionnaire responses are summarised in Table 1.

The differences between users' subjective experience in the three eye gaze treatments are revealed by using the non-gaze based MTZ treatment as a reference category. Three non-parametric Wilcoxon tests were performed by pairing the four dependent measures as reported for the mouse (MTZ) with the measures reported for each of the eye gaze methods.

When using the DTZ method, as compared to using MTZ, participants' reports on the measure of presence were non-significant (Z=-1.90, ns). Reports on enjoyment were higher when using the DTZ method as compared to MTZ, although this result was only marginally significant (Z=-1.30, p=0.06). Finally, after interacting with the DTZ eye gaze method, participants' reports on sickness tended to be higher than those given for MTZ but this result was marginally significant (Z=-1.40, p=0.08).

When compared to using the MTZ, participants' reports on presence and enjoyment for the HTZ method were non significant. Conversely, reports on the measure of sickness were higher for the HTZ eye gaze method than the equivalent MTZ treatment (Z=-2.52, p<0.05).

The results yielded in the paired test for STZ and MTZ were identical to that found in the HTZ method: users' reports on presence and enjoyment did not differ in the two conditions, while the sickness measure was higher in the STZ eye gaze method as opposed to the MTZ treatment (Z=-2.31, p<0.05).

7. SUMMARY AND CONCLUSIONS

We have devised several methods for allowing users to browse very large image spaces using either eye-gaze control as a sole method of input, or gaze control combined with other input modalities. We used the publicly available Google Earth image data set and application, as representative of a massive continuous image space, to perform a series of studies to evaluate the effectiveness of these methods relative to a mouse only method.

We were encouraged to find that each method was effective in traversing the image space, although none of the gaze based methods proved as efficient as the more conventional mouse based input. We were also encouraged by the generally positive comments from the test user group, admittedly young and technically aware, who were largely supportive and interested by the possibilities these methods offer. Although our test sample was smaller than we would have liked, we were pleased to note that there were no clear differences or disadvantages to these methods in relation to our "presence", "enjoyment" and "external awareness" criteria. However, each of the gaze methods scored poorly on the "sickness" criteria, and this

Table 1: Summary of subjective responses

	DTZ (N=9)		HTZ (N=10)		STZ (N=11)		MTZ (N=30)	
	M	*S.D.*	*M*	*S.D.*	*M*	*S.D.*	*M*	*S.D.*
Presence	28.89	8.01	24.40	5.74	27.09	6.25	25.40	5.92
Enjoyment	16.89	3.52	13.60	5.56	15.45	4.08	13.73	3.91
External awareness	12.89	2.47	12.10	2.56	10.64	2.80	11.67	2.86
Sickness	12.00	4.15	16.50	8.85	12.00	3.82	7.90	4.11

is a cause for concern. We believe, however, that these are in part due to the relatively short period of familiarisation, and to some limitations inherent in the equipment (notably with the zoom range for HTZ), which might be overcome by expected advances in technology.

The naturalistic search task gives greater variability in results compared to a controlled artificial task, but is more consistent with the study aims. We also note that the methods are more effective in some tasks than others, this requires further investigation. We would also like to undertake a longer study to determine the effects of user familiarisation with each of the novel control methods.

Although effective in its own right, and echoing Sibert & Jacob's [20] view that "*Eye gaze interaction is a useful source of additional input and should be considered when designing interfaces in the future*", we suspect that eye-gaze control will also serve well as a way of augmenting more conventional input methods as indicated by our HTZ and DTZ methods. It has great potential to make interfaces more responsive and better able to anticipate the intentions of increasingly sophisticated interface users.

Assuming modest improvements in eye-gaze measurement technology and techniques – as well as greater availability – we are encouraged that both "hands-free" methods (STZ and HTZ) offer a viable image control and search method, notably for those with severe motor disability, but also for those who routinely monitor and search large image spaces and wish to use their hands for other tasks, such as data entry. Clearly such navigation methods might equally be applied to scanning and traversing three dimensional image sets, such as tomographic scans or architectural designs, and this remains a future task for investigation.

8. ACKNOWLEDGEMENTS

The authors would like to express their thanks to Asimina Vasalou (Imperial College London) and Päivi ("Curly") Majaranta (University of Tampere, Finland) for their generous assistance during the development of this paper.

9. REFERENCES

[1] Adams, N., Witkowski, M. and Spence, R. (2007) The Exploration of Large Image Spaces by Gaze Control, Proc. COGAIN-07, 78-81.

[2] Ashmore, M., Duchowski, A. T., & Shoemaker, G. (2005) Efficient eye pointing with a fisheye lens, Proc. Graphics Interface GI-2005, 203-210.

[3] Bates, R. and Istance, H. (2005) Fly Where You Look: Enhancing Gaze Based Interaction in 3D Environments, Proc. COGAIN-05, 30-32.

[4] Bertera, J.H. and Rayner, K. (2000) Eye Movements and the Span of the Effective Stimulus in Visual Search, *Perception & Psychophysics*, **62**(3), 576-585.

[5] Cooper, K., de Bruijn, O., Spence, R. and Witkowski, M. (2006) A Comparison on Static and Moving Presentation Modes for Image Collections, Proc. AVI-06, 381-388.

[6] Duchowski., A.T. (2003) *Eye Tracking Methodology: Theory & Practice*. Springer-Verlag, London, UK.

[7] Duchowski, A.T., Cournia, N. and Murphy, H. (2004) Gaze-Contingent Displays: A Review, *Cyberpsychology & Behavior*, **7**(6), 621-634.

[8] Fono, D. & Vertegaal, R. (2005) EyeWindows: evaluation of eye-controlled zooming windows for focus selection. Proc. CHI-05, 151-160.

[9] Gips, J. and Olivieri, P. (1996) EagleEyes: An Eye Control System for Persons with Disabilities, Proc. 11th Int. Conf on Technology and Persons with Disabilities, 13pp.

[10] Gutwin, C. (2002) Improving Focus Targeting in Interactive Fisheye Views, Proc. CHI-02, 267-274.

[11] Jacob, R.J.K. (1995) Eye Tracking in Advanced Interface Design, in: Barfield, W. and Furness, T.A. (eds.) Virtual Environemts and Advanced Interface Design, New York:Oxford University Press, 258-288.

[12] Kumar, M., Paepcke, A. and Winograd, T. (2007) EyePoint: Practical Pointing and Selection Using Gaze and Keyboard, Proc. CHI-07, 421-430.

[13] Larsson, P., Västfjäll, D., and Kleiner, M. (2001). The Actor-observer Effect in Virtual Reality Presentations. *CyberPsychology and Behavior*, **4**(2), 239-246.

[14] Lepinski, G.J. and Vertegaal, R. (2007) Using Face Position for Low Cost Input, Long Range and Oculomotor Impaired Users, Proc. COGAIN-07, 71-73.

[15] Majaranta, P. and Räihä, K.J. (2002) Twenty Years of Eye Typing: Systems and Design Issues, Proc. ETRA-02, 15-22.

[16] Mello-Thomas, C. (2003) Perception of Breast Cancer: Eye-Position Analysis of Mammogram Interpretation, *Acad. Radiol*., **10**, 4-12.

[17] Miniotas, D. and Špakov, O. (2004) An Algorithm to Counteract Eye Jitter in Gaze-Controlled Interfaces. *Information Technology and Control*, **1**(30), 65–68.

[18] Miniotas, D., Špakov, O. and Scott MacKenzie, I. (2004) Eye Gaze Interaction with Expanding Targets, Proc. CHI-04, 1255–1258.

[19] Pirolli, P., Card, S.K. and van der Wege, M.M. (2000) The Effect of Information Scent on Searching Information Visualizations of Large Tree Structures, Proc. AVI-00, 161-172.

[20] Sibert, L.E. and Jacob, R.J.K. (2000) Evaluation of Eye Gaze interaction, Proc. CHI-00, 281-288.

[21] Spence, R. and Apperley, M.D. (1982): Data Base Navigation: An Office Environment for the Professional. *Behaviour and Information Technology*, **1**(1), 43-54.

[22] Starker, I. and Bolt, R.A. (1990) A Gaze Responsive Self-Disclosing Display, Proc. CHI-90, 3-9.

[23] Tiersma, E.S.M., Peters, A.A.W., Mooij, H.A. and Fleuren, G.J. (2003) Visualising Scanning Patterns of Pathologists in the Grading of Cervical Intraepithelial Neoplasia, *J. Clin. Pathol*., **56**, 677-680.

[24] Zelinsky, G.J. and Sheinberg, D.L. (1997) Eye Movements During Parallel-Serial Visual Search, *J. Exp. Psychol.: Human Perception and Performance*, **23**(1), 244-262.

[25] Zhai, S., Morimoto, C. and Ihde, S. (1999) Manual and Gaze Input Cascaded (MAGIC) Pointing, Proc. CHI-99, 246-253

Evaluation of Pointing Performance on Screen Edges

Caroline Appert[1,2,3]
appert@lri.fr

Olivier Chapuis[2,3]
chapuis@lri.fr

Michel Beaudouin-Lafon[2,3]
mbl@lri.fr

[1]IBM Almaden Research Center
San Jose, CA, USA

[2]LRI - Univ. Paris-Sud & CNRS
Orsay, France

[3]INRIA
Orsay, France

ABSTRACT

Pointing on screen edges is a frequent task in our everyday use of computers. Screen edges can help stop cursor movements, requiring less precise movements from the user. Thus, pointing at elements located on the edges should be faster than pointing in the central screen area. This article presents two experiments to better understand the foundations of "edge pointing". The first study assesses several factors both on completion time and on users' mouse movements. The results highlight some weaknesses in the current design of desktop environments (such as the cursor shape) and reveal that movement direction plays an important role in users' performance. The second study quantifies the gain of edge pointing by comparing it with other models such as regular pointing and crossing. The results not only show that the gain can be up to 44%, but also reveal that movement angle has an effect on performance for all tested models. This leads to a generalization of the 2D Index of Difficulty of Accot and Zhai that takes movement direction into account to predict pointing time using Fitts' law.

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: Information Interfaces and Presentation (e.g., HCI) – User Interfaces, Input Devices and Strategies

Keywords

Edge pointing, Screen Edges, Fitts' Law, performance modelling

General Terms

Human Factors, Experimentation, Performance

1. INTRODUCTION

Common graphical desktop environments display a number of interactive widgets along the physical edges of the screen. Microsoft Windows© and several X Window environments, e.g., GNOME and KDE, feature a *task bar*. This task bar contains buttons to navigate among application windows, and shortcuts to the files and applications used most often. It is also used to display notification icons, current time, sound controls or system status. Mac OS X© displays the menus of the foreground application and some notification icons in a *menu bar* that is always at the top of the screen. It also features a *dock* holding icons to quickly launch frequently used files and applications. Users can change the task bar or dock location but the system constrains them to one of the four physical edges of the screen (three for Mac OS X because of the menu bar).

Placing widgets along the edges makes it easier for users to organize their workspace, i.e., their windows and icons, in the central area of the screen without occluding these widgets. However, it also maximizes the distance between the working area and these "edge widgets". Since Fitts' law [9] predicts that the larger the distance between the cursor and a target, the longer the time to reach that target, edge widgets may impede pointing performance.

While pointing in the central screen area has been extensively studied and Fitts' law has been shown to hold in most cases, e.g. [8, 16], the situation with targets located on screen edges may be different. A typical pointing movement is composed of two main phases: an acceleration phase at the beginning of the movement and a deceleration phase at the end of the movement to stop the cursor within the target bounds [19]. Figure 1 (left) shows the typical profile of the speed curve for pointing at a "regular" target. When pointing at an edge target, however, users can take advantage of the physical boundary to stop the movement. They only have to stay within the bounds of the target along the direction collinear to the edge, while maintaining a high speed (prone to overshooting) along the main movement direction. Figure 1 (right) shows the expected speed curve when pointing at a target on a screen edge. Accordingly, edge pointing should be faster than "regular" pointing. The intuition that pointing at edge widgets should be faster has already been noted, e.g., [18], Chapter 4, or [5], but, to the best of our knowledge, it has never been empirically tested. Thus, we do not know if users can perceive the potential advantage and actually use edges in practice.

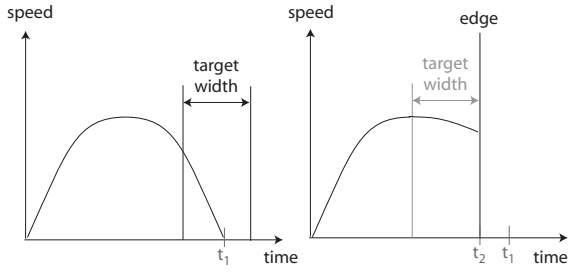


Figure 1: Speed curves for regular pointing (left) and for edge pointing (right).

AVI'08, 28-30 May, 2008, Napoli, Italy

In this article, we present two experiments to better understand edge pointing and help interface designers in their desktop layout choices. The first experiment identifies the relevant factors involved in an edge pointing task and measures their effects on mouse movements and pointing performance. Results show that some factors such as cursor shape or movement direction have an impact on completion time and the use of edges. The second experiment quantifies the gain of using edges by comparing edge pointing with regular pointing and crossing [1]. It shows that movement angle has a strong effect on performance in all three cases and that differences between models increase with angle. We then propose a generalization of the 2D Index of Difficulty of Accot and Zhai [2] that captures the relation between pointing difficulty and movement direction to provide better predictions of pointing performance.

2. RELATED WORK

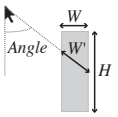
Regular pointing has been extensively studied and providing a full review of the literature is beyond the scope of this article. Since we are interested in identifying relevant factors that influence completion time of an edge pointing task, we give an overview of factors that have already been tested in regular pointing and the main findings of these studies.

The most common way of studying pointing is to measure movement time (MT) according to Fitts' Index of Difficulty (ID) on a one-dimensional pointing task [9]. Fitts' ID is a function of the ratio of two other factors: the distance to the target (D) and the width of the target (W):

$$MT = a + b \cdot ID, \quad \text{where} \quad ID = \log_2\left(\frac{D}{W} + 1\right)$$

This law means that the larger and the closer the target, the shorter the time required to point at it. Numerous studies have validated this model, see [14] for a review.

Over the past fifteen years, a number of studies have attempted to refine this model by taking into account other factors that might influence pointing performance in a realistic two dimensional environment. Since many targets are rectangular, a number of models of 2D pointing have been proposed. For example, MacKenzie and Buxton compared several models [15] and found that $ID_{W'}$ and ID_{min} were the most promising, with ID_{min} providing slightly better predictions :



$$ID_{W'} = \log_2\left(\frac{D}{W'} + 1\right) \quad (1)$$

$$ID_{min} = \log_2\left(\frac{D}{min(W,H)} + 1\right) \quad (2)$$

Accot and Zhai [2] criticized the similar importance attributed to target width and height in these models. They proposed a more complex model, noted ID_{az} in this article, that assigns a specific role to each of these two dimensions, and showed that it provides better predictions. They define target width as the side collinear to the movement direction, i.e., the amplitude constraint, and target height as the side orthogonal to the movement direction, i.e., the directional constraint. Each of these dimensions makes its own contribution to the task difficulty. In their study, $p = 2$, $\omega = 1$ and $\eta = \frac{1}{7.3}$ were the best values for the three free parameters:

$$ID_{az} = \log_2\left(\left[\omega\left(\frac{D}{W}\right)^p + \eta\left(\frac{D}{H}\right)^p\right]^{\frac{1}{p}} + 1\right) \quad (3)$$

Regarding movement direction, the ISO9241-9 standard for evaluating pointing devices [12] recommends to lay out targets in a circular pattern and to impose a specific order of appearance that forces participants to perform movements in every direction to obtain results that are valid whatever the movement direction. However, some studies have attempted to isolate and measure the effect of movement angle on completion time. Mackenzie and Buxton [15] used three different angles (0, 45 and 90 degrees) and found that moves along the horizontal and vertical axes were about the same while moves along the diagonal axis took 4% longer. Grossman and Balakrishnan [10] tested angles 0, 22.5, 45, 67.5 and 90 degrees and found that users were the fastest in horizontal movements. To our knowledge, the studies that have tested a wider range of angles have not given more fine-grained results. For example, Whisenand and Emurian [20] found that diagonal movements were slower than straight movements and that horizontal movements were the fastest. Hancok and Boot [11] and Boritz et al. [7] also tested angles all around the cursor with both left and right-handed users and found that movements to the right were the fastest with the right hand for right-handed users and a symmetric result for left-handed users.

Finally, a few studies have measured the effect of other factors such as target feedback or cursor shape. Akamatsu et al. [3] compared five different sensory feedback conditions (no feedback, auditory, colour, tactile, and a combination of the three). They found that feedback of any type decreases the final positioning time (between entering the target an selecting it) but has no significant effect on overall completion time. Regarding cursor shape, Po et al. [17] compared a circle cursor and four arrow cursors (upper-left, upper-right, lower-left and lower-right) and showed that (i) an arrow cursor is more efficient when it is oriented in the direction of movement and that (ii) a circle cursor is the most efficient on average and its performance is independent of the movement angle.

3. STUDY 1: RELEVANT FACTORS

The goal of Experiment 1 was to measure the effect of variables involved in an edge pointing task. First, since it is a pointing task, we tested the effect of two common variables: Index of Difficulty (ID) and movement angle. Second, we tested the effect of variables that can help users *feel* the edges. While interacting with a direct input device such as a stylus provides a physical feedback of screen edges, indirect input devices such as a mouse, which are commonly used with desktop interfaces, do not have this property. Let us see what happens in today's standard desktop interfaces.

The standard arrow cursor, which points toward the upper left, leads to a situation where only one pixel (the tip of the arrow) is (barely) visible when the cursor is located at the very bottom or right edge of the screen (Figure 2-c). This could reduce the potential gain of using edges since users have to perform a visual search to make sure that they are on the right target (and possibly additional movements to locate the cursor). This probably explains why, in such situations, additional feedback is added to the target under the cursor. For example, on Windows XP, the task bar icon under the cursor is slightly highlighted. On the contrary, Macintosh menus, which are always located at the top of the screen do not provide any additional feedback (until the mouse is clicked) since the cursor remains completely visible even when moved as far up as possible (Figure 2-b).

In this experiment, we considered three factors specific to edge pointing. First, we considered targets on the top (North) and bottom (South) edges. While the task bar and the Mac OS X dock can

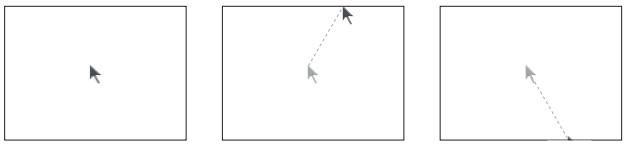


Figure 2: Edge and cursor visibility.

be on the left or right edges as well, we omit these cases to simplify the design and leave them for future work. The important point is that we have a condition where the arrow cursor almost disappears (South) and one where it is always visible (North). The second factor is target feedback: either targets are highlighted when the cursor is over them, or they are not. The third factor is the cursor type. We tested the traditional *arrow* cursor as well as a *circle* cursor. The *circle* cursor is symmetric and its hotspot is at its center. It remains visible whichever edge it is pushed against (Figure 3). This factor will help assess whether the observed effects of *Angle* and *Edge* are due to the *arrow* cursor orientation [17].

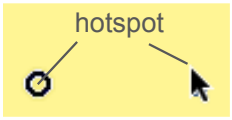


Figure 3: Circle cursor and Arrow cursor.

3.1 Apparatus

The experiment was run on a 2.66 GHz bi-processor PC running Linux with a Nvidia Quadro FX 1000 graphics card connected to a 1680 × 1050 LCD display (99 × 98 dots per inch). We used a standard optical mouse with the default linear X Window acceleration function. Our program was implemented in Java using the Touchstone run platform [13] and the SwingStates Toolkit [4].

3.2 Subjects

Twelve unpaid adult volunteers (11 male, 1 female), from 24 to 34 year-old (average 27.92, median 27), all right-handed, served in the experiment.

3.3 Task and Experiment design

Our experiment was a $2 \times 2 \times 2 \times 4 \times 7$ within-participant design. The following list summarizes the factors we tested:

- 2 *Cursor* conditions: *arrow* and *circle*,
- 2 *Feedback* conditions: *highlight* and *none*,
- 2 *Edge* conditions: *top* edge or *bottom* edge,
- 4 *Width* conditions: 20, 50, 100 and 200 pixels,
- 7 *Angle* conditions: -90, -60, -30, 0, 30, 60 and 90 degrees.

We used different angles and target widths to study edge pointing in a realistic context of use. -60, -30, 0, 30 and 60 degrees cover a good range of situations when the user is working at arbitrary screen locations. We also included -90 and 90 degrees to represent the frequent situation where a user moves the cursor horizontally to switch among window icons in a task bar or to explore different menus in the menu bar. For target widths, 20 pixels is roughly the size of a notification item while 50, 100 and 200 pixels represents a range of sizes for icons in the task bar or menus in the Mac OS X menu bar. To limit the number of trials, we used a fixed distance of 500 pixels and a fixed height of 20 pixels for the target (which is the typical height for a menu item or an icon in the task bar). Figure 4 illustrates the simple task participants had to perform: first click on a circular starting point and then click on the target. The next trial started only when the participant had clicked the target, i.e. every trial had to end successfully even if it included clicks outside the target.

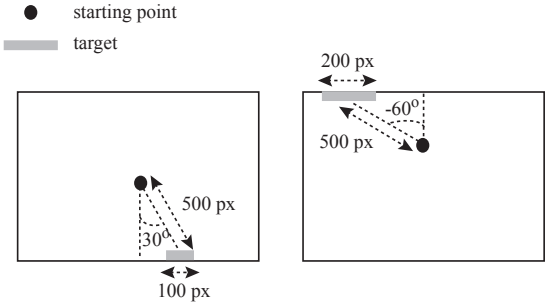


Figure 4: Two instances of the task used in Experiment 1. Left: $Angle = 30$, $Width = 100$, $Edge = bottom$. **Right:** $Angle = -60$, $Width = 200$, $Edge = top$.

We grouped trials into blocks according to the $Cursor \times Feedback$ condition, each block containing the 56 combinations $Edge \times Width \times Angle$. To counterbalance the presentation order of $Cursor \times Feedback$ blocks, we used a Latin Square to compute 4 presentation orders per participant, resulting in a design containing 4 trial replications per participant[1]. Each participant thus executed 16 blocks. Each block was divided into two series, one per *Edge* condition. We divided our participants into two groups of six participants. Participants in the first group performed these series in the order *top* then *bottom* while participants in the second group performed them in the order *bottom* then *top*. Within a series, participants had to perform 28 trials per $Width \times Angle$ condition, presented in a random order ($4 \times 7 = 28$ trials). Thus, the total number of logged trials in our experiment was: 16 blocks × 2 series × 28 trials × 12 participants = 10752 trials. Before starting the experiment, participants were instructed to point as fast and as accurately as possible and had to perform a series of trials with a sample of all the conditions they would face during the experiment.

Our software collected three main measures: completion time, number of "errors" (clicks outside the target) and click position.

3.4 Predictions

Before running the experiment, we made the following predictions:

H_1: *circle* cursor is more efficient than *arrow* cursor for $Edge = bottom$ because of the visibility problem caused by *arrow* cursor.

H_2: $Feedback = highlight$ is more efficient than $Feedback = none$ especially in condition $Cursor = arrow \times Edge = bottom$. Feedback should help users quickly perceive that they are in the target, making them more confident and avoiding a costly visual search for cursor location.

H_3: As in regular pointing, the larger the target, the shorter the completion time. Since we use a single target height and hypothesize

[1]Note that we do not use the same Latin Square for each participant so as to ensure that within a group of 6 participants, the $4! = 24$ possible orders are presented.

that participants will use edges to stop their movement, completion time should be a linear function of $ID_e = \log_2(1 + \frac{D}{Width})$. Contrary to Accot and Zhai [2] who defined width and height according to movement angle, we consider that target width is always the length of the target side collinear to the screen edge[2].

3.5 Results

We collected a total of 10752 trials. 690 of them included clicks outside the target (error rate = 6.41%). We did not remove these trials for our analyses since participants had to end each task successfully (errors are thus included in task completion time as a penalty).

There was a significant learning effect: *Block number* has a significant effect on completion time ($F_{31,341} = 2.4$, $p < 0.001$) and completion time decreases according to *Block number*. This should not affect the validity of our analyses since our counterbalancing strategy ensured that each condition appeared in every position across participants. This is supported by an analysis of variance that did not reveal an effect of presentation order on completion time: the interaction effect *Block number* $\times$ *Cursor* $\times$ *Feedback* on completion time is not significant.

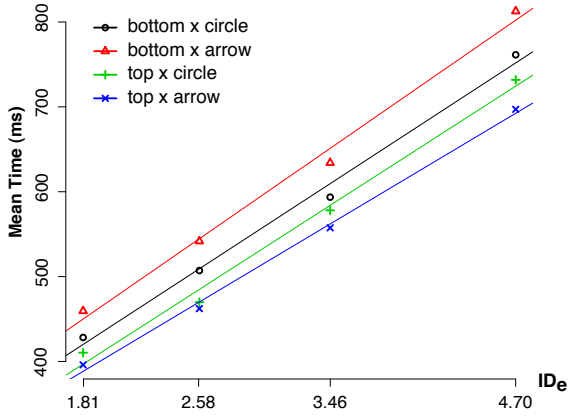


Figure 5: Mean time as a function of ID_e by *Edge* $\times$ *Cursor*

Analysis of variance did not reveal a significant effect of *Cursor* on task completion time but revealed a strong interaction effect of *Edge* $\times$ *Cursor* on task completion time ($F_{1,11} = 65.2$, $p < 0.0001$). Tukey post hoc tests showed that *bottom* $\times$ *circle* is significantly faster than *bottom* $\times$ *arrow* (a difference in mean of 40 ± 6 ms representing a speed up of 6.5%) and that *top* $\times$ *arrow* is significantly faster than *top* $\times$ *circle* (a difference in mean of 19 ± 6 ms representing a speed up of 3.4%). We found no significant difference between *bottom* $\times$ *circle* and *top* $\times$ *circle*. These results support hypothesis H_1. H_3 is also supported since we observed a significant simple effect of ID_e on task completion time ($F_{3,33} = 262.8$, $p < 0.0001$). Figure 5 illustrates these results.

Hypothesis H_2 however is rejected since there was no significant effect of *Feedback* on task completion time (nor any significant interaction effect of *Feedback* with any other factor on completion time). Analyses of the number of errors revealed that *Feedback* has a significant effect on number of errors ($F_{1,11} = 20.1$, $p = 0.0009$) with participants making significantly more errors in the *Feedback=highlight* condition (6.5%) than in the *Feedback=none* condition (5.1%). It seems that providing feedback by highlighting the object under the cursor is more disturbing than helpful in our experimental task. This is consistent with Akamatsu et al. [3] who observed no significant effect of feedback on completion time. We also observed a significant effect of ID_e on number of errors ($F_{3,33} = 12.8$, $p < 0.0001$). This is not surprising since clicking in a small target requires more precision than clicking in a large one. A linear regression of completion time as a function of ID_e, $MT = 208 + 114.ID_e$, shows a high correlation with an adjusted $r^2 = 0.992$ (we consider here 4 mean completion times per ID_e).

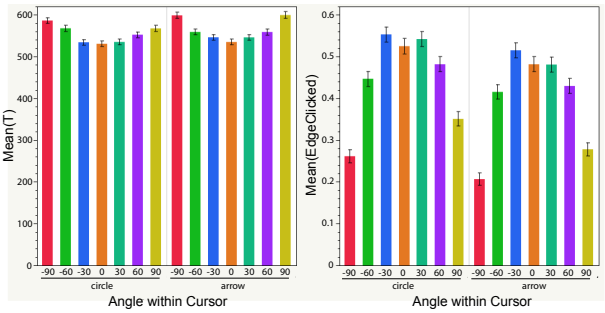


Figure 6: Mean time (left) and mean number of clicks along the edge (right) by *Angle* $\times$ *Cursor*

Our analyses showed that movement direction is an important factor for edge pointing. First, participants were faster at pointing upward than downward since we observed a significant effect of *Edge* on task completion time ($F_{1,11} = 45.8$, $p < 0.0001$). Second, performance varies according to the movement angle: *Angle* has a significant effect on task completion time for both edges ($F_{6,66} = 11.8$, $p < 0.0001$ for $Edge = bottom$ and $F_{6,66} = 4.1$, $p < 0.0014$ for $Edge = top$). This is probably because it is easier to use edges to stop when the movement is orthogonal to the edge, i.e. when *Angle* is close to 0. Comparing the mean completion time and the mean number of times users stopped on an edge according to the angle of movement supports this interpretation (Figure 6): participants stop more often on an edge for angles close to 0. Note that these results differ from those on regular pointing, in which users are faster in horizontal movements than in vertical ones [10, 15, 20].

We also observed an *Angle* $\times$ *Cursor* interaction effect on completion time ($F_{6,66} = 6.2$, $p < 0.0001$ for $Edge = bottom$ and $F_{6,66} = 2.7$, $p = 0.0202$ for $Edge = top$). This interaction effect seems stronger for $Edge = bottom$ probably because this edge suffers from the cursor visibility problem. Finally, there was a significant *Angle* $\times$ ID_e interaction effect ($F_{18,198} = 3.7$, $p < 0.0001$ for $Edge = bottom$ and $F_{18,198} = 4.1$, $p < 0.0001$ for $Edge = top$). The comparison of mean completion times across *Angle* $\times$ ID_e conditions revealed that performance is less sensitive to angle for easy pointing tasks, i.e., low IDs, than for difficult ones.

This first experiment revealed that edge pointing exhibits some differences with previous results on regular pointing, especially regarding the effect of movement angle. This is a motivation to further study edge pointing in order to better understand its underlying model and compare it with other models for target selection.

[2]Actually, using Accot and Zhai's definitions of width and height would have been confusing since we would have had to swap these two variables according to the value of the *Angle* factor. Indeed, Accot and Zhai use two movement angles (vertical and horizontal) and define target height as the directional constraint and target width as the amplitude constraint. In our case, the range of angles is much larger so one target dimension cannot be mapped directly to one of these constraints, as illustrated by Figure 12.

4. STUDY 2: PERFORMANCE GAIN

The goal of Experiment 2 is to identify a model for edge pointing. Our approach consists in comparing edge pointing with well-known models such as regular pointing or crossing, as well as a model that has not been studied yet, i.e., pointing a *Semi-Infinite* target, and that we hypothesize to be close to edge pointing.

4.1 Candidates for a model

Regular Pointing. In this model, based on Fitts' law, the user has to stop within the bounds of a finite target and click to select it. Edge pointing follows this model if users do not use edges to stop their movement.

Crossing+Click. Crossing was introduced by Accot and Zhai [1]: A target is a segment, and selection consists in overshooting the target with the pen down. Accot and Zhai showed that crossing a segment whose width is W can be more efficient than pointing a target of width W. Crossing follows Fitts' law but has lower empirical coefficients (a and b) when the segment is orthogonal to the movement direction. Crossing and edge pointing share the following property: the user does not have to perform the last part of the movement which consists in precisely stopping within the target. While crossing seems a good candidate to model edge pointing, crossing does not require a click to select the target (the selection is completed as soon as the user has crossed the segment). Therefore, we compare a variant of crossing that we call *Crossing+Click* which consists in first crossing the target and then clicking to actually select it. A pilot experiment revealed that it was hard for participants to know which target side they had to cross and to be sure that they had actually crossed it when the target was on the edge. Thus, in this experiment, we used a black line to indicate which side to cross (Figure 7-a) and the target was highlighted as soon as it had been crossed.

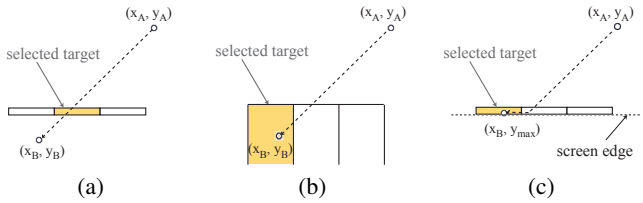


Figure 7: Selection by crossing (a), by pointing a semi-infinite target (b) and by edge pointing (c).

Semi-Infinite Pointing. A close look at current implementations of edge pointing in standard desktop environments shows that mouse movements along the x-axis are still taken into account once the top or bottom of the display is reached. We therefore introduce semi-infinite pointing. Figure 7 illustrates the difference with crossing. If a target is selected by crossing, only the position on the x-axis at crossing time is taken into account. This means that if the cursor has a diagonal trajectory and its speed would make it stop further along the edge, the part of the movement beyond the edge is ignored. On the contrary, if a target is selected by edge pointing, it is the x-position of the cursor when the click occurs that is taken into account to determine which target is selected. Therefore, in an edge pointing task, targets can be seen as semi-infinite, i.e., they are not bounded along the orthogonal direction of the edge. As mentioned earlier, pointing at targets with various W/H ratios has already been studied but only on a limited set of angles (0, 45 and 90 degrees in [15] and 90 degrees in [2]). Each study yielded a formula (ID_{min} and ID_{az}) that does not include the angle of movement.

We hypothesize that edge pointing is close to pointing a semi-infinite target, i.e., pointing a target with a W/H ratio close to zero, and that both models (ID_{min} and ID_{az}) do not capture this configuration properly since Experiment 1 has revealed a significant effect of angle of movement on movement time in edge pointing.

4.2 Task and Experiment design

We used the same hardware and software as in Experiment 1. Eight participants, all having already completed Experiment 1, also served in Experiment 2. The task also consisted in selecting a target but under different model conditions (Figure 8).

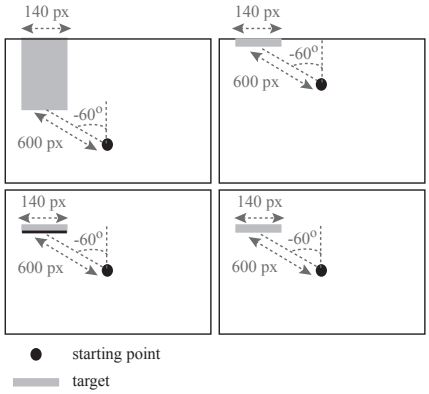


Figure 8: The 4 *Model* conditions for a given *Edge* × *Width* × *Distance* × *Angle* condition. Clockwise from upper-left: *Semi-Infinite*, *Edging*, *Pointing* *Crossing+Click*.

To limit the length of the experiment and focus on the study of the underlying model, we did not include *Feedback* and *Cursor* as factors in this experiment. Participants had to perform target acquisition tasks with a circle cursor and no feedback. This allowed us to study a wider range of ID. Our experiment was a $4 \times 2 \times 3 \times 2 \times 7$ within-participant design with the following factors:

- 4 *Model*: *Pointing*, *Crossing*, *Semi-Infinite* and *Edging*,
- 2 *Edge*: *top* edge or *bottom* edge,
- 3 *Width*: 35, 70 and 140 pixels,
- 2 *Distance*: 300 and 600 pixels,
- 7 *Angle*: -90, -60, -30, 0, 30, 60 and 90 degrees.

The trials were grouped into 12 blocks, 4 *Model* conditions repeated 3 times. Each *Model* block was divided into two sub-blocks, one per *Edge* condition and each of these sub-blocks contained 3 *Width* × 2 *Distance* × 7 *Angle* = 42 trials. The target height was 320 pixels in the *Semi-Infinite* condition while it was 20 pixels in all other conditions. To counterbalance the presentation order of conditions, we created 4 groups of 2 participants and computed 12 presentation orders for the *Model* condition using three Latin Squares. We concatenated 3 orders to compose a sequence of 12 blocks so we obtained 4 sequences, one per group of two participants. Within a group, one participant saw this sequence with sub-blocks in the order $Edge = bottom$ then $Edge = top$ while the other participant saw this sequence with sub-blocks in the order $Edge = top$ then $Edge = bottom$. Finally, the 42 trials of a *Model* block were presented in a random order. To summarize, the total number of logged trials in our experiment was: 12 blocks × 2 sub-blocks × 42 trials × 8 participants = 8064 trials. As in Experiment 1, participants were instructed to acquire the target as fast and as accurately as possible and had to perform a series of trials with a sample of all the conditions before starting the experiment.

4.3 Results

Before analyzing the results, we first checked that participants did not use the physical edge of the screen in the *Semi-Infinite* condition in order to avoid a confound with the *Edging* condition. The cursor reached the edge in only 0.34% of the trials and 99% of mouse clicks occurred within the first 250 pixels of the 320-pixels target.

Learning effect and error rate (6.05%) were similar to the ones observed in Experiment 1. Here again, our design counterbalanced learning effects since we did not observe a significant interaction effect of *Block number* $\times$ *Model* on completion time.

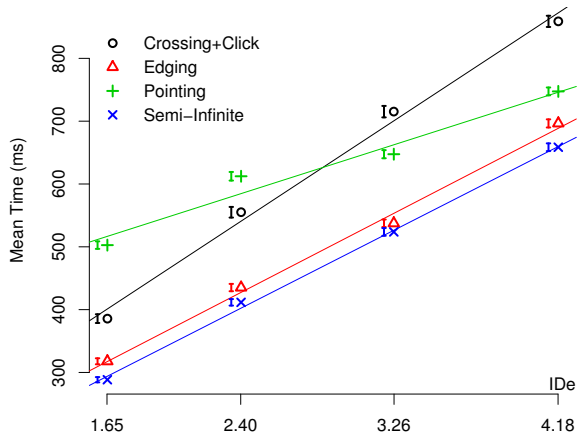


Figure 9: Mean time as a function of ID_e by *Model* (error bars are shown to the left of each symbol).

Analysis of variance revealed a significant effect of *Model* ($F_{3,21} = 141.3$, $p < 0.0001$) and ID_e ($F_{3,21} = 440.3$, $p < 0.0001$) on task completion time. We also observed a significant *Model* $\times$ ID_e interaction effect on task completion time ($F_{9,63} = 28.7$, $p < 0.0001$)[3]. Figure 9 illustrates these results: performance comparison among conditions depends on ID_e. First, Tukey post hoc tests showed that *Crossing+Click* is significantly faster than *Pointing* for easy tasks (i.e. $ID_e = 1.65$) and significantly slower than *Pointing* for difficult tasks (i.e. $ID_e = 4.18$). This result is consistent with Accot and Zhai [1]. Second, the difference between *Pointing* and *Edging* is larger for easy tasks than for difficult ones: Tukey post hoc tests showed that *Edging* is significantly faster than *Pointing* for all ID_e values, but the difference between mean completion times is 36.8% for $ID_e = 1.65$ while it is only 6.8% for $ID_e = 4.18$. Focusing our analyses on the *Edging* and *Semi-Infinite* conditions, we still observe a significant effect of *Model* ($F_{1,7} = 19.4$, $p = 0.0031$) and ID_e, but no *Model* $\times$ ID_e interaction effect ($F_{3,21} = 2.6$, $p = 0.0803$) on completion time[4]. Tukey post hoc tests showed that *Semi-Infinite* is significantly faster than *Edging* with a difference in mean of 26 ± 6 ms, this difference being almost similar across ID_e.

In summary, *Edging* and *Semi-Infinite* seem to follow a similar underlying model for ID_e and only differ by a small constant. *Pointing* and *Crossing* seem to follow different models.

Contrary to Experiment 1, we found no significant effect of *Edge* on completion time ($F_{1,7} = 2.9$, $p = 0.1339$), and no significant interaction effect of *Edge* with any other factor on completion time. This difference between the two experiments is probably due to the use of a single symmetric *circle* cursor. This allows us to simplify our analyses by considering *Angle* without distinguishing the *Edge* conditions. We found a significant effect of *Angle* ($F_{6,42} = 13.1$, $p < 0.0001$) and a significant ID_e $\times$ *Angle* interaction effect on completion time. Here again, we observe that movement time depends on movement direction (*Angle*) especially for easy selection tasks whatever the *Model* condition (*Model* $\times$ ID_e $\times$ *Angle* interaction effect was not significant).

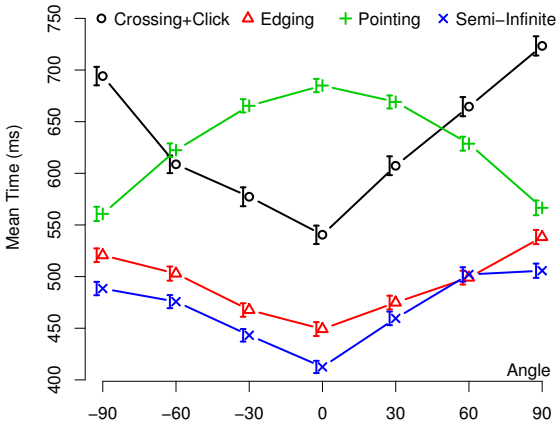


Figure 10: Mean time as a function of *Angle* by *Model* (error bars are shown to the left of each symbol).

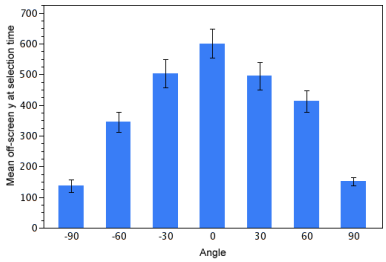


Figure 11: Mean cursor off-screen y-coordinate at target selection time according to *Angle* (*Model* = *Edging*).

Analysis of variance also revealed a significant *Model* $\times$ *Angle* interaction effect ($F_{18,26} = 19.7$, $p < 0.0001$) on completion time as illustrated in Figure 10. First, *Pointing* is faster for horizontal movements than for vertical movements, while *Crossing+Click* is faster for vertical movements than for horizontal movements. These results are consistent with the ones reported in previous work: [10, 20] showed that pointing is faster for horizontal movements than the two other angles they tested and [1] showed that crossing an orthogonal goal is faster than crossing a collinear goal in a continuous movement. Second, differences between *Pointing* and both *Edging* and *Semi-Infinite* are higher for vertical movements (i.e. *Angle* close to zero). For instance, Tukey post hoc tests showed that *Edging* is significantly faster than *Pointing* for *Angle* $= 0$ (a speedup of 34.0%) while there is no significant difference for *Angle* $= \pm 90$. This is probably due to the "virtual" target height in the *Edging* condition that offers a lower *amplitude constraint* for angles close to 0 than for angles close to 90 or -90. The histogram in Figure 11 supports this interpretation: it plots

[3] A finer analysis considering *Width* and *Distance* separately showed that this interaction effect was mainly an effect of *Width*.

[4] And no significant *Model* $\times$ *Width* and *Model* $\times$ *Distance* interaction effects when we consider *Width* and *Distance* separately.

the "virtual" y-coordinate[5] of the cursor at target selection time according to *Angle* in the *Edging* condition and shows that participants stopped their movement further away for angles close to 0 (i.e. vertical movements).

In summary, *Edging* and *Semi-Infinite* seem to follow a similar underlying model for *Angle* while *Pointing* seems to follow a different one. This result also supports our hypothesis regarding the similarity between the underlying models of *Edging* and *Semi-Infinite*.

5. DISCUSSION

The first important finding of this study is that users do take advantage of edges to facilitate target acquisition. Our analyses reveal that acquiring a target on an edge is similar to acquiring a target with a very large height: completion times for both tasks follow a similar function in terms of ID_e (Figure 9) and *Angle* (Figure 10).

The second important finding is that pointing at a target on an edge is quite different from pointing at the same target in the middle of the screen. First, the relationship between movement time and ID_e is different for the two conditions: while in both cases it is an increasing function of ID_e, differences between regular pointing and edge pointing are much larger for low ID_e values than for high ones (Figure 9). Second, the relationship between movement time and movement direction is different: for edge pointing, movement time seems to be a linear *increasing* function of the absolute value of *Angle* while for edge pointing, it seems to be a linear *decreasing* function of the absolute value of *Angle*. This results in performance differences between regular pointing and edge pointing between $+33 \pm 49$ ms (i.e. 4.4% of movement time) and -278 ± 20 ms (i.e. 44.6% of movement time).

As far as we know, the only model that takes movement direction into account is $ID_{W'}$ (eq. 1), which was introduced with ID_{min} (eq. 2) by Mackenzie and Buxton [15]. In their study, $ID_{W'}$ was shown to be less accurate than ID_{min}. Accot and Zhai raised issues with both models and introduced ID_{az} (eq. 3). The table below reports the linear regressions of completion times as a function of ID using each of these models (we consider the 42 mean completion times per condition *Angle* × *Distance* × *Width* for a given *Model*[6]):

Model	$MT = a + b.ID_{W'}$			$MT = a + b.ID_{min}$			$MT = a + b.ID_{az}$		
	a	b	r^2	a	b	r^2	a	b	r^2
Edging	205	127	0.80	80	144	0.90	-37	166	0.92
Semi-Inf.	178	129	0.80	59	143	0.87	-59	166	0.90
Pointing	233	106	0.73	-215	188	0.71	-253	181	0.76

Since ID_{az} contains three free parameters, we tested different combinations for $\omega \in [0,10] \times \eta \in [0,10] \times p \in \{0,1,2\}$ with a step of 0.1 for ω and η. Since the simple values $\omega = \eta = p = 1$, corresponding to $ID = log_2(\frac{D}{W} + \frac{D}{H} + 1)$, did not provide noticeably worse correlation coefficients, we use these values in the table.

Once again, these results support the hypothesis that *Edging* and *Semi-Infinite* follow the same underlying model but differ from traditional *Pointing*. The correlation coefficients however are not as good as for regular pointing, calling for a more detailed analysis.

[5]Even though the cursor is graphically blocked on the edge, we recorded input events directly from the mouse to compute the "virtual" off-screen location at target selection time.

[6]For *Edging* and *Semi-Infinite*, we approximate "infinite" height to 250 pixels, i.e. the height of the area that contains 99% of mouse clicks in the *Semi-Infinite* condition (see Section 4.3).

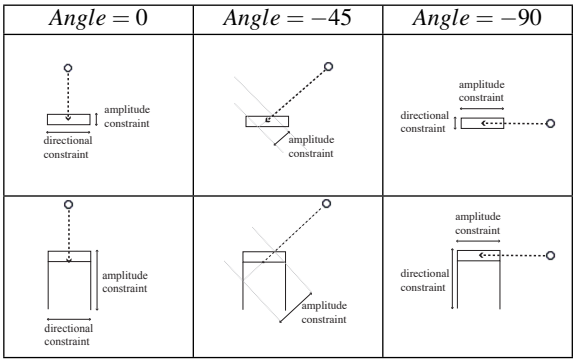


Figure 12: Amplitude and directional constraints for regular pointing (top) and edge pointing (bottom) according to movement direction.

Let us come back to the notions of *amplitude* and *directional* constraints defined by Accot and Zhai [2]. The *Amplitude* constraint is the interval within which the user must stop along the movement direction while the *Directional* constraint is the interval within which the user must stop along the direction orthogonal to the movement. In their study, Accot and Zhai only evaluated non-diagonal movements, so these constraints were simple functions of target width and target height. Figure 12 suggests that taking movement direction into account should help describe the task more accurately. We propose to introduce movement direction in the ID_{az} model based on two ideas mentioned by Accot and Zhai [2]: (i) satisfying an amplitude constraint takes more time than satisfying a directional constraint and (ii) the shortest side must dominate the ID.

To this end, we add a term that emphasizes the contribution of the shortest side to the ID, and we make this term a function of $|Angle|$. Figures 10 and 12 show that the larger the difference between the orientation of the shortest side and movement direction, the smaller the amplitude constraint. In the *Edging* and *Semi-Infinite* conditions (where W is the shortest side), this difference is an *increasing* function of $|Angle|$ while in the *Pointing* condition (where H is the shortest side), this difference is a *decreasing* function of $|Angle|$. We therefore propose the following model where the $|Angle|$ term captures the relationship between orientation of the shortest side and movement direction:

$$ID_{Angle} = \log_2\left(\frac{D}{W} + \frac{D}{H} + f(|Angle|).\frac{D}{min(W,H)} + 1\right)$$

$$f(|Angle|) = 0.6 \times sin(|Angle|) \text{ for } Edging \text{ and } Semi\text{-}Infinite$$

$$f(|Angle|) = 0.6 \times cos(|Angle|) \text{ for } Pointing$$

The table below and Figure 13 show that ID_{Angle} provides much better predictions than the other models studied above:

Model	$MT = a + b.ID_{Angle}$		
	a	b	r^2
Edging	-57	156	0.97
Semi-Inf.	-82	156	0.96
Pointing	-335	191	0.96

To select the functions $f(|Angle|)$, we balanced a trade-off between simplicity and prediction accuracy after systematically considering the following functions: $\{x \times |Angle|,\ x \times sin(|Angle|)\}$ for *Edging* and *Semi-Infinite* and $\{x \times (\frac{\pi}{2} - |Angle|),\ x \times cos(|Angle|)\}$ for *Pointing*, with $x \in [0,10]$ with a precision of 0.05.

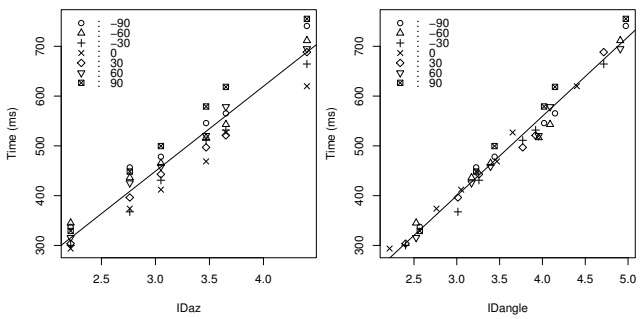


Figure 13: Movement time as a linear function of ID_{az} (left) or as a linear function of ID_{Angle} (right) for *Edging*.

6. CONCLUSION AND FUTURE WORK

We have presented an empirical study to better understand pointing at targets on screen edges. We have also proposed an analysis of the differences between regular pointing and edge pointing and shown that the angle of movement affects the amplitude constraint for a rectangular target. In order to account for these differences with Fitts' law, we have extended Accot and Zhai's definition of index of difficulty (ID) for bivariate pointing. While our model provides better predictions in the study presented here, its validity must be further tested by considering larger sets of target heights and edge orientations, i.e., left and right as well as top and bottom.

Having assessed the effect of various factors on edge pointing performance and compared different pointing models, we can draw the following recommendations to improve current desktop environments. First, the cursor should always be visible even when located on a screen edge. We have shown that a circular cursor shape does improve performance but that target highlighting does not. Other alternatives worth exploring in future work include displaying a small halo around the cursor when it is on the edge [6] or having virtual edges within a few pixels of the physical edges so that the cursor does not move to the physical edge and stays visible. Second, we encourage the use of edges for placing a widget if this does not significantly affect its average distance to the cursor in a typical context of use. The edge creates a "semi-infinite" target that can be acquired up to 44% faster than a regular target at the same distance in the central screen area. Third, we found that movements orthogonal to a given edge, i.e., with a zero angle, afford better performance. Designers should therefore lay out frequently used "edge widgets" close to the center of the edge. Note, however, that we have not tested the special case of corners, which are probably even faster to acquire than edge widgets.

Another research direction for this work is to explore whether "virtual" edges, such as the borders of a window, that would block the cursor under certain conditions, could also improve selection time in specific situations. Obviously, it is important to clearly identify when and how to activate and deactivate such virtual edges so that the user can easily access the rest of the screen. One idea would be to activate the virtual edges while *transient* graphical components, e.g., a pop up menu, are displayed.

7. ACKNOWLEDGEMENTS

We wish to thank Emmanuel Pietriga, our experiment participants and the anonymous reviewers for their feedback. Caroline Appert was supported by a grant from the French Lavoisier program.

8. REFERENCES

[1] J. Accot and S. Zhai. More than dotting the i's — foundations for crossing-based interfaces. In *Proc. CHI '02*, pages 73–80. ACM, 2002.

[2] J. Accot and S. Zhai. Refining Fitts' law models for bivariate pointing. In *Proc. CHI '03*, pages 193–200. ACM, 2003.

[3] M. Akamatsu, I. S. Mackenzie, and T. Hasbroucq. A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics*, 38(4):816–827, 1995.

[4] C. Appert and M. Beaudouin-Lafon. Swingstates: Adding state machines to the swing toolkit. In *Proc. UIST '06*, pages 319–322. ACM, 2006.

[5] AskTog. A quiz designed to give you Fitts, 1999. http://www.asktog.com/columns/022DesignedToGiveFitts.html.

[6] P. Baudisch and R. Rosenholtz. Halo: a technique for visualizing off-screen objects. In *Proc. CHI '03*, pages 481–488. ACM, 2003.

[7] J. Boritz, K. S. Booth, and W. B. Cowan. Fitts's Law Studies of Directional Mouse Movement. In *Proc. GI '91*, pages 216–223. Canadian Hum.-Comp. Comm. Soc., 1991.

[8] S. K. Card, W. K. English, and B. J. Burr. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys, for text selection on a crt. *Human-computer interaction: a multidisciplinary approach*, pages 386–392, 1987.

[9] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *J. Exper. Psych.*, 47:381–391, 1954.

[10] T. Grossman and R. Balakrishnan. A probabilistic approach to modeling two-dimensional pointing. *ACM Trans. Comput.-Hum. Interact.*, 12(3):435–459, 2005.

[11] M. S. Hancock and K. S. Booth. Improving menu placement strategies for pen input. In *Proc. GI '04*, pages 221–230. Canadian Hum.-Comp. Comm. Soc., 2004.

[12] ISO. 9241-9 Ergonomic requirements for office work with visual display terminals (VDTs)-Part 9: Requirements for non-keyboard input devices. *Inter. Org. for Standard.*, 2000.

[13] W. E. Mackay, C. Appert, M. Beaudouin-Lafon, O. Chapuis, Y. Du, J.-D. Fekete, and Y. Guiard. Touchstone: exploratory design of experiments. In *Proc. CHI '07*, pages 1425–1434. ACM, 2007.

[14] I. S. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.*, 7:91–139, 1992.

[15] I. S. MacKenzie and W. Buxton. Extending Fitts' law to two-dimensional tasks. In *Proc. CHI '92*, 219–226. ACM, 1992.

[16] I. S. MacKenzie, A. Sellen, and W. Buxton. A comparison of input devices in element pointing and dragging tasks. In *Proc. CHI '91*, pages 161–166. ACM, 1991.

[17] B. A. Po, B. D. Fisher, and K. S. Booth. Comparing cursor orientations for mouse, pointer, and pen interaction. In *Proc. CHI '05*, pages 291–300. ACM, 2005.

[18] J. Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley, 2000.

[19] N. Smyrnis, I. Evdokimidis, T. Constantinidis, and G. Kastrinakis. Speed-accuracy trade-off in the performance of pointing movements in different directions in two-dimensional space. *Experimental Brain Research*, 134(1):21–31, 2000.

[20] T. G. Whisenand and H. H. Emurian. Some effects of angle of approach on icon selection. In *Proc. CHI '95*, pages 298–299. ACM, 1995.

Surface - Oriented Interaction

Starburst: a Target Expansion Algorithm for Non-Uniform Target Distributions

Patrick Baudisch, Alexander Zotov, Edward Cutrell, and Ken Hinckley
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
{baudisch,alexz,cutrell,kenh}@microsoft.com

ABSTRACT

Acquiring small targets on a tablet or touch screen can be challenging. To address the problem, researchers have proposed techniques that enlarge the effective size of targets by extending targets into adjacent screen space. When applied to targets organized in clusters, however, these techniques show little effect because there is no space to grow into. Unfortunately, target clusters are common in many popular applications. We present Starburst, a space partitioning algorithm that works for target clusters. Starburst identifies areas of available screen space, grows a line from each target into the available space, and then expands that line into a clickable surface. We present the basic algorithm and extensions. We then present 2 user studies in which Starburst led to a reduction in error rate by factors of 9 and 3 compared to traditional target expansion.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

Keywords: target acquisition, target expansion, labeling, Voronoi, mouse, pen, touch input.

1. INTRODUCTION

Acquiring a small target on a computer screen can be challenging, resulting in long targeting times and high error rates. One technique designed to help users acquire small targets is snap-to-target (e.g., [23]), which continuously sets the selection focus to the closest target. Snap-to-target effectively partitions screen space. Figure 1b labels pixels according to which target they snap to; the result is a so-called *Voronoi tessellation* [12]. Users benefit from this target expansion: instead of having to aim for the small target, users click anywhere inside the tile containing the target. This generally reduces targeting time and error rate.

Unfortunately, performance benefits depend on the homogeneity of the target layout. When applied to a target located inside a cluster of targets snap-to-target shows little effect. As illustrated by Figure 1b, targets located inside a cluster are surrounded by little empty screen space. As a result, the tiles generated by the expansion are small—associated targets remain hard to acquire. When used on a device with imprecise input, such as a touch-screen kiosk, the acquisition of such targets will be error prone. The same holds for pen input, as we demonstrate in the two user studies presented in this paper.

In real-world applications locally dense clusters of targets emerge for a variety of reasons. The user interface may represent a real-world geometry with a non-uniform structure, such as cities on a map (Figure 2a). In other cases, it is users who manually create clusters, e.g., when grouping icons on their desktops or when organizing links inside a web page (Figure 2b and c). Or clusters may emerge from the structure of visualized data (Figure 2d).

AVI'08, May 28–30, 2008, Napoli, Italy.

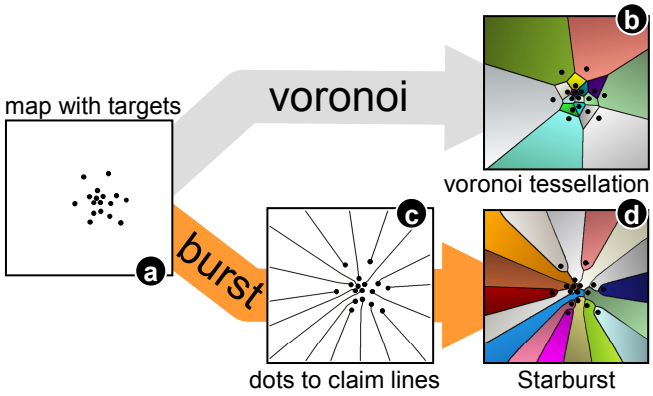


Figure 1: (a→b) Traditional snap-to-target techniques expand targets into *immediately adjacent* space. For targets located inside a cluster, however, that expansion is minimal. (a→c→d) The proposed *Starburst* algorithm connects targets to peripheral screen space to produce reasonably sized tiles for all targets, including those located inside a target cluster.

Limitations in handling target clusters are not unique to snap-to-target, but faced by all techniques based on the repartitioning of screen space, such as Bubble Cursor [14]. Some techniques even impact performance negatively if applied to target clusters. Interactions between closely adjacent *Expanding Targets* cause targets to "escape" from the user [22], resulting in a fisheye navigation problem, as discussed by Gutwin [17].

We propose addressing the problem by expanding targets in a goal-directed way.

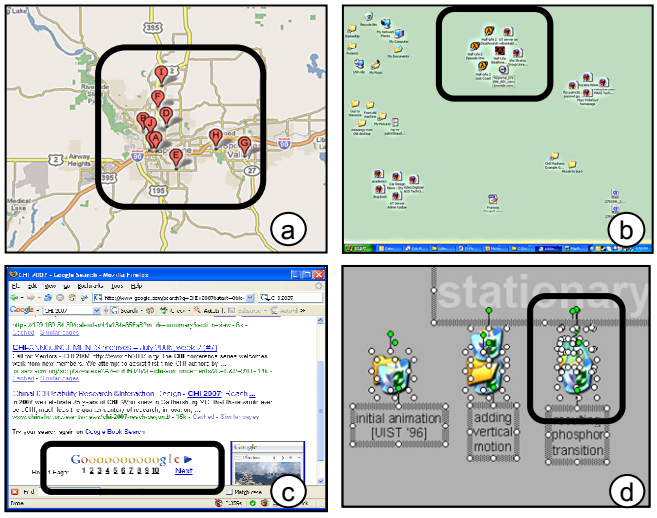


Figure 2: Non-uniform target distributions are commonplace. Examples: (a) yellow-page application showing a map of restaurants, (b) icons on a computer desktop, (c) links in a web page, and (d) handles on geometric objects in PowerPoint™.

2. THE STARBURST ALGORITHM

Figure 1 illustrates the main idea of the proposed *Starburst* technique: While the Voronoi tessellation behind a traditional snap-to-target expands targets directly into target tiles (Figure 1a→b), the proposed Starburst algorithm expands targets first into so-called *claim lines* (Figure 1a→c). Claim lines lead away from the centers of clusters and into empty screen space. Then claim lines expand into clickable surfaces (Figure 1d). The resulting layout is characterized by lines escaping from the cluster center, which gave the technique its name.

By providing targets located inside a cluster with access to empty screen space, the Starburst algorithm is able to assign screen space to targets that remain small if expanded using the traditional Voronoi approach. If used on a device with limited input accuracy, such as a pen-based tablet or a touch screen-based kiosk system, this can lead to substantial performance improvements. In our user studies, expanding targets using Starburst led to a reduction in error rate by a factor of up to 9 compared to target expansion using traditional Voronoi tessellation. The proposed algorithm thereby makes the concept of target expansion applicable to scenarios that have not been accessible to these techniques so far.

2.1. Walkthrough of the algorithm

Figure 3 shows how the Starburst algorithm converts a given target layout (Figure 3a) into a Starburst tile layout (Figure 3i).

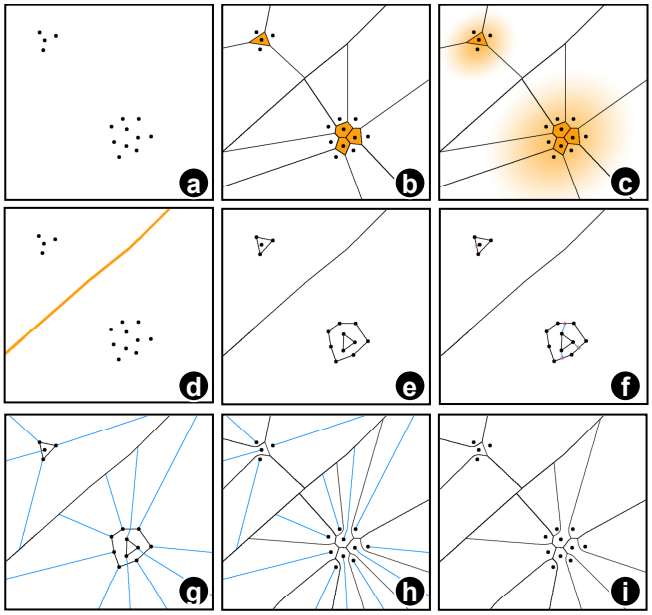


Figure 3: A walkthrough of the Starburst algorithm (a) Targets to be expanded, (b) Voronoi tessellation and identification of *recipients*, (c-d) clustering of targets into cliques, (e) nested rings, (f-g) claim line construction, (h) expansion of claim lines into tiles, and (i) final removal of claim lines.

1. Identifying targets that require additional expansion. The Starburst algorithm begins by performing a Voronoi tessellation [12] on the targets (Figure 3b). The algorithm then identifies small tiles in that Voronoi layout. Tiles which have surfaces that fall below the average tile size by a threshold (we used a factor of 5) are tagged as tiles in need of expansion. In Figure 3b these *recipients* are highlighted in orange. All other targets are tagged *donors*.

2. Organizing targets into cliques. Starburst manages the redistribution of screen space based on what we call *cliques*. A clique is a set of collocated donors and recipients. Within a clique, donors provide the screen space used to expand recipients. The Starburst algorithm computes cliques in three steps. First, it creates cliques by clustering recipients based on adjacency. In Figure 3c this results in a clique with three recipients and a clique with a single recipient. Second, the algorithm adds all donors immediately adjacent to a clique of recipients to that clique. In case a donor is adjacent to multiple cliques the donor is added to the clique with the smallest average tile size. In the case of Figure 3c this adds three donors to the single-recipient clique in the top left, all others to the three-recipient clique. Third, the Starburst algorithm adds additional donors if they are particularly large or if they are located in an area in which the clique lacks good donors. In order to be included, a candidate must be adjacent to a clique and its surface must significantly exceed the average tile size *in that clique* (we used a threshold factor of 5). In Figure 3b, all donors were already added in the previous step, so no further addition takes place. Once cliques have been formed, the Voronoi tessellation and the recipient/donor labeling is dropped (Figure 3d).

The goal of the next steps is to provide targets located on the inside of a clique with access to screen space in the periphery of the clique. In order to reach the periphery, claim lines of inner targets need to pass between outer targets and so passages between targets become potential bottlenecks. We therefore create a representation that reflects these potential bottlenecks.

3. Organizing targets into nested rings: Starburst organizes the targets of each clique into a set of nested rings (Figure 3e). The algorithm starts by computing the convex hull over all targets of a clique. All targets located on that convex hull form the *outer ring*. Then Starburst computes the second ring by computing a convex hull over the remaining targets, and so on.

4. Routing claim lines. Next the algorithm creates the claim lines. The algorithm starts with the innermost ring and connects all its targets to the immediately enclosing ring (Figure 3f). Each claim line is connected to the *nearest* edge of the outer ring that can be reached with a straight line without intersecting the inner ring. This guarantees that claim lines never intersect. If multiple claim lines are connected to the same edge, the algorithm spaces them out equidistantly; single claim lines are connected in the middle of the ring edge. This helps balance the width of the tiles at the point where they pass between the targets. Then the algorithm repeats this step, i.e., all targets on the next ring plus the newly added targets are routed to the ring another layer out. In Figure 3e, the deepest clique has two nested rings, so a single iteration is sufficient for connecting all targets to the outer ring. Now the algorithm spreads the claim lines radially into the clique's peripheral screen space (Figure 3g).

5. Growing claim lines into tiles. In the last step, Starburst creates the target tiles. The algorithm does this by assigning all pixels on screen to the target with the closest claim line as shown in Figure 3h. This completes the processing and Figure 3i shows the final result without the claim lines.

2.2. Algorithms, complexity, and performance

The overall complexity of the Starburst algorithm is $O(n^2)$ with n being the number of targets, which allows for real-time performance with dozens of targets or several hundred targets if only a subset of them moves.

Details on the computational complexity: Step 1: We compute the initial Voronoi tessellation and its Delaunay triangulation [20] using a modified Fortune algorithm in $O(n \log n)$ time [12]. We compute the size of the Voronoi tiles in $O(h)$, where h is number of edges, by reusing the quad edge data structure from the Fortune algorithm. Step 2: We compare the size of each tile with its $O(h)$ neighbors in $O(n^2)$ worst case time ($O(n \log n)$ average time).

Step 3: Constructing of the nested rings, known as *onion-peeling*, can be performed in O(n log n) time [25], but since we already computed the Delaunay triangulation we perform onion-peeling in O(n) time. Step 4: In the worst case, onion-peeling generates O(n) rings, in which case routing n claim lines through n rings requires O(n^2) time. Step 5: We perform another Fortune Voronoi tessellation, this time on the claim line segments, resulting in straight line segments and parabolic segments in O(n log n) time.

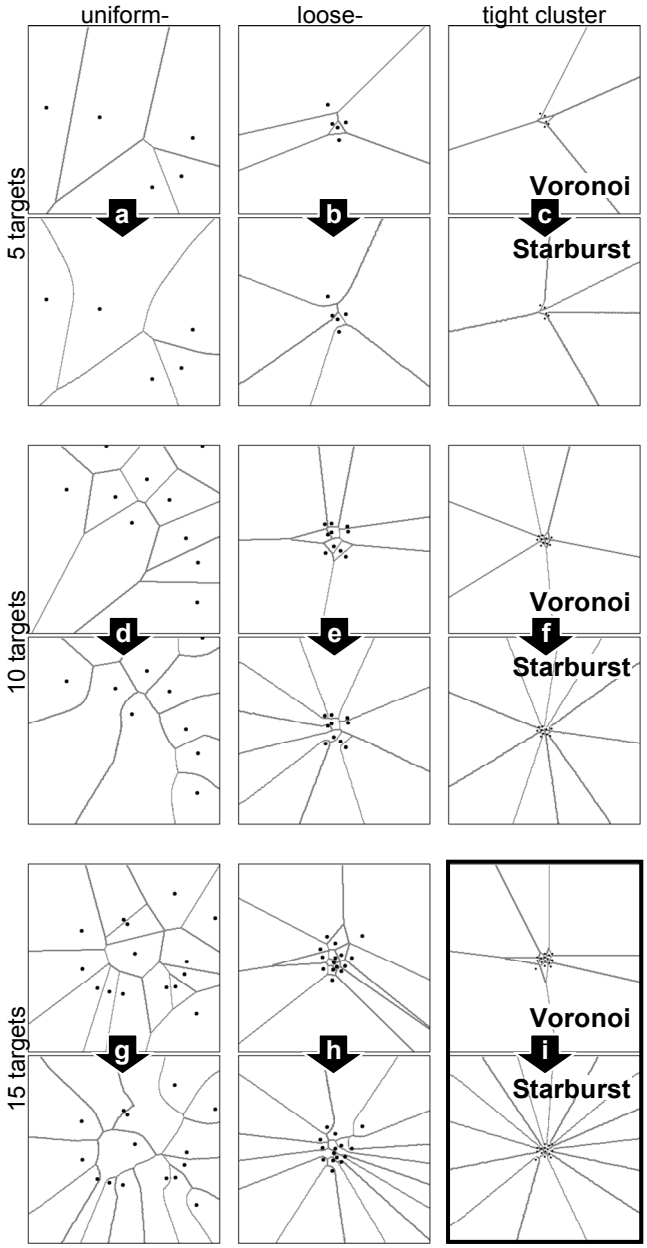


Figure 4: Examples of Voronoi tessellations (top of each pair) and the corresponding Starburst tessellation (bottom of each pair) for target layouts with 5, 10, or 15 targets and clusters of different tightness (layouts used in the user study).

2.3. Sample layouts

Figure 4 and Figure 5 show sample layouts generated using the Starburst algorithm described above and contrasts them with the corresponding Voronoi layouts.

Figure 4 shows Starburst layouts for nine single-cluster target layouts, a subset of the layouts we evaluated experimentally in our user studies. The left column of Figure 4 shows tile layouts resulting from uniform target distributions. The Voronoi-based approach was designed for uniform target distributions [14] and works as expected. Tiles in the Starburst layouts are rounder, but overall of similar quality as the Voronoi tiles. The layouts in the center column, in contrast, contain clusters. The clusters cause the Voronoi layouts to degrade visibly and inside-the-cluster targets are assigned very small tiles. The Starburst layouts, in contrast, continue to offer reasonably-sized tiles for all targets. For target layouts containing tighter clusters this effect intensifies. The vertical axis in this figure reflects the number of targets in each layout. As we move down in the diagram the target count increases. As a result, the number of inaccessible targets in the Voronoi condition increases as well. The Starburst layouts, in contrast, remain functional. Figure 5 shows a selection of multi-cluster layouts. We see similar effects as in the single cluster examples.

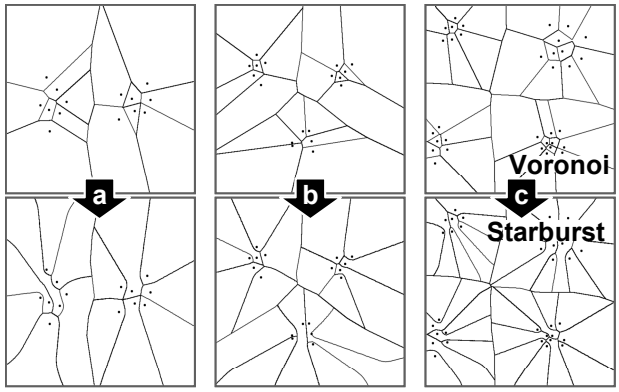


Figure 5: Examples of Voronoi (top) and Starburst (bottom) tessellations for layouts with (a) 2, (b) 3, and (c) 4 clusters.

Layouts generated by the Starburst algorithm are quite robust, i.e., insertion, removal, or relocation of targets impacts the tile layout only locally. This helps users build up spatial memory when using a Starburst layout over time.

2.4. User interface for Starburst

To outlines of Starburst tiles are irregular and therefore generally not "guessable". A user interface deploying Starburst therefore needs to convey tile shapes to the user.

On devices supporting a hover state, such as table computers, target expansion using Starburst can be presented to the user interactively—on hover as shown in Figure 6a-c. (a) By default, only screen content is visible. (b) As the pointer moves across the screen, targets within an n-pixel radius around the pointer get increasingly "excited" and the respective tile overlays turn opaque. The tile under the pointer is highlighted. (c) Tiles away from the pointer fade to transparent, yet stay opaque long enough to allow users to tap.

Some devices, such as resistive touch screens or table top systems, do not support a tracking state. On these systems, tile boundaries are overlaid permanently onto screen content, rather than revealing them on hover. Tiles outlines can interfere with line-shaped document features as shown in Figure 6d. Such interference can often be reduced by encoding tile outlines using features not contained in the underlying document (see *multiblending* [2]).

Note that screen devices without a tracking state support *none* of the interactive expansion techniques mentioned earlier, such as Expanding Targets. Also Bubble Cursor is not applicable to such display systems; removal of the bubble visuals would reduce bubble cursor to the underlying space partitioning algorithm, i.e., a Voronoi tessellation.

2.5. Limitations

Similar to other target expansion techniques, Starburst helps overcome limitations in the clickable size of targets. A potential limitation of Starburst is that it tends to generate long and narrow targets, a type of shape that can be more difficult to acquire than rounder, wider targets [15]. In the section "Improving space allocation", we describe extensions to the algorithm that result in wider target shapes.

What remains are limitations based on the visual size of target layouts. Larger and tighter clusters result in thinner pathways at the point where claim lines pass the rings. When these pathways get so thin that they are hard to visually trace or when their thickness reaches screen resolution, some targets cannot be expanded anymore and the Starburst algorithm has reached its limit. Fortunately, as our user studies indicate, this point is reached much later than the motor space limits faced by Voronoi-based approaches.

Figure 6: (a-c) on-hover exploration and (d) permanent overlay of Starburst tessellation using an emboss effect

3. RELATED WORK

Starburst is related to target acquisition and labeling.

3.1. Targeting and target expansion

In order to help users acquire small targets, researchers have proposed expanding targets in various ways.

Expansion of targets in motor space: researchers have proposed slowing down the pointer motion on and around small targets (e.g., *sticky icons* [30], also suggested by [29], *semantic pointing* [9]). Such adjustments of *control display ratio* (or *cd ratio*) increase the target's size in motor space. *Object pointing* [16] suggests removing space between targets altogether, letting users jump between targets.

Approaches based on cd ratio adjustment require users to cross the target for the cd ration enhancement to become active [3]. Researchers have therefore proposed *magnetism* [5] and *gravity* [8]. *Snap-and-go* [3] uses invisible guides that direct the user's motion while the actual propulsion still comes from the user.

On touch and pen-based systems, motor space enhancements are typically applied by using take-off selection [24]. The 1:1 mapping of these screen devices is used only to determining the initial contact position; then users iterate under a local cd ratio adjustment and commit by lifting their pen or finger off the screen (*high-precision touch screen* [26]). Benko et al. allow users to control cd-ratio manually using a second finger or the non-dominant hand [7].

Expansion of targets in visual and motor space: Some researchers have proposed manual expansion of targets using an intermitted zoom step [1, 26]. In order to apply target expansion to touch and pen-based systems with *land-on* selection [24], the motor space size of targets needs to be increased permanently. *Expanding targets,* proposed by McGuffin and Balakrishnan, refers to an expansion of the target in visual *and* motor space as the pointer approaches it [22]. For an isolated target, the motor space of the target is determined by the expanded space and McGuffin et al. found that the targeting performance is largely determined by the size of that expanded state [21].

For clusters of adjacent targets, however, target expansion in visual space causes targets to push each other away [21]. Although the visuals of each target expand fully, the proximity of the adjacent targets affects a target's ability to expand in motor space. In tightly packed clusters, *no* motor space expansion can take place.

Expansion of cursor vs. expansion of targets: To prevent these problems, researchers have looked at ways to expand targets without pushing other targets away. Bubble cursor is one such solution [14]. It shows an on-hover bubble around the pointer that varies in size, such that it contains the closest target. Bubble cursor has been applied to a variety of target acquisition techniques, such as the tractor beam [23]. There are three different ways of looking at bubble cursor. When focusing on its effect on motor space, bubble cursor divides screen space up resulting in a Voronoi diagram. The second way of looking at bubble cursor is to consider it a snap-to-target mechanism. And third, it can be considered an *area cursor* (*sticky icons* [30], also *prince technique* [18]) of adaptive size. With respect to the underlying motor space properties all three viewpoints are equivalent, although each perspective inspires a different visual user interface.

3.2. Target acquisition as a labeling problem

Another approach to associating small targets with larger motor space areas is to create a layer of *handles*—one handle for each target—that is overlaid onto the actual document content. Many programs, such as MS PowerPoint™ and Adobe Illustrator™ use little white circles to represent corners of graphical primitives that would otherwise measure only a single pixel (Figure 7a). In case a primitive is too small to fit all handles (Figure 7b, c), PowerPoint drops some of them, and finally (Figure 7d) it decouples the handles from the actual object in order to prevent handles from overlapping. Despite the decoupling, the association between handle and target is clear because of their proximity. In the case of multiple objects (Figure 7e), handles do overlap and once more it is difficult to acquire them.

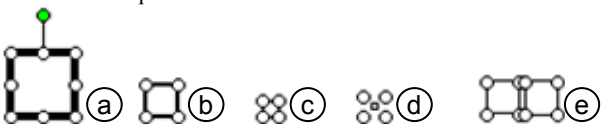


Figure 7: Resize handles in MS PowerPoint™

The idea of decoupling handles from the target can be pushed further. While we are not aware of any such research specifically designed to help users acquire small targets, a lot of research has been done on *labeling* screen objects (e.g., [19]). *Excentric labels* [11] assign labels to an entire cluster of small objects by using an explosion-drawing-like display (Figure 8). To avoid overlap between labels, they are placed at a distance from the actual targets. To associate labels and targets, this approach relies on lines and in some cases also color. While the purpose of the labels is to hold a piece of text or an icon explaining the referenced object, one could imagine using external labels for the purpose of making the associated object clickable.

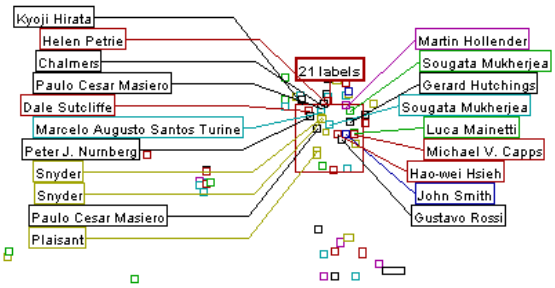


Figure 8: Excentric labels [11]

A potential limitation of this approach is that the lines produce clutter. This can make it hard for users to locate a label belonging to a particular target. Bell et al. propose an algorithm that minimizes connecting lines by placing labels onto the actual object whenever the size and shape of the target permit it [6]. The use of such *internal* labels can reduce visual search as targets and label are associated by proximity, while users need to trace a line in order to locate an *external* label.

Following this analogy, layouts produced by the Voronoi algorithm consist exclusively of internal "labels", at the expense of offering no control over their size. The Starburst algorithm, in contrast, keeps internal "labels" only if they are large enough. Otherwise it expands into an external "label". Unlike external and excentric labels, however, Starburst creates lines, tiles, and targets in the same plane, so labels never occlude targets. In that sense, Starburst shares some properties with circuit board routing [10].

4. DESIGN DISCUSSION

In this section, we give a brief overview of the design alternatives we explored and discuss their strengths and limitations. Our first two approaches were based on refining Voronoi tessellations.

4.1. Refining Voronoi by moving boundaries

Figure 9 shows a Voronoi layout and a modification obtained by moving and rotating a tile boundary. While this approach allowed for certain layout improvements, the use of straight tile boundaries turned out to be a major limitation, because many target layouts require non-straight boundaries (see, for example, the center areas of the layouts generated by Starburst in Figure 4).

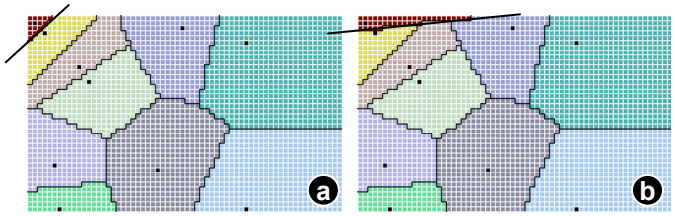


Figure 9: Boundary adjustment approach: (a) Voronoi Tessellation; (b) expansion of the tile in the top left corner by moving and rotating its boundary.

4.2. Refining Voronoi by reassigning pixels

To address this limitation, we explored algorithms that represented screen space as pixels, rather than tile boundaries. Cellular automata and pixel rewriting allow creation of a rich spectrum of shapes [13]. The high degree of flexibility, however, made it difficult to control tile growth and to direct target growth towards available space. We often obtained inefficient shapes (Figure 10c) and improving one tile often came at the expense of making another tile significantly worse (Figure 10d).

4.3. Claim lines

Based on these insights, we started looking for an algorithm that would offer flexibility *and* control. Claim lines provide tiles with a much-needed skeleton—a concept well understood in computer graphics [28]. That skeleton allowed us to direct target growth towards available space and prevent uncontrolled expansion. Yet, the resulting target tiles were not limited to straight edged or convex shapes.

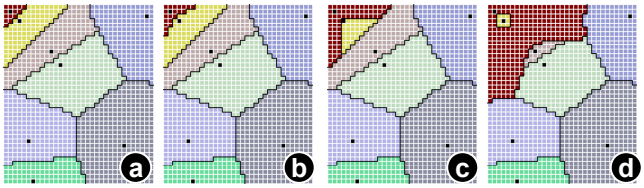


Figure 10: Pixel rewriting approach: (a) Voronoi tessellation; (b) expansion of the top left target using pixel rewriting; (c, d) further expansion leading to undesirable target shapes.

We went through several design iterations to determine a claim line skeleton that would offer enough flexibility to avoid bottlenecks yet be simple enough to allow for good control.

Our first attempt used single-segment claim lines, which it created by drawing a straight line from a common "center point" located inside the cluster through the individual targets. This approach turned out to be too limited and long strips of targets resulted in inefficient space usage.

To address these shortcomings, we switched to multi-segment lines. We tried to avoid bottlenecks by making claim lines repel each other, yet that made it difficult to direct claim lines towards available screen space.

Our final version, the nested ring approach, finally, reduced the number of line segments to what was absolutely necessary and offered a good handle on bottlenecks. This resulted in cleaner layouts, faster computation, and the desired degree of control.

5. IMPLEMENTATION

Figure 11 shows our Starburst test environment. It allows placing targets and generating tile layouts using a variety of algorithms. It was implemented using the .NET WinForms framework and runs on Microsoft Windows XP Tablet PC Edition.

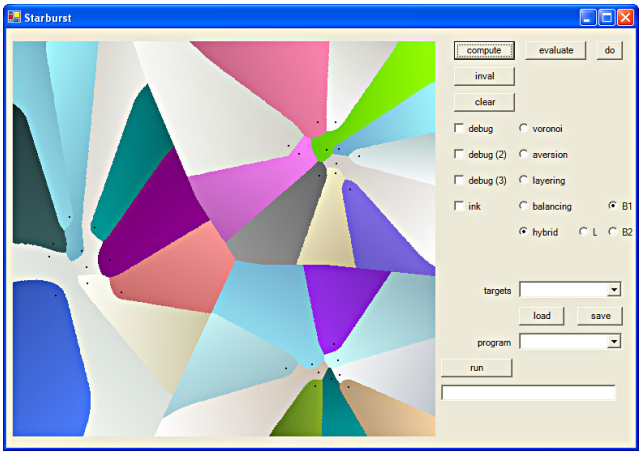


Figure 11: The Starburst test environment for Tablet PC

6. USER STUDIES

To objectively evaluate the performance of the Starburst algorithm, we conducted two controlled experiments comparing Starburst with traditional Voronoi target expansion.

The goal of the first experiment was to verify that our technique indeed reduces the motor skills required to select clustered targets. Voronoi and Starburst both make use of the entire screen space—the average size of generated tiles is therefore the same. Starburst

does not *increase* tile sizes compared to Voronoi, but *balances* tile sizes; its median target size is higher that Voronoi's, not its mean. On the flipside, as discussed earlier, targets generated by Starburst tend to be longer and thinner. We were wondering how the two effects would play out against each other. The first study investigated this by highlighting the entire target tile (Figure 12a).

After finding a very strong effect in the first study (a reduction of error by a factor of nine) we conducted a second study. This time we looked at a more realistic scenario simulating a user encountering a target layout for the first time or who works with a layout undergoing perpetual change. How effectively would users acquire targets now? We implemented this scenario by highlighting the target only, not the tile, so that users had to visually examine layouts for every trial to determine where to tap (Figure 12b).

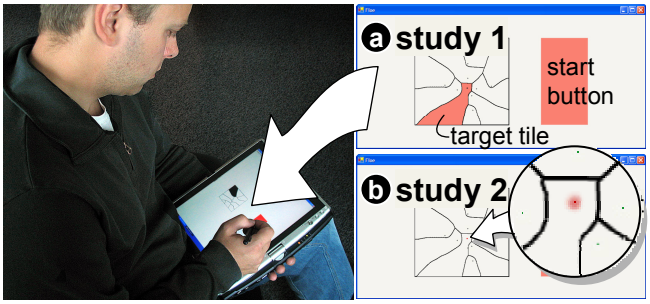


Figure 12: Participants tapped the start button and then the tile associated with the target. (a) In study 1, the entire target tile was highlighted, (b) in study 2 only the target itself.

7. USER STUDY 1

The participants' task in the first study was to acquire targets with a pen on a tablet computer (Figure 12). Target acquisition was supported by expanding all targets in the target layout into a space-filling layout of tiles. Participants could acquire a target by acquiring any part of the associated tile. As mentioned above, the entire tile associated with the target was shaded red (Figure 12a).

Our main hypothesis was that participants would acquire with less errors if layouts were generated using Starburst.

7.1. Interfaces

There were two interface conditions. In the *Starburst* condition, target tile layouts were generated using the algorithm described at the beginning of this paper. In the *Voronoi* condition, target tile layouts were generated using the traditional Voronoi approach.

Both interfaces provided permanently visible tile boundaries, i.e., a set of black lines as shown in Figure 12. We chose this interface style, because it is available on all devices—unlike interface styles relying on hover.

7.2. Target layouts

Target layouts measured 256 x 256 pixels and 2" x 2" (5 x 5 cm) on screen. To keep the number of trials manageable and since multi-cluster layouts are structurally similar (Figure 5) we used uniform and single-cluster layouts only. Figure 4 show examples for each of the nine types of target layouts used in the study: each target layout contained 5, 10, or 15 targets; targets were organized either in a uniform distribution (*uniform*), in a normal distribution with standard deviation of 32 pixels (*loose*), or in a normal distribution with standard deviation of 8 pixels (*tight*). For each of the nine layout types we randomly generated 5 target layouts. Each participant completed each layout using each of two interfaces. This resulted in 3 target counts x 3 densities x 5 layouts x 2 interfaces = 90 layouts.

7.3. Task

The participants' task was to acquire targets using the pen. Each trial proceeded as follows. (1) The current target was highlighted in gray and the start button turned red as shown in Figure 12. (2) Participants tapped the start button (100 x 256 pixels, 0.8" x 2"/2cm x 5cm) located right of the target layout. This was acknowledged with a "click" sound and started the timer for that trial. (3) Participants acquired the highlighted target by tapping anywhere within its tile using the pen. This stopped the timer. Success/failure was confirmed using auditory feedback.

While participants acquired one target per trial, performance was measured on a per-layout basis. A per-target comparison did not make sense, because target sizes and shapes of the tiles in a layout were not independent from each other; adding space to one target to make it easier to acquire came at the expense of making another one smaller and thus harder to acquire.

This meant that participants needed to perform 10 times more target acquisitions for the same number of data points than in a normal target acquisition study. In order to keep the number of repetitions manageable, distance and angle of the target were *not* varied in this experiment. Instead we used the aforementioned start button located at a fixed position. While the start button placement could impact targeting times of individual targets, its effect balanced out across entire layouts.

7.4. Procedure

Each participant acquired every target of the 90 tile layouts once, i.e., there were 45 target layouts, each one tessellated differently for each of the two interface conditions. Each participant therefore performed a total of 3 levels of target counts (5, 10, or 15 targets) * 3 densities (uniform, loose, tight) * 5 layouts * 2 interfaces = 900 trials. To minimize learning and ordering effects, the order of all 900 trials was randomized, so that in the general case the entire target layout changed from trial to trial. Overall, the user study took about 20 minutes per participant.

7.5. Apparatus

Participants performed all tasks using a Toshiba Portégé M200 Tablet PC, with a 12.1" inch LCD monitor running the Microsoft Windows XP Tablet PC Edition operating system. The screen measured 7½" x 9¾" (19cm x 25cm), offered 1400 x 1050 pixel resolution (140dpi), and was used in portrait orientation. Participants performed all interaction using a pen. The tablet keyboard was hidden ("slate mode"). The tablet was placed on a table, but participants were allowed to hold the tablet in the lap instead, if they preferred (Figure 12). The experimental application was implemented using the .NET WinForms framework.

7.6. Participants

12 volunteers (10 male) between the ages of 20 and 40 were recruited from our institution. Each one received a lunch coupon for our cafeteria as a gratuity for their time. All had experience with graphical user interfaces, TabletPC, and pen input. Nine participants were right handed. All had normal or corrected to normal vision and normal color vision.

7.7. Hypotheses

We had the following three hypotheses:

(H1) Participants would acquire target layouts faster and with fewer errors for the clustered target layouts (*loose* and *tight* conditions) when using the Starburst interface.

(H2) The performance benefit of the Starburst condition would increase with the number of targets in a layout. The reason is that a higher target count would cause more targets to be enclosed inside clusters in the Voronoi condition.

(H3) The performance benefit of the Starburst condition would be greater in the *tight* condition. In the Voronoi condition, the tighter packing would make tiles of targets located inside a cluster even smaller.

We did not expect any performance benefits for the Starburst interface in the uniform layout conditions because neither of the techniques should produce any small targets.

7.8. Results

Performance was measured in error rates and targeting times for each condition.

7.8.1. Error rates

We aggregated selection errors across all 5 layouts per condition to compute an error metric for each condition. We then performed a 3 (*TargetCount*) × 3 (*Density*) × 2 (*Technique*) within subjects analysis of variance. We found significant main effects for all three variables. For *TargetCount* ($F(2,22)=92.5$, $p<<0.001$), accuracy decreased as the number of targets increased. Similarly for *Density* ($F(2,22)=158.4$, $p<<0.001$), as the density increased, so did the error rate. Finally, for *Technique* ($F(1,11)=272.1$, $p<<0.001$), Voronoi was associated with significantly higher error rates than Starburst (14% vs. 2% error).

In addition, all interactions tested were significant: *TargetCount x Density*, $F(4,44)=24.1$, $p<<0.001$; *TargetCount x Technique*, $F(2,22)=51.2$, $p<<0.001$; *Density x Technique*, $F(2,22)=204.8$, $p<<0.001$; and *TargetCount x Density x Technique*, $F(4,44)=12.9$, $p<<0.001$. Figure 13 illustrates all the error rates for each technique and all display conditions. Post hoc paired t-tests were performed comparing each technique at each condition and significant differences are denoted by "*" (Bonferroni adjustment for multiple tests, $p<0.005$).

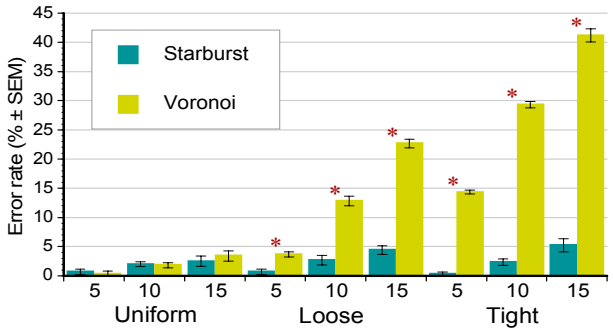


Figure 13: Error rates over layout types (+/- std error of mean)

7.8.2. Target acquisition times

Before analyzing target acquisition times, outliers were removed from the analysis based on a heuristic of any acquisition longer than 2 seconds (this is well over 3 standard deviations from the mean for a given condition). A total of 55 out of 10745 trials were removed from the data (45 from the Voronoi conditions).

As with error rates, for time analyses we collapsed target acquisition times across all 5 layouts per condition, computing the median target acquisition time for each condition. We performed a 3 (*TargetCount*) × 3 (*Density*) × 2 (*Technique*) within subjects analysis of variance for acquisition time. We found significant main effects for all three variables. For *TargetCount* ($F(2,22)=244.4$, $p<<0.001$), acquisition time increased as the number of targets increased. Similarly for *Density* ($F(2,22)=76.3$, $p<<0.001$), as the density increased, so did target acquisition time. Finally, for *Technique* ($F(1,11)=65.9$, $p<<0.001$), Starburst was significantly faster than Voronoi.

In addition, all interactions tested were significant: *TargetCount x Density*, $F(4,44)=20.7$, $p<<0.001$; *TargetCount x Technique*, $F(2,22)=11.0$, $p<0.01$; *Density x Technique*, $F(2,22)=47.5$, $p<<0.001$; and *TargetCount x Density x Technique*, $F(4,44)=7.8$, $p<0.01$. Figure 14 illustrates targeting times for each technique and all display conditions. Post hoc paired t-tests were performed comparing each technique at each condition and significant differences are denoted by "*" (Bonferroni adjustment for multiple tests, $p<0.005$).

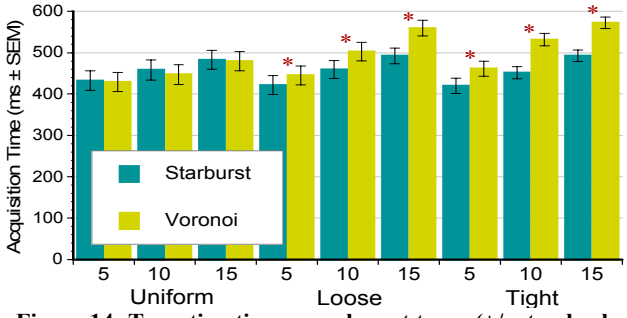


Figure 14: Targeting times over layout types (+/- standard error of the mean).

7.9. Discussion

In summary, the study results support all three hypotheses. Participants acquired tiles layouts generated using Starburst faster and with a substantially lower error rate than tiles generated by the Voronoi conditions. This supports our hypothesis that the improved balancing of target sizes outweighs the drawback resulting from the degeneration of tile shapes. Tighter clusters and more targets increased the gap in performance.

8. USER STUDY 2

As mentioned earlier, the purpose of the second study was to investigate the more realistic scenario where users encounter a target layout for the first time. The second study was identical to the first, except:

Interfaces: only the target itself was highlighted, but not the corresponding tile, so that users had to visually examine the layout to determine where to click. Since targets were very small, they were also provided with a pale red glow to make them easier to locate, as shown in Figure 12b. As before, targets were revealed upon completion of the previous trial. All participants tapped start in immediate succession to completing a trial and did not inspect layouts before tapping start.

Additional density condition: We only tested the 5 and the 10 target conditions, but not the 15 target conditions (Figure 5). Participants therefore now performed 2 target counts x 3 densities x 5 layouts x 2 interfaces = 60 layouts.

Participants: 6 participants (5 male); all with GUI experience; 2 Tablet PC users and pen input experience; 5 right handed and one left handed. All had normal or corrected to normal vision and normal color vision.

Hypotheses: As in the first study, we expected to see a benefit in error rate. Since the visual analysis of the Starburst layout would take time, we did not expect to see a benefit in task time though.

8.1. Results

Performance was measured in error rates and targeting times for each condition.

8.1.1. Error rates

Analyses for Study 2 were nearly identical to Study 1. While the accuracy rates tended to be slightly lower (reflecting the increased task difficulty), the pattern was the same. We performed a 2 (*TargetCount*) × 3 (*Density*) × 2 (*Technique*) within subjects analysis of variance. We found significant main effects for all three variables. For *TargetCount* ($F(1,5)=42.9$, $p<0.001$), accuracy de-

creased as the number of targets increased. Similarly for *Density* ($F(2,10)=97.9$, $p<<0.001$), as the density increased, so did the error rate. Finally, for *Technique* ($F(1,5)=37.6$, $p<0.002$), Voronoi was associated with significantly higher error rates than Starburst (10% vs. 4% error).

Unlike study 1, only 2 interactions were significant: *TargetCount x Density*, $F(2,10)=17.8$, $p<0.001$; and *Density x Technique*, $F(2,10)=39.6$, $p<<0.001$. Figure 15 illustrates all the hit rates for each technique and all display conditions. Post hoc paired t-tests were performed comparing each technique at each condition and significant differences are denoted by "*" (Bonferroni adjustment for multiple tests, $p<0.008$).

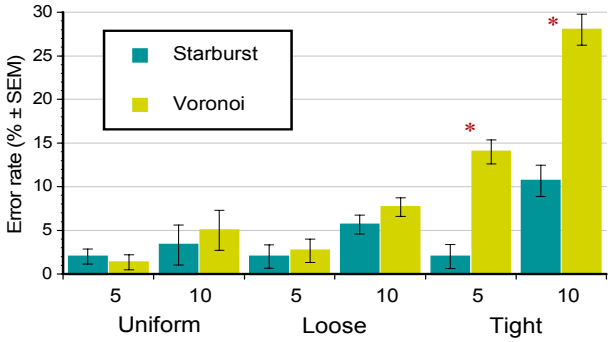


Figure 15: Error rates over layout types (+/- std error of mean)

8.1.2. Target acquisition times

As expected, the time for target acquisition was generally longer than in study 1, reflecting the greater difficulty of the task. We performed a 2 (*TargetCount*) × 3 (*Density*) × 2 (*Technique*) within subjects analysis of variance for acquisition time. We found significant main effects for all three variables. For *TargetCount* ($F(1,5)=18.3$, $p<0.001$), acquisition time increased as the number of targets increased. Similarly for *Density* ($F(2,10)=6.49$, $p<0.02$), as the density increased, so did target acquisition time. Finally, for *Technique* ($F(1,5)=10.7$, $p<0.02$), Starburst was significantly faster than Voronoi.

No interactions were significant. Figure 16 illustrates targeting times for each technique and all display conditions. As above, post hoc paired t-tests were performed comparing each technique at each condition and significant differences are denoted by "*" (Bonferroni adjustment for multiple tests, $p<0.008$).

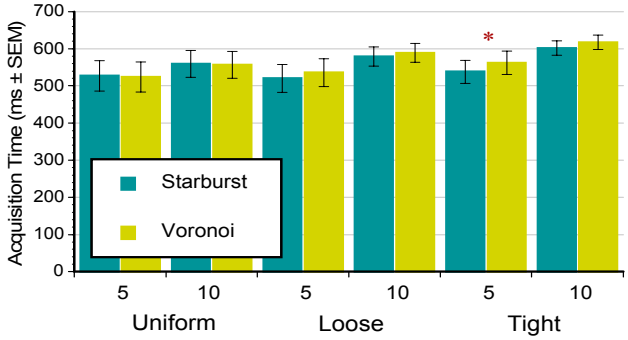


Figure 16: Targeting times over layout types (+/- standard error of the mean)

8.2. Discussion

Also the second study results support our hypotheses. While the visual analysis of the Starburst layout resulted in longer task times and higher error rates in both interface conditions compared to the first study, the Starburst layout still outperformed the Voronoi layout on both measures.

9. IMPROVING SPACE ALLOCATION

The Starburst algorithm, as described throughout this paper, improves target tile layouts by reallocating screen space from donors to recipients. While the algorithm delivers good results for the average case, it can lead to suboptimal results if the supply of screen space is distributed unequally around a cluster. In the example shown in Figure 17a, for example, the five claim lines in the bottom left access only limited amounts of screen space. In the following, we present an extension of our algorithm that causes it to take the availability of screen space into account. The extension replaces step 4 of the original algorithm as follows.

4a. Locate available screen space. To probe space availability this algorithm casts rays from the outer ring into the periphery, intersects them with the clique boundaries (dashed and dotted lines in Figure 17b), and measures the length of the ray. Sectors that are too "shallow" are excluded from the following space allocation steps (finely dotted lines in Figure 17b).

4b. Place claim line endpoints. The algorithm places claim line endpoints into the sectors marked as available. For a reasonably small number of targets per clique, such as 20, the algorithm partitions screen space radially as shown in Figure 17c.

4c. Route claim lines between targets and endpoints. The algorithm descends claim lines from the endpoints to the closest segment of the outer ring. Then it flips pairs of connections until claim lines do not intersect each other anymore. It repeats this step for all remaining ring layers.

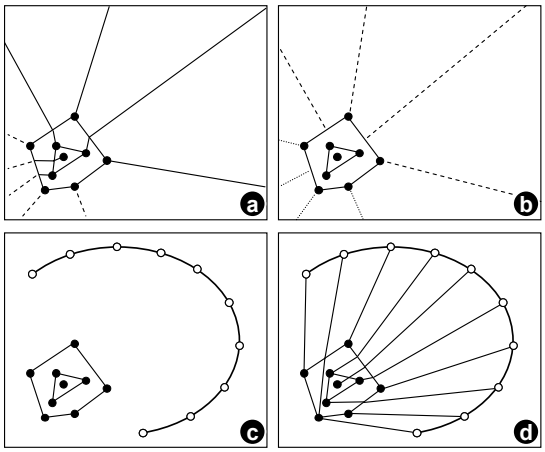


Figure 17: (a) The 5 dashed claim lines have limited access to screen space. The extension (b) locates available screen space, (c) places claim line endpoints into the available screen space, and then (d) routes claim lines from endpoints to targets.

Figure 18 juxtaposes a tile layout generated using the basic Starburst algorithm with the corresponding layout produced by the extended version.

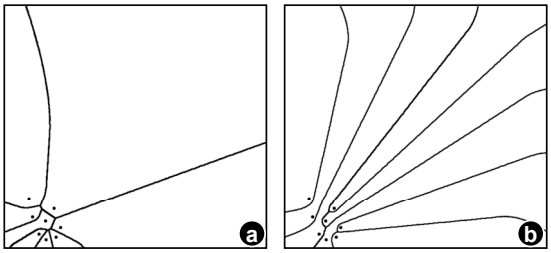


Figure 18: (a) A tile layout generated using the basic Starburst method and (b) using the extended version

For clusters with more than 20 targets, spreading claim line endpoints along a single arc produces very thin tiles that can be hard to acquire [15]. To avoid this, our algorithm handles large numbers of endpoints by laying them out in two or more layers as shown in Figure 19a (this example uses the 8 targets layout from Figure 18 to allow juxtaposing the resulting layouts). When growing claim lines into tiles in step 5, endpoints are given additional "attraction". This causes tiles to inflate around their endpoints, which provides tiles with a "handle", making them easier to acquire. Figure 19b shows the resulting tile layout.

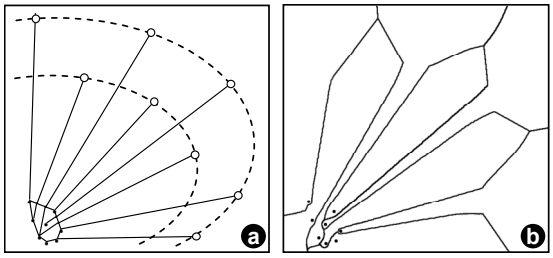


Figure 19: (a) Organizing claim line endpoints in multiple layers (b) helps thicken targets in this tile layout.

10. CONCLUSIONS

In this paper, we presented Starburst, an algorithm that extends the concept of target expansion to target layouts that contain clusters. Our user studies support our claims that the presence of target clusters limits the applicability of Voronoi-based target expansion techniques and demonstrated substantial performance benefits for the proposed Starburst technique.

As future work we plan to extend the algorithm to allow it to expand starting with arbitrary target shapes, such as buttons in graphical user interfaces. We also plan to experimentally evaluate Starburst's on-hover user interface, e.g., by comparing it against bubble cursor.

ACKNOWLEDGMENTS

We thank Heather Thorne, Raman Sarin, Tovi Grossman, and Merrie Morris for their comments on earlier drafts of this paper. Thanks to Michael McGuffin and Tovi Grossman for allowing us to use demo sequences of expanding targets and bubble cursor in our demo video.

REFERENCES

1. Albinsson, P.-A. and Zhai, S. High precision touch screen interaction. In *Proc CHI'03*, pp. 105-112.
2. Baudisch, P. and Gutwin, C. Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending. In *Proc. CHI'04*, pp. 367-374.
3. Baudisch, P., Cutrell, E., Hinckley, K., and Eversole, A. Snap-and-go: Helping Users Align Objects Without the Modality of Traditional Snapping. In *Proc. CHI'05,* pp. 301-310.
4. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P. Bederson, B., and Zierlinger, A. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-operated Systems. In *Proc. Interact'03*, pp. 57-64.
5. Beaudouin-Lafon, M. & Mackay, W. Reification, Polymorphism and Reuse: Three Principles for Designing Visual Interfaces. In *Proc. AVI'00,* pp.102–109.
6. Bell, B., Feiner, S., and Höllerer, T. View Management for Virtual and Augmented Reality. In *Proc. UIST '01*, 101-110.
7. Benko, H., Wilson, A., and Baudisch, P. Precise Selection Techniques for Multi-Touch Screens. In *Proc. CHI'06*, pp. 1263-1272.
8. Bier, E. and Stone, M. Snap dragging. In *Proc. SIGGRAPH'86,* pp. 233–240.
9. Blanch, R. Guiard, Y., Beaudouin-Lafon, M. Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation. In *Proc. CHI'04*, pp. 519–526.
10. Dion, J. (1987). *Fast printed circuit board routing.* ACM Press New York, NY, USA.
11. Fekete, J.-D., and Plaisant, C. Excentric labeling: dynamic neighborhood labeling for data visualization. In *Proc. CHI'99*, pp. 512–519.
12. Fortune, S. A sweepline algorithm for Voronoi diagrams. In *Algorithmica* 2(1):153-174, March 1987.
13. Furnas, G.W. and Qu, Y. Shape manipulation using pixel rewrites. In *Proc. Visual Computing 2002* (VC'02), published in *Proc. DMS2002*, pp. 630-639.
14. Grossman, T. and Balakrishnan, R. Bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proc. CHI 2005,* p. 281-290.
15. Grossman, T., and Balakrishnan, R. A probabilistic approach to modeling two-dimensional pointing, *TOCHI Volume 12,* Issue 3 (September 2005), p. 435-459.
16. Guiard, Y., Blanch, R., and Beaudouin-Lafon, M. Object pointing: A complement to bitmap pointing in GUIs. In *Proc. GI'04,* pp. 9-16.
17. Gutwin, C. Improving Focus Targeting in Interactive Fisheye Views. In *Proc. CHI'02,* pp. 267–274.
18. Kabbash, P. and Buxton, W. The prince technique: Fitts' law & selection using area cursors. In *Proc. CHI'95,* pp. 273–279.
19. Kakoulis, K. and Tollis, I. *Intl. Journal of Computational Geometry and Applications* 13(1):23–59. (2003).
20. Lischinski D. Incremental Delaunay triangulation. In *Graphics Gems IV*. Academic Press, pp. 47–59 (1994).
21. McGuffin, M, and Balakrishnan, R. Fitts' Law and Expanding Targets: Experimental Studies and Designs for User Interfaces. *TOCHI (*12)4:388-422, Dec. 2005.
22. McGuffin, M., and Balakrishnan, R. Acquisition of Expanding Targets. In *Proc. CHI'02*, pp. 57-64.
23. Parker, J., Mandryk, R., Nunes, M., and Inkpen, K. Improving target acquisition for pointing input on tabletop displays. In *Proc. INTERACT 2005*, pp 80-93.
24. Potter, R. L., Weldon, L. J., and Shneiderman, B. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proc. CHI'*88, pp. 27-32.
25. Preparata, F., Shamos, M. 1985. Computational Geometry: An Introduction. Texts and Monographs in Computer Science. Springer-Verlag, New York
26. Ramos, G.**,** Cockburn, A., Beaudouin-Lafon, M. and Balakrishnan, R. Pointing Lenses: Facilitating Stylus Input through Visual- and Motor-Space Magnification. In *Proc. CHI'07*. pp. 757 – 766.
27. Sears, A. and Shneiderman, B. (1991). High precision touchscreens: design strategies and comparisons with a mouse. *Int. J. Man-Mach. Stud.* 34(4):593-613.
28. Sederberg, T. and Parry, S. Free-form deformation of solid geometric models. In *Proc. SIGGRAPH 86*, pp. 151-160.
29. Swaminathan, K. and Sato, S. (1997) Interaction design for large displays. In *Interactions* 4(1):15–24.
30. Worden, A., Walker, N., Bharat, K and Hudson, S. Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons. In *Proc. CHI '97*, pp. 266–271.

Physical Handles at the Interactive Surface: Exploring Tangibility and its Benefits

Lucia Terrenghi[1], David Kirk[2] , Hendrik Richter[3], Sebastian Krämer[3], Otmar Hilliges[3], Andreas Butz[3]

[1]Vodafone GRUOP R&D
Chiemgauerstrasse 116
D-81549 Munich
lucia.terrenghi@vodafone.com

[2]Microsoft Research
7 J J Thomson Ave, Cambridge
CB3 0FB, UK
dakirk@microsoft.com

[3]LMU University of Munich
Amalienstrasse 17,
D-80333 Munich
{hendrik.richter; sebastian.krämer; otmar.hilliges;butz} @ifi.lmu.de

ABSTRACT

In this paper we investigate tangible interaction on interactive tabletops. These afford the support and integration of physical artefacts for the manipulation of digital media. To inform the design of interfaces for interactive surfaces we think it is necessary to deeply understand the benefits of employing such physical handles, i.e., the benefits of employing a third spatial dimension at the point of interaction.

To this end we conducted an experimental study by designing and comparing two versions of an interactive tool on a tabletop display, one with a physical 3D handle, and one purely graphical (but direct touch enabled). Whilst hypothesizing that the 3D version would provide a number of benefits, our observations revealed that users developed diverse interaction approaches and attitudes about hybrid and direct touch interaction.

Categories and Subject Descriptors

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

Keywords

Tangible, Hybrid, GUI, Interfaces, Design.

1. INTRODUCTION AND MOTIVATION

Progress in the field of display technologies has enabled novel forms of interactive surfaces, which often accommodate co-located input and output [7], [25], [34], thus supporting direct touch and direct manipulation [28] of digital information. The detection of multiple fingers, hands, styli and objects widens the design space for novel interaction techniques and interfaces. Furthermore, such computationally enabled surfaces can be expected to become increasingly embedded into everyday life environments, such as walls or furniture. They will be accessible to a variety of user groups and will support activities which are not necessarily related to office work. This requires the design of novel solutions, which afford social and casual interaction with digital media, and support leisure and collaborative activities, for example, browsing and sharing digital photos.

As the designers of such interactions, we have to conceive of and construct interactive systems which are attuned to the requirements of these physical and social spaces in which users are situated, in such a way as to allow us to take advantage of the rich potential of digital technology. When considering how this might be achieved a plethora of forms of interaction have been proffered but two broad classes of interactive systems in particular have begun to capture popular imagination. There are systems that support direct touch control of a graphical user interface (GUI) (e.g., [30], [35]), and those that bring tangible physical objects (TUIs) to a computationally enhanced surface (e.g. [10], [12], [16], [22], [25], [32], [33]). In each case technology is designed such that it appropriates humans' manipulation skills and mental models gained from interactions with the physical world and integrates them with the extensive possibilities of digital media.

The two approaches, though, are different in aspects of physical interaction that are drawn upon in the design of hybrid, physical/digital systems. In the case of GUIs for direct touch, designers often rely, for example, on the metaphorical 2D representation of physical artefacts to suggest the hand gestures or marking strokes to be operated. In the case of TUIs, designers exploit the degrees of freedom, manipulation vocabulary and haptic feedback enabled by the 3rd spatial dimension of the physical transducer.

Thus, when designing such systems, designers have mostly created and exploited design principles from either WIMP-based interaction (i.e., GUI design) or physical interaction (i.e., product design). Most of these principles are derived either from the comparative observation of physically enhanced vs. WIMP-based interaction (e.g., [2], [23]), or from the dedicated analysis of one of the two (e.g., [13], [18]). Although this has produced valuable insights, which are also fostered by ergonomics, cognitive psychology, and sociology (e.g., [8], [14], [19]), the spatial combination of physical manipulation and display of digital output in direct touch interactive surfaces creates new design challenges and opportunities, indeed there is significant potential, given advances in technology, to construct 'Hybrid' interfaces which combine elements of *both* tangible interaction and the ability to perform direct touch style manipulations with digital/graphical representations. With this potential functionality then, the previous evaluative studies do not provide significant guidance as to the relative benefits of when and how to exploit the '3rd Dimension' in an interaction scenario. If the facility for essentially manipulable 2D graphical content is concomitant, why design into a 3rd dimension, and if one does, what impact might this have on the user's behaviour?

To investigate the effect of tangibility and physicality more closely, we built 3D and 2D versions of PhotoLens, a system for photo browsing on an interactive tabletop. Herein we present our design rationale for the 3D PhotoLens, discussing it in relation to what existing research literature suggests are potential benefits of tangible devices. We then present a comparative evaluation study

AVI'08, May 28-30, 2008, Napoli, Italy.

wherein users explored both this interface and the 2D graphical alternative (direct touch enabled). This allowed us to evaluate how pushing the interaction into a tangible 3rd dimension influenced patterns of user behaviour. We discuss the observed design implications of doing this and highlight key questions which arise.

2. RELATED WORK

Integration of aspects of physicality in the design of interactive systems has followed different paths of "material embodiment" and "metaphorical representation" [9] as technology has matured.

The seminal work on Toolglasses and Magic Lenses by Bier et al. [3] introduced a see-through style of GUIs, which metaphorically evoked filters. These interfaces were operated with two hands, using touch-pads, track-balls or mice as input (i.e., the input region was detached from the output display). One of the main benefits was to afford two-handed interaction, thus overcoming mode changes and taking advantage of human bimanual skills (which in turn shows cognitive and manipulation benefits [20], especially when the hands do not perform simultaneously [4]).

Fitzmaurice et al.'s work on Bricks and the ActiveDesk [10] goes a step further in this direction: The materiality and "graspability" of the bricks as input devices (which have more degrees of freedom and a richer manipulation vocabulary than the mouse), together with the direct contact between input object and output surface, aimed at "facilitating two-handed interactions, spatial caching, and parallel position and orientation control" [10]. This work forms the basis for the Tangible User Interfaces paradigm. The main benefits claimed in this area of research are intuitiveness [15], motor memory [19], and learnability [26].

Because of the physical affordances of the table, such as horizontal support for physical artefacts, several instantiations of the TUI paradigm can be found in conjunction with tabletop displays: These are usually ad-hoc designed tools, whose formal shape can more (e.g., [32]) or less (e.g., [12], [25]) literally represent a metaphor. Furthermore, the integration of such physical artefacts in the design of applications for multi-user tabletop displays is often motivated by the goal of supporting casual co-located collaboration, as suggested for example in [16], [22], [33].

The use of tangible interaction is claimed to be beneficial for collaborative work and group awareness [8], [14], as it implies the mutual visibility of explicit actions among participants [27].

This work on the interweaving of physical and digital aspects in interface design for interactive surfaces suggests a variety of benefits: cognitive (e.g., intuitiveness and learnability), manipulative (e.g., motor memory), collaborative (e.g., group awareness), experiential, as well as in terms of efficiency. But the empirical work that supports such claims is actually limited and mostly focuses on one aspect in isolation from the others, thus taking for granted, to some extent, some of the benefits of integrating aspects of physical interaction in the design of hybrid ones. From our perspective, we think that the mutual influences of the different aspects cannot emerge if we do not start distinguishing what are the very aspects of physical interaction we integrate in the design of hybrid, physical-digital interactive systems, while considering, at the same time, their implications on different levels of the experience of use (e.g., discoverability of the interface, easiness, fun). These aspects become crucial when we expect interactive surfaces to support everyday life including causal and leisure interactions.

To address these issues we draw upon the aspects of physical interaction for the design of hybrid systems suggested by Terrenghi et al [31]. Such aspects are: metaphorical representation; space-multiplex input; direct spatial mapping between input and output; continuity of action; 3D space of manipulation; rich multimodal feedback. Through the comparative analysis of design solutions that integrate (or not) some of these aspects, we can then start eliciting the effects and implications of such integrations more consciously.

3. DESIGNING THE PHOTOLENS

To unpack tangibility and its effects on interaction behaviours, we built the PhotoLens system, a hybrid tool for browsing and organization of photos on an interactive tabletop display. The choice of developing an interface for photo-browsing is particularly linked to this notion of evolving interaction paradigms being tethered to the support of digital interactions in more social and casual areas. The rapid shift of photography from analog to digital, together with the reduced cost of taking pictures, has caused a substantial growth of personal photo collections, and the technology that we use to capture, display and interact with them [4]. On the other hand, the size and orientation of the displays of Desktop PCs, together with their WIMP paradigm, neither provide social affordances suitable for co-located sharing and collaborative manipulation and organization of collections (an imperative feature of users' interactions with photos [11]), nor the creation of temporal spatial structures, as our physical artefacts do [18].

In the envisioned scenario, the collections of different users (e.g., friends, family members) can be displayed on the tabletop. Photo collections are visualized in piles, in analogy to [21] and [1]. Piles can be freely translated on the tabletop (i.e., no automatic snapping to a grid) by touching and dragging the image on the top with a finger or with a stylus (see Figure 1, a). In order to save real estate and avoid clutter, we use PhotoLens to gain a localized, unfolded view of the pictures contained in one pile, without interfering with the information landscape of the shared display (see Figure 1, b and c). For a complete overview of how the PhotoLens works see figure 1 and the description below.

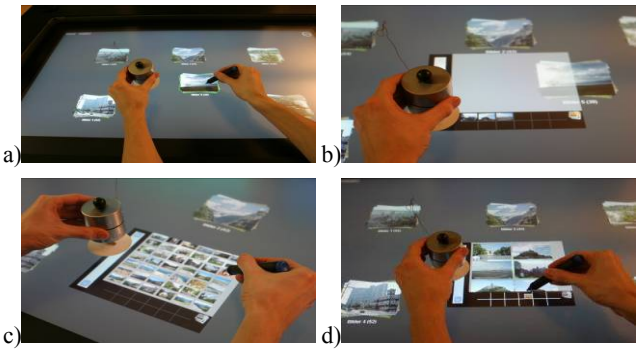


Figure 1: Interaction with the 3D PhotoLens:

The illustrations in figure 1, show how the PhotoLens works: a) Piles can be moved freely on the table using the stylus. b) The digital lens only appears when the physical tool is placed on the table. c) The pile unfolds in a thumbnail view and moving the handle up and down the scroll bar scrolls through the thumbnails. d) The view can be zoomed in and out by rotating the upper part of the tool and selected pictures can be copied to a temporary tray (retained independent of the pile viewed). Additionally, a new pile containing photos from different collections can be created by tipping on the icon in the right bottom corner of the lens.

3.1 Rationale and Expectations

Previously Terrenghi et al. [31] have observed that despite some interactive systems allowing for bimanual interaction on a display

(which is known to offer both physical and cognitive benefits [20]), people tend to use only one hand and preferably the dominant one when manipulating digital media, possibly due to their acquaintance with the WIMP paradigm. Therefore we expected the use of a physical tool, associated with a digital frame and a stylus for interaction, to more explicitly suggest two-hand cooperative work. Indeed, by providing both a tool and a stylus we wanted to suggest the use of the non-dominant hand for navigation tasks (i.e., grasping and rotating the tool) and of the dominant hand for fine-grained tasks (i.e., selecting and dragging pictures). The stylus is indeed typically held with the dominant hand, so that we expected users to use the non-dominant hand for interacting with the physical tool in order to use their hands cooperatively, such as in Guiards' kinematic chain [13]. To make this affordance even more explicit, and given predominant right-handedness, we designed the graphical lens so that it would extend on the right-up side of the physical tool (see Figure 1, b). We then mapped navigation functionalities, e.g. placing (appearing of the lens frame), scrolling and zooming to the physical tool.

Additionally, we expected that the physical affordances of the tool, like placement and rotation, would support the offload of cognitive effort thanks to the haptic feedback it provides. The tool, indeed, can be operated without looking at it, thus not hindering users' visual attention. The effect of its manipulation is mapped in real time and in the same area (e.g., zooming and scrolling of the pictures in the lens), thus providing an isomorphic visual feedback of action. In this sense we expected that the continuity of action it supports (rotation and translation) and the multimodal feedback (haptic and visual) would provide a higher sense of control. In this sense we refer to Buxton's work on the effect of continuity of action on chunking and phrasing [5], as well as on Balakrishnan and Hinckley' investigation on the value of proprioception in asymmetric bimanual tasks [2].

Since the graphical lens appears when the tool is placed on the table, and disappears when the tool is lifted, we expected this feature to support an efficient use of the real estate: users could indeed display the lens only when required. Furthermore, the fact that the lens can be physically picked up in the 3D space and moved to another pile, makes it unnecessary to drag it in 2D across the screen, stretching arms and sidling between other piles, thus providing motor benefits.

Although we are aware of the social benefits of tangibility claimed in the related literature, our current technical setup only recognizes two input points (i.e., interaction with only one PhotoLens at a time). Thus, interactions with the system are in this instance based on individual action, which makes the social affordances of such interfaces a consideration for future work.

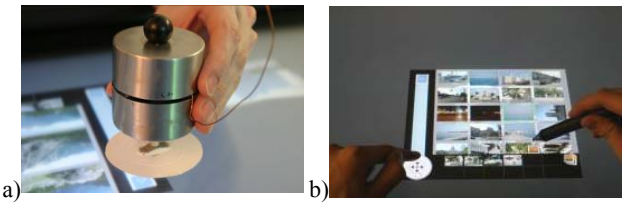


Figure 2: a) The physical component of the 3D PhotoLens. b) The purely graphical 2D PhotoLens.

3.2 Technical Implementation

The technical setup of PhotoLens consists of an interactive table and a modified wireless mouse for the implementation of the physical handle. The components of the mouse were rearranged in a metal cylinder with a diameter of 7 cm and height of 9 cm, which we took from a disassembled kitchen timer (see Figure 2, a). The size of the tool is determined by the features of the mouse.

The interactive table consists of an LCD monitor with a resolution of 1366 x 768 pixels in 16:9 format and a diagonal size of 100 cm, embedded in a 15 cm wide wooden frame. Input is provided by a DViT [29] overlay frame, which uses four cameras in the corners of the frame to track two input points simultaneously. An input point can either be a pen, a tool, or simply a users' finger. The frame we use has limitations when wide input points are on one of its diagonals, as this causes a mutual occlusion. The thinner the body of the input mediator, the lower the risk of occlusion, and the more accurate the tracking, for this reason we created a base for the physical tool (see Figure 2, a), so that its stem creates a smaller shadow and hence provides more accurate tracking.

3.3 Constructing a Comparative Graphical PhotoLens

For comparative purposes our 2D PhotoLens had inherently the same functionality as the 3D version, it was a direct touch enabled graphical interface, but did not extend into the 3rd dimension. Lacking a physical handle for picking it up, the 2D PhotoLens is permanently displayed on the tabletop and can metaphorically overlap photo piles when it is dragged onto them. The control for scrolling and zooming of the PhotoLens is represented by an interactive circle, as illustrated in Figure 3.

Figure 3: Screen shot of the 2D PhotoLens.

When a user touches the small circle on the graphical control wheel and slides her finger along the circular trajectory of the graphical control, clockwise rotation zooms in and counter-clockwise rotation zooms out. When the user touches the center of the same graphical wheel, four perpendicular arrows appear (see Figure 2, b): these resemble the symbol of movement used in the GUI of several desktop applications (e.g., Microsoft PowerPoint, Adobe Photoshop). Sliding the finger up and down along the line of the scrollbar, the thumbnails scroll up or down, as in a desktop GUI. When the control circle is touched and dragged away from the pile for more than 5 cm, the whole lens moves along, for example onto another photo pile or into an empty area of the table.

4. STUDY METHODOLOGY

4.1 Study Design

To engage users with the interface they were asked to bring a sample of 80 personal digital photos (from a trip or vacation) to the study session. During study trials participants completed two tasks with their photos using both interfaces (i.e. 3D and 2D, whose order of execution was counterbalanced across trials and participants), giving a total of 4 trials. In each trial participants were presented with 6 piles of 80 photos (80 random images from their own collection in one pile, with other piles being made up of images provided by the researchers, and used to simulate the presence of a companion's images). In one trial the participants were told to interact with only their pile, selecting 12 images suitable

for use as desktop wallpapers and in the other trial they interacted with all of the piles, searching for 12 images to accompany a proposed calendar. In both cases participants were told to store selected images in the PhotoLens temporary tray creating a new pile on the tabletop.

Before each task, the user had that current task explained and the interface demonstrated (including demonstration of the potential for using two handed interactions). After the trials, the participants completed an evaluation questionnaire and discussed their experiences with the experimenter in charge of the session.

Our participants were 12 right-handed adults (mostly university students, with different majors, in an age range from 20 to 30 years old), comprised of 6 men and 6 women, all with normal or corrected to normal vision, and all having normal arm mobility.

4.2 Analysis

To help ground our deeper analysis and to understand broader patterns of action at the interface, we calculated the extent of use of differing forms of interface manipulation (i.e. different forms of handed interaction) in the two conditions. And then to ground these actions in a more reflective consideration of subjective response to differing interface styles, we solicited feedback of users' perceptions of use from their experiences of the two interfaces, (using Likert scales from 1 to 5, negative to positive) to get general response on key characteristics such as ease of use and enjoyment, and certain specific manipulative actions such as *zooming*, *scrolling,* and *placing* the lens (see Figure 13).

All user trials were video recorded; our evaluation is mainly based on direct consideration of these video materials. The footage was studied by an interdisciplinary design team and subjected to an interaction analysis [17]. The focus of the analysis was to look for patterns of common interaction strategies and specific moments of novel interaction, or moments when the interaction faltered. Attention was also given to moments of initiation of interaction. This approach to the data was taken as it was felt more appropriate than traditional attempts to exclusively quantify behaviours at the interface. The paradigm of digital interaction that was being explored, i.e. leisure technology (photo browsing in this case) does not fit a traditional model of recording task completion times. It was felt that by taking a fine-grained, micro-analytic approach to recovering patterns of activity and breakdown during interface interaction a richer understanding could be derived of how, qualitatively, a third dimension in an interface was appropriated and understood by users. Consequently the ensuing results section seeks to articulate some vignettes of interaction, some moments of user activity, which we felt were of particular interest and were particularly illuminating in our attempts to understand the impact of tangibility on interactive behaviour.

5. RESULTS

Our results are split into two sections, the first highlights some patterns of handed interaction at the interface, the second providing a more detailed view of some of the common elements of interaction during tasks.

5.1 Forms of Handed Interactions across Modalities

As we can observe in Table 1, our participants demonstrated diverse approaches to interacting with the interface which might suggest that they were developing different mental models of system function or simply approaching the interface with different pre-conceived manipulation skills, habits and preferences for physical and digital media.

We observed 5 predominant forms of interaction with the interface as shown in Table 1, logically conforming to those actions immediately possible (NB, none of the participants, selected the photos with the non-dominant hand). These broader patterns of action framed our subsequent inquiry and, thus, our interaction analysis partially draws on such a classification of conditions to identify, analyze and describe snippets of interactions which we found relevant for what can be considered a 'catalogue of interaction experiences', which we articulate and present below.

Table 1. Average percentage of time spent in differing forms of handed interactions in both physical (3D) and purely graphical (2D) conditions (standard deviations in brackets).

Forms of handed interactions	3D	2D
Two-handed interaction with PhotoLens.	9.0% (11.9)	19.4% (24.7)
The non-dominant hand interacts with the control wheel for scrolling and zooming.	44.7% (15.6)	37.5% (24.8)
The dominant hand interacts with the control wheel for scrolling and zooming.	2.1% (5.6)	17.4% (24.6)
The dominant hand interacts with the photos for selection tasks.	31.3% (21.6)	17.0% (11.2)
No hands are on the interactive area	12.9% (6.4)	8.8% (7.8)

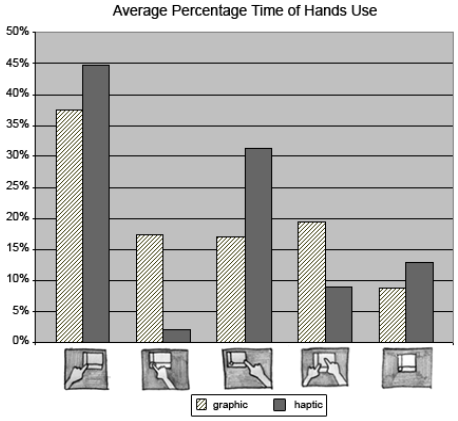


Fig. 4. Representation of average percentage of time spent in differing forms of handed interactions.

5.2 A Catalogue of Interaction Experiences

In this section we present vignettes of interaction following the common life-cycle of interface activities during elements of the photo-browsing task.

Approaching the Task

At the beginning of the task, in both modalities, the participants were asked to select 12 photos from their own pile, which was displayed in the bottom right corner of the table. Piles could be moved freely across the table, so as to enable epistemic actions, i.e., allow users to create spatial arrangements as they liked and found more comfortable for interaction. Despite such a feature, we noticed some interesting differences amongst subjects in the way they approached the task and the posture they adopted.

The participant in Fig. 5, for example, first moves the pile in front of her away using the stylus in her right hand, gaining space; then she moves her pile from the right to the center of the table. In this way she creates a focused interaction area, where she can easily

visualize and reach the photos of her collection/pile. She than grasps the physical handle from the border of the table with her left hand, and starts browsing through the photos.

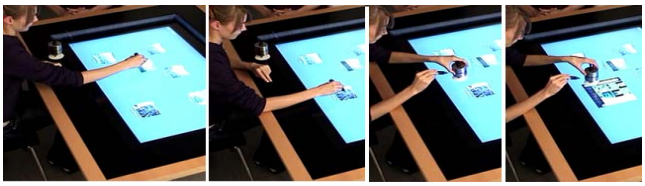


Fig. 5. Moving the artifacts towards the body.

A different interaction style can be observed in Fig. 6, where the participant moves her body towards the pile to be sorted, rather than the alternate. In this case she first places the physical handle on the screen of the table with the dominant hand; she then drags it on the table towards the pile in the right bottom corner. Thus, in order to better reach the interaction area, she moves the chair to the right side of the table, in the proximity of the pile she wants to sort, and she then starts interacting with the PhotoLens.

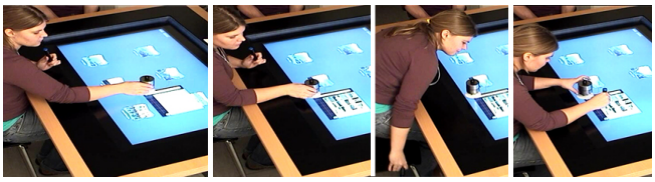


Fig. 6. Moving the body towards the artifacts.

Browsing the Photo Collection by Scrolling and Zooming.

By rotating and sliding the control wheel (either the 3D or the 2D one) users could browse thought the photo collection, thus exploring the content of the pile. Our design choice of placing the control wheel at the left bottom corner of the lens was meant to afford two-handed manipulation of the PhotoLens, and manipulation of the control wheel with the non-dominant hand. This was not, however, always the approach taken by our participants.

In Fig. 7 the participant interacts with the control wheel with the pen, held in the dominant hand, while the non-dominant hand is rested on the border of the table. In this way the participant partially occludes her own view, which brings her to alternatively lift the pen and her hand from the table to better see the pictures in the thumbnail view (e.g., second frame of Fig. 7). Furthermore, as she explained in the post-test questionnaire, she found it more difficult to manipulate the small sensible area of the 2D wheel for zooming, in comparison to grasping the physical handle: We can speculate that this is why, as we could observe in the video analysis, in the 2D modality she mostly used the scrolling function of the wheel to browse through the photo collection, but hardly changed the zooming factor.

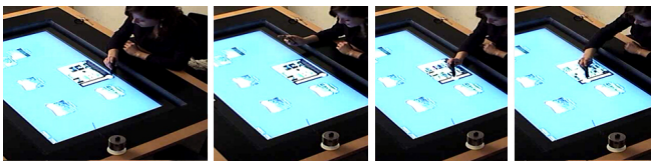


Fig. 7. One-handed interaction with the 2D PhotoLens.

Alternatively, when interacting with the 3D PhotoLens, she manipulated the physical control wheel with the non-dominant hand only, exploring the content of the collection both with scrolling and zooming (e.g., see the third frame in Fig. 8). In such an interaction pattern, both the hands were kept on the interactive area of the table during the whole interaction with one pile.

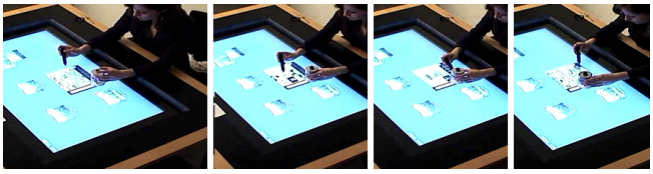


Fig. 8. Two-handed interaction with the 3D PhotoLens.

Selecting Photos in the Lens.

By providing our participants with a stylus we expected them to interact with the dominant hand for selection tasks: none of the participants (who were all right handed), indeed, performed selection tasks with the non-dominant hand. Additionally, because of the laterality of the control wheel and of the scrolling bar, we expected interaction patterns similar to drawing ones [13] to emerge. In these cases a tool (e.g., a ruler) is usually held with the non-dominant hand, while the dominant one performs micrometric tasks in the proximity of the tool (e.g., draws a line). The type of patterns we assisted to were often rather different across modalities, though, in the way people alternatively or simultaneously used the non-dominant and dominant hand.

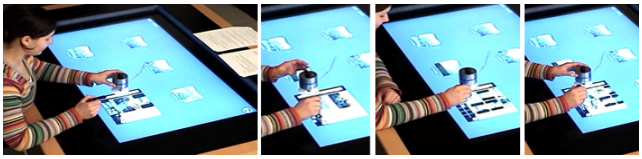


Fig. 9. Alternate use of the dominant and non-dominant hands with the 3D PhotoLens

As we can see in Fig. 9, as an example of interaction with the 3D PhotoLens, the participant first positions the physical tool on a photo pile with the non-dominant hand, and starts browsing through the photos by scrolling and zooming. In this phase she keeps the dominant hand in the proximity of the interactive area, holding the stylus. After she has set a preferred height in the scroll bar, and a desired zooming factor, she then releases the non-dominant hand (Fig. 9, second frame) and rests it at the border of the table (Fig. 9, third frame). She then proceeds in the task by selecting the photos with the dominant hand: Such a cycle of interactions unfolds again when the zooming and scrolling are newly set with the non-dominant hand (Fig. 9, fourth frame).

Surprisingly, in the 2D modality participants kept more continuously both hands simultaneously on the interactive area (see time percentage in Table 1). As shown in Fig. 10, for example, the participant keeps his left forefinger on the 2D control wheel during the whole interaction with a pile: I.e., both when the dominant hand is selecting photos (e.g., second and third frame) or it is just held in the proximity of the lens (e.g., frame 4).

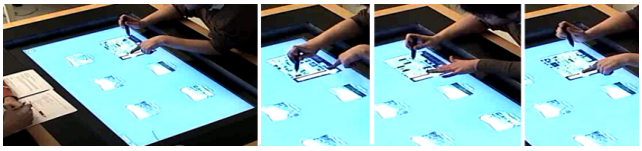


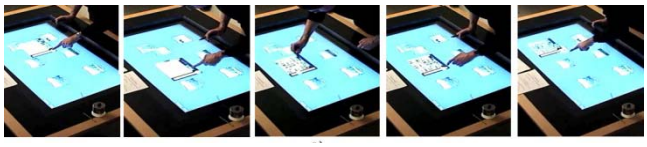
Fig. 10: Concomitant use of the dominant and non-dominant hands with the 2D PhotoLens.

Although the 2D graphic PhotoLens is permanently present on the interactive surface, and can be moved on the table only when it is dragged, several participants mentioned in the post-test questionnaire that they constantly kept their fingers on the wheel as they had the feeling that the lens would disappear otherwise.

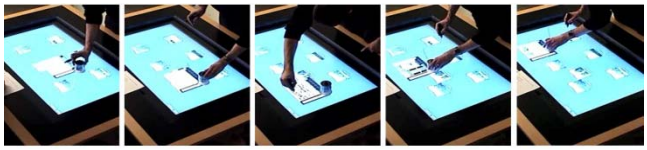
Placing and Moving the PhotoLens.

When participants were asked to create a new collection by selecting photos across several piles on the table, different strategies of moving the lens and photos could be noticed, showing differences among both subjects and modalities in how people took the tool to the pile or vice versa.

In Fig. 11 we can observe how a user interacts with the 2D (Fig 11, a) and the 3D PhotoLens (Fig. 11, b). To reach the piles he stands up. In the 2D modality he drags the lens towards different piles with a finger of the non-dominant hand. When selecting photos from one collection (e.g., third frame Fig. 12, a) he rests his non-dominant hand on the border of the table: he than uses it again for moving the lens towards another pile (e.g., fourth and fifth frame Fig. 12, a), while resting the right hand to the border this time. All in all, he never moves the piles, and alternatively uses the non-dominant and dominant hand for respectively moving the lens on the table and selecting photos within the lens. In the 3D modality he adopts a very similar strategy. He first places the physical handle with the dominant hand on a pile: then he swaps hands for browsing, and again for selecting. In these cases one of the hands is always rested at the border of the table. In order to move the lens towards another pile he slides the physical tool on the table surface (e.g., fourth and fifth frame in Fig.11).

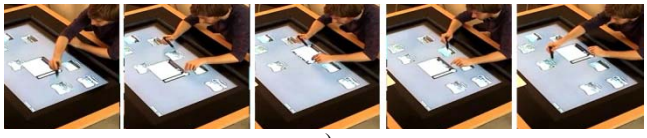


a)



b)

Figure 11: Moving the tool and the body towards the piles: a) 2D PhotoLens; b) 3D PhotoLens.



a)

b)

Figure 12: Moving the tool and the piles towards the body: a) 2D PhotoLens; b) 3D PhotoLens.

A different approach can be observed in Fig. 12. In this case the participant tends to move the piles and the lens towards his body. In the first frame of Fig. 12, a, he drags a pile towards himself with the dominant hand: with the non-dominant one (second and third frame) he than moves the 2D Photolens towards the pile to interact with it. In the fourth and fifth frame, he moves other piles towards himself with the dominant hand, while slightly moving the PhotoLens between one interaction cycle and another one with the non-dominant hand. The interaction takes place in the proximity of his body, and the dominant and non-dominant hands are alternatively used for moving respectively the piles and the lens.

When interacting with the 3D PhotoLens (Fig. 12, b) he adopts a similar allocation of tasks to dominant and non-dominant hand (i.e., moving the piles and the lens accordingly). In this case he takes advantage of the graspability and mobility of the physical handle in the 3D space to alternatively place it at the border of the table (e.g., second and fifth frame in Fig. 12, b).

5.3 Perceived Experience

Figure 13 reports the results of post-test questionnaires (average values on a Likert scale). Despite the physical control being easier to use on average (with a remarkable difference in ease of use between the two interfaces for the zooming function in particular), participants reported that overall it is more fun to use their hands on the screen than the tool. To explore this response a little more it is worth referring to participants' comments.

For some the 3D PhotoLens was easier to use, especially in the zooming function, as it does not require such precise interaction as with the graphical wheel. In this respect they told us: "With the physical tool you only have to rotate"; "With the physical tool you don't have to think about what you can do, you see it immediately"; "You don't need to look for the exact point where to put your finger to rotate"; "The rotation for zooming reminds the use of analogue cameras"; and finally "it is easy to place it and rest it in one position: with the digital lens I had the feeling I needed to hold it in place". When considering *why* the graphical interface is fun to use, participants cited such factors as: "It is more natural to interact *directly* with your hand than with a device"; "With your hand you are *directly* on the image, the tool is too far away from it"; "You need to get used to a device, sometimes the zooming with the tool is too fast, you have a better control with your hand directly"; "When you interact with the tool you don't have the feeling 'on the finger tips' of where the scrollbar ends". Such comments raise interesting questions about subjective perceptions of directness, control, haptic feedback, discoverability, easiness and enjoyment of interaction, especially when the interaction purposes are not merely linked to models of efficiency and performance. Aspects of easiness and enjoyment of interaction, for example, do not appear to be causally related.

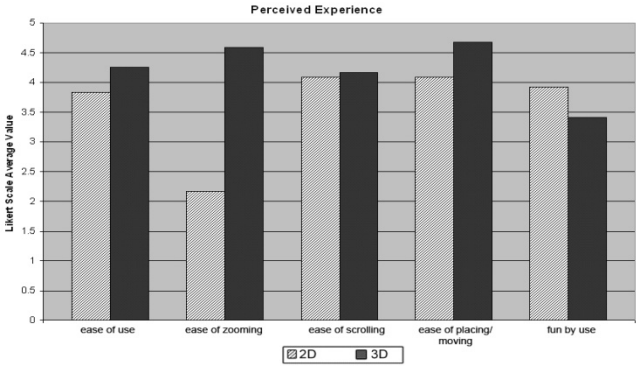


Figure 13: The results of perceived experience in terms of average of the Likert scale values.

6. DISCUSSION

Having presented vignettes of action and grounded them in details of common practice, it is germane to discuss implications of these observations for our discussion of tangibility. The appropriation of an experimental methodology allowed us to inform our critical enquiry of tangibility by forcing users into making comparative use of two functionally similar but fundamentally altered interfaces. By forcing this comparative evaluation with a direct-touch enabled GUI, we have been able, perhaps more explicitly than in past studies [23], to explore the effects of pushing an interface

into a 3rd dimension. Our analysis followed the common life-cycle of interactions at the interface during photo browsing and manipulation; flowing from the initiation of contact, through pile browsing, selecting images and then moving the 'lens' onto new piles and iterating. From observation of each of these common stages of interface use we feel that there are three key aspects of activity raised that we should discuss further, *Idiosyncrasy of action*, *Concomitant bimanualism* and *Sequential action and laterality*.

6.1 Idiosyncrasy of Action

Not all of our participants used the interface similarly: which means that individual actions were often highly idiosyncratic regardless of the interface that participants used, as shown by the standard deviations presented in Table 1. Even in our first stage of analysis, considering the initiation of interaction, participants clearly approached the task (bodily) in different ways. Some understood that piles of pictures could be dragged towards themselves and others relied on moving a tool to the digital objects of interest. This latter form of interaction potentially demonstrated an existing mental model, perhaps created from years of WIMP use, where the fundamental paradigm is to manipulate an interceding tool and take that to the objects of interest (e.g. tools mediated by mouse movement in a Photoshop environment). This is as opposed to bringing artefacts of interest to the tool of use, such as might happen in the real world (e.g. 'examining' or 'framing' tools like microscopes). None-the-less such patterns of interaction at the interface were not consistent across all subjects, although this is perhaps to be expected with such open interfaces and relatively open tasks.

This idiosyncratic action has two implications. Firstly it highlights the issue of 'discoverability' at the interface begging reflection on some of the claimed benefits of intuitiveness of the interface in some of the TUI literature [15], [19]. We had designed the 3D interface to suggest a style of use. However, during our study (which represents users' initial explorations of such interfaces) many did not use the interface as intended. Some failed to discover for themselves our prompted scheme of interaction. This suggests that even if an interface is designed to incorporate a 3rd dimension, there is no guarantee that all users will appropriate it as the designer intends, so some of the performance benefits expected will not materialize. This strongly suggests that consideration be given to ensuring that 3D interface elements have an inherent level of discoverability. Especially if a specific style of interaction (e.g. bimanual) purportedly offers some kind of benefit.

Secondly, however, this observed idiosyncrasy potentially implies that one should perhaps design for conflicting user preferences. In this open scenario, with a less constrained study task than in some previous experiments [20], we saw that users adapted their use of the interface to suit factors such as comfort (hence the one handed interactions). If this is how users are going to act, perhaps we should in future be less concerned with the *a priori* shaping of the minutiae of interaction (such as appropriate handed interactions). Instead, we should actively consider designing tangible elements that can be appropriated by the user in personal ways.

6.2 Concomitant Bimanualism

This form of interaction refers to users using both hands simultaneously to operate the interface. Relatively speaking, this did not happen that much, however, when it did happen it was more likely to occur in interactions with the 2D interface than with the 3D interface. The reason given for this by the users appears to centre on mistaken mental models of the operation of the 2D interface. Some of the users really felt that if they took their left hand away from the surface the Lens would disappear (contrary to what they were shown). Conversely, for these people the physical handle of the 3D interface held some form of object permanence: once placed, the physical handle was comfortably left alone.

Here then, our choice of a comparative analysis has been particularly beneficial. Had we not had the comparison with a 2D interface we would have had a poorer understanding of the effects of using a 3D handle, seeing sequential actions during its use and assuming that this was entirely user-comfort driven. From understanding the bimanual response to the 2D interface we see that an implication of building into the 3rd dimension, beyond apparent user comfort, is the inherent substantiality of a 3D interface control creates assurances of consistent action. A benefit of 3D elements is possibly therefore that they suggest a more consistent and accurate control than a comparable 2D interface.

6.3 Sequential Action and Laterality

Previous research [31] suggests users of such interfaces utilise one-handed interactions, and we also assumed that our interface design would promote a lateral division of handed interactions. For a large number of users this was often the pattern of behavior observed. Particularly those using the 3D interface rather than the 2D. So in this respect our design solution worked and we can confirm that the introduction of a tangible 3D element to the interface appeared to support the lateral division of handedness, promoting bimanualism (albeit sequential rather than concurrent). Given research [20] has suggesting performance benefits of bimanualism we have effectively observed a benefit from pushing the interface into a 3rd dimension.

However, there is a problem posed by the questionnaire data. Previous work discussing the benefits of tangibility has taken a more engineering led approach to the evaluation. They have considered metrics of performance such as speed and task completion, and in this respect some of our questionnaire results concur with their findings, subjective responses from our users suggest that there were performance benefits for the 3D interface. However, this critically conflicts with their perceived *preference* for the 2D interface, which they found more *fun to use*. And it is the reasoning behind this which is of particular interest here. It appears that certainly for some users there was a significant increase in the perception of direct engagement with the 2D interface. Contrary to regular expectations that tangibility and three-dimensionality enhance physical engagement with digital information, we would suggest that such a process can perhaps, unwittingly, create a perceptible barrier between user and data. In the TUI ideal, physical elements are both input and output. From testing our own design we would suggest that if the 3D elements of an interface are not deeply considered they can unfortunately all too easily traverse a hidden line into becoming just another tool for *mediating* action at the interface, another form of 'mouse'. The level of direct engagement between user and digital artifact can be less than that found in direct touch enabled GUIs and consequently, it seems, this impacts user enjoyment., which is after all, the *critical metric* for evaluating interactions with leisure technologies.

7. CONCLUSION

While the field of interactive surfaces is still in its infancy, we think that through the design of interactive systems which consciously combine physical and digital affordances, and the systematic evaluation thereof, we can learn about people's interaction schemas. To this end we need to investigate what the very differences, benefits and trade-offs of physical and digital qualities in the interaction actually are, and how they affect the user experience in different contexts. Which solutions provide the best mental model for bimanual cooperative work? Where shall we draw

the line between graphical metaphorical representation and embodiment of the functionalities in a physical tool? Agarawala et al.'s recent work [1] on physics enhanced desktop-metaphors makes an interesting case for this discussion: in this work, physics-based behaviors are simulated so that icons can "be dragged and tossed around with the feel of realistic characteristics such as friction and mass". Accordingly it is timely to explore the borders and influences between the *look* and the *feel*, the *visual* and the *haptic*. In this respect, our research agenda is to pursue comparative design and evaluation, contributing to a deeper understanding of human interaction behaviour through the design of comparable solutions which tackle specific aspects of the interaction (e.g., physicality and tangibility), and at the same time provide experiences which are open for people's expression of preferences and relate to realistic everyday life scenarios (e.g., photo browsing). The main focus then of our comparative evaluation is not the success of design solutions *per se*, but rather on the discovery and understanding of factors affecting user experience. By combining empirical and explorative approaches, we attempt to recognize patterns which shed light on relationships between design solutions and resulting experience, informing the design of hybrid systems.

8. ACKNOWLEDGMENTS

We kindly thank all the participants of our study.

9. REFERENCES

[1] Agarawala, A., Balakrishnan, R. Keepin' it Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen. In *Proc. of CHI'06,* 1283-1292.

[2] Balakrishnan, R. and Hinckley, K. The role of kinesthetic reference frames in two-handed input performance. In *Proc. of UIST '99*, 171-178.

[3] Bier, E.A., Stone, M.C., Pier, K., Buxton, W. and DeRose, T.D. Toolglass and magic lenses: The see-through interface. In *Proc. of SIGGRAPH '93,* 73-80.

[4] Buxton, W., and Myers, B., A. A Study in TwoHanded Input. In *Proc. of CHI '86* , 32-326.

[5] Buxton, W. Chunking and Phrasing and the Design of Human-Computer Dialogues. In *Proc. of the IFIP World Computer Congress*, Dublin, Ireland, 1986, 475-480.

[6] Crabtree, A., Rodden, T., and Mariani, J. 2004. Collaborating Around Collections: Informing the Continued Development of Photoware. In *Proc. of CSCW '04,* 396-405.

[7] Dietz, P., Leigh, D. DiamondTouch: A Multi-User Touch Technology. In *Proc. of UIST'01*, 219-226.

[8] Dourish, P.: Where the Action is. The Foundations of Embodied Interaction. *Bradford Books*, 2001.

[9] Fishkin, K. P. A Taxonomy for and Analysis of Tangible Interfaces. *Journal of Personal and Ubiquitous Computing*, 8 (5), September 2004, 347-358.

[10] Fitzmaurice, G., Ishii, H., Buxton, W. Bricks: Laying the foundations for graspable user interfaces. In *Proc. of CHI'95*, 432-449.

[11] Frohlich, D., Kuchinsky, A., Pering, C., Don, A., and Ariss, S. 2002. Requirements for photoware. In *Proc. of CSCW '02*, 166-175.

[12] Gorbet M.G., Orth M., Ishii H., Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography. In *Proc. of CHI ' 98*, 49-56.

[13] Guiard, Y., "Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model," *J. Motor Behaviour, 19 (4), 1987, 486-517.*

[14] Hornecker, E., Buur, J.: Getting a Grip on Tangible Interaction: A Framework on Physical Space and Social Interaction. In *Proc. of CHI 2006*, 437-446

[15] Ishii, H. and Ullmer, B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proc. of CHI'*97, 234-241.

[16] Jordà, S., Geiger, G., Alonso, A., Kaltenbrunner, M. The reacTable: Exploring the Synergy between Live Music Performance and Tabletop Tangible Interfaces. In *Proc. of TEI'07.*

[17] Jordan and Henderson. Interaction Analysis: Foundations and practice. *Journal of the Learning Sciences*. 4 (1), 1995

[18] Kirsh, D. The Intelligent Use of Space. In *Artificial Intelligence*, 73, 1995, 31-68.

[19] Klemmer, S. R., Hartmann, B., and Takayama, L. How Bodies Matter: Five Themes for Interaction Design. In *Proc. of DIS '06*, 140-149.

[20] Leganchuk, A., Zhai, S., Buxton, W. Manual and Cognitive Benefits of Two-Handed Input: An Experimental Study. In *Transactions on Computer-Human Interaction*, Vol5 (4), 1998, 326-359.

[21] Mander, R., Salomon, G., and Wong, Y. Y. 1992. A "pile" metaphor for supporting casual organization of information. In *Proc. CHI'*92, 627-634.

[22] Mazalek, A., Reynolds, M., Davenport, G. TViews: An Extensible Architecture for Multiuser Digital Media Tables. *IEEE Computer Graphics and Applications*, vol. 26, no. 5, pp. 47-55, Sept/Oct, 2006.

[23] Patten, J. and Ishii, H. 2000. A comparison of spatial organization strategies in graphical and tangible user interfaces. In *Proc. of DARE'00*, 41-50.

[24] Rekimoto, J. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proc. of CHI '01*, 113-120.

[25] Rekimoto J., Ullmer B., and Oba H., DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions. In: *Proc. of CHI'01*.

[26] Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., and Silverman, B. 1998. Digital manipulatives: new toys to think with. In *Proc. CHI'98*, 281-287.

[27] Scott, S.D., Grant, K., D., & Mandryk, R.: System Guidelines for Co-located, Collaborative Work on a Tabletop Display. In *Proc. of ECSCW '03*, 159-178.

[28] Shneiderman, B., Direct manipulation: A step beyond programming languages, IEEE Computer 16, 8, August 1983, 57-69.

[29] SmartTech DViT http://www.smarttech.com/DViT/

[30] Terrenghi, L., Fritsche, T., Butz, A.: The EnLighTable: Design of Affordances to Support Collaborative Creativity. In *Proc. of Smart Graphics Symposium 2006*, 206-217.

[31] Terrenghi, L., Kirk, D., Sellen, A., Izadi, S. Affordances for Manipulation of Physical versus Digital Media on Interactive Surfaces. In *Proc. of CHI'07*, 1157-1166.

[32] Ullmer, B. and Ishii, H. The metaDESK: models and prototypes for tangible user interfaces. In *Proc. of UIST '97*, 86-96.

[33] Underkoffler, J., Ishii, H., Urp: a luminous Tangible Workbench for Urban Planningand Deisgn. In *Proc. of CHI'99*, 386-393.

[34] Wilson, A. PlayAnywhere: a Compact Interactive Tabletop Projection-vision System. In *Proc. UIST '05*, 83-92.

[35] Wu, M., Balakrishnan, R. Multi-finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays. In *Proc. of UIST '3*, 193-202

TapTap and MagStick: Improving One-Handed Target Acquisition on Small Touch-screens

Anne Roudaut[1] Stéphane Huot[2] Eric Lecolinet[1]

Anne.Roudaut@enst.fr, Stephane.Huot@lri.fr, Eric.Lecolinet@enst.fr

[1]TELECOM ParisTech, CNRS LTCI
46 rue Barrault
75013, Paris, France

[2]LRI – Univ. Paris-Sud & CNRS, INRIA
F-91405 Orsay
France

ABSTRACT

We present the design and evaluation of TapTap and MagStick, two thumb interaction techniques for target acquisition on mobile devices with small touch-screens. These two techniques address all the issues raised by the selection of targets with the thumb on small tactile screens: screen accessibility, visual occlusion and accuracy. A controlled experiment shows that TapTap and MagStick allow the selection of targets in all areas of the screen in a fast and accurate way. They were found to be faster than four previous techniques except *Direct Touch* which, although faster, is too error prone. They also provided the best error rate of all tested techniques. Finally the paper also provides a comprehensive study of various techniques for thumb based touch-screen target selection.

Categories and Subject Descriptors

H.5.2. User Interfaces: Input Devices and Strategies, Interaction Styles, Screen Design; D.2.2 User Interfaces

General Terms

Design, Human Factors

Keywords

Mobile devices, one-handed interaction, thumb interaction, touch-screens, interaction techniques.

1. INTRODUCTION

Many mobile devices are now fitted with touch-screens that enable us to interact directly with our fingers. However, most graphical interfaces still require users to click on small widgets by using a stylus. As highlighted in [7, 10], this interaction style is not the best way to interact with small devices in a mobile context: it requires too much attention (especially if the user is moving) and forces users to use both hands (one hand holding the device while the other manipulates the stylus). Ideally, mobile interaction should just require one hand, with the thumb being used for selecting objects. In fact, direct selection on the screen is intuitive and fast, and using only one hand is central as users may perform several simultaneous tasks.

AVI'08, 28-30 May , 2008, Napoli, Italy

However, direct thumb interaction on small touch-screens raises several issues: a) hand and thumb morphology makes it difficult to reach the corners of the screen; b) the thumb may occlude large parts of the screen that can contain the desired target; c) the relatively large contact zone between the fingertip and the tactile screen makes selection ambiguous, especially in applications that require users to click on tiny widgets for triggering actions. Despite these issues, this *Direct Touch* technique is still the most widely used.

Figure 1. TapTap and MagStick

Alternate techniques have been proposed to improve accuracy and eliminate visual occlusion, but they make the interaction slower and more complex. In this paper, we first present a thorough analysis of the properties of the techniques published so far. We then introduce two novel interaction techniques, **TapTap** and **MagStick** (Fig. 1) that solve the problems raised by our analysis (screen accessibility, visual occlusion and accuracy). Finally, we performed a controlled experiment which proved that our two techniques outperform the ones proposed previously (*Direct Touch*, *Offset Cursor* [11], *Shift* [14] and *Thumbspace* [6]).

2. RELATED WORK

Research on thumb interaction with mobile devices is a relatively recent field. The state of the art thus still largely relies upon research on interaction with regular touch-screens. In spite of recent innovations, the issues of reaching far targets, visual occlusion and accuracy are not yet completely solved.

Target accessibility. The borders of the screen are more difficult to reach [6], especially with the thumb because the morphology of the hand constrains thumb movements. This will degrade interaction in the screen areas that are farthest from the natural thumb extent (*i.e.* the top and left border for a right-handed user). Besides, thumb movements may also be hampered near borders because of the thickness of the device's edges around the screen.

Visual occlusion. When interacting, the finger hides a part of the screen and can even totally occlude small targets. This problem is more pronounced when interacting with one hand because the thumb pivots around the thumb joint and can hide half the screen.

Accuracy. A study [9] showed that 9.2 mm is the minimum size for targets to be easily accessible with the thumb. Some mobile devices, such as the iPhone or the HTC Touch, rely on a limited set of large buttons. But, this approach reduces the number of targets because of the lack of screen estate and is thus inappropriate for many applications. Besides, the exact location of the pointer tends to be imprecise because of the large contact surface between the thumb and the screen. As current touch-screen hardware technology computes the barycenter of the multiple contact points, small variations in the way of pressing the thumb can provoke jerky movements of the pointer.

In the following, we group the existing attempts to solve those issues in three categories depending on how they handle input: "tapping","dragging"and "hybrid" techniques.

2.1 Tapping Techniques

Tapping techniques capture the position of the pointer when the thumb touches the screen. The most widely used technique on regular or small touch-screens, Direct Touch, relies on this intuitive principle. The user must tap the screen precisely at the location where the target is displayed. This technique is fast but it is also very error prone for selecting small targets because, as mentioned previously, the location of the contact point is hard to anticipate. Finally, Direct Touch does not tackle the problem of targets located at the borders of the screen.

2.2 Dragging Techniques

Dragging techniques come from the *take-off* paradigm [11] which consists in pressing the screen, dragging a cursor, and lifting the finger to validate the selection. The former technique, Offset Cursor [11,13], was designed to avoid finger occlusion on large touch-screens and to solve the accuracy problem of Direct Touch. A cursor is always displayed at a fixed distance above the contact point to help the user reaching the topmost locations of the screen. Offset Cursor was shown to induce far fewer errors [11, 13] than Direct Touch, but it is also significantly slower. In [14], Vogel et al. noticed that users often overshoot or undershoot targets. They assumed that it is difficult for the users to estimate the offset distance and that a lengthy adjustment of the cursor, called *net correction distance*, is thus necessary to acquire targets.

Another point is that Offset Cursor does not cover the entire extent of the screen. As the cursor is always located at the same distance from the top of the finger, targets at the bottom of the screen remain unreachable. Moreover, the thickness of screen edges makes it difficult to select targets located near the corners. An adaptative horizontal offset has been proposed in [5] to improve Offset Cursor: this offset is null at the center of the screen and grows smoothly towards the left and right borders. This technique makes it easier to reach items that are close to the left and right borders, but requires slightly more training.

Thumbspace [6] has been designed to improve access to the borders and corners of the screen. It uses an on-demand "radar view" that the user can trigger at the center of the screen. Interacting directly on this radar view allows the user to reach all locations on the screen. Thumbspace thus works as an absolute positioning touchpad superimposed on the standard touch-screen. A drawback of this approach is that the thumb is above the cursor in some areas of the screen, thus causing an occlusion. To get around this issue, the authors proposed to use Thumbspace for targets that are difficult to reach and Direct Touch for near targets.

Thumbspace also relies on *Object Pointing* [3]. The original feature of this interaction technique is that the cursor never visits empty regions and jumps from one target to another, according to the direction of the pointer. Thumbspace uses this strategy with a triggering threshold of 10 pixels to avoid jerky cursor movements. The screen is subdivided into "proxy" areas which are associated to a unique target. This way of "tiling" makes unused background areas active and thus provides more motor space for selecting each target. However, this approach may lose in efficiency when many targets are present on the screen or if they are close to each other.

2.3 Hybrid Techniques

Shift [13] attempts to decrease the selection time of Offset Cursor by a hybrid approach: a coarse Direct Touch on the target can be followed by a precise cursor adjustment if needed. Touching the screen triggers a callout that shows a copy of the occluded area in a non-occluded area. The actual selection point (under the finger) is represented by a cursor in the callout, and the user adjusts its position to fine tune selection before releasing his finger. This technique reduces the *net correction distance* and selection time as the user touches the screen directly on the target. Besides, the callout only appears when needed, after a delay that depends on the target size (the larger the target, the longer the delay). This strategy should improve selection time as the callout is only used for fine-tuning small target selections. However, Shift does not completely solve the screen coverage problem as it requires users to put their fingers close to the target location. Finally, the experiment that was presented in [13] was performed by using both hands to manipulate the device.

2.4 Summary

Direct Touch is the fastest technique proposed so far. However, it remains unusable in most real-life applications because of its high error rate. Some alternatives, inspired by the *take-off* paradigm [11], have been proposed. However, even if they solve the accuracy problem of Direct Touch, the other issues of thumb interaction remain unaddressed. Offset Cursor avoids occlusion and increases accuracy but it limits access to targets at the bottom of the screen and it is not very well suited for reaching targets in the right and left corners (this problem can however be solved by using an adaptive horizontal offset). Thumbspace was specifically designed to address this accessibility problem in the corners, but it does not prevent occlusions in the center of the screen. Finally, Shift which was evaluated by using both hands, does not fully address the corner accessibility issue of the thumb as users must tap close to the desired targets. To sum up, as illustrated in Table 1, efficient solutions have been proposed to solve the problems involved with thumb interaction individually, but none of the existing techniques address them all together. This is the challenge we met by designing TapTap and MagStick, two new interaction techniques that we introduce in the next sections.

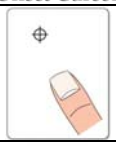
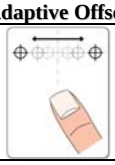
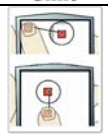
	Direct Touch	Offset Cursor	Adaptive Offset	Thumbspace	Shift	TapTap	MagStick
Overview							
Target Accessibility							
	Grayed areas are difficult to reach – Hatched areas are impossible to reach						
Thumb Occlusion	Everywhere	None	None	Center (if same relative and absolute positions)	On top left	None	None
Pointing Accuracy	Coarse	Medium (net correction distance time)	Medium (net correction distance time)	Fine (facilitated by Object Pointing)	Medium (small targets) and coarse (large targets)	One coarse and one fine (increase target size)	Fine (facilitated by Semantic Pointing)

Table1. Comparison of the features of one-handed interaction techniques

3. TAPTAP AND MAGSTICK

TapTap and MagStick are specifically designed for interacting with the thumb on small touch-screens. Both techniques address the issues of thumb interaction that we previously pointed up. Their respective designs result from a twofold strategy: TapTap was conceived as an improvement of Direct Touch and solves its accuracy and accessibility problems and MagStick is an improvement of Offset Cursor and other techniques based on the take-off principle. A video demonstration can be viewed at http://www.anneroudaut.fr

3.1 TapTap

TapTap comes from a simple idea: if a single tap is not efficient for selecting a small target accurately, a second tap should suffice to disambiguate the selection. More precisely, the first tap defines an area of interest on the screen (Fig. 2a); this area is then magnified and displayed as a popup on the center of the screen (Fig. 2b); the second tap selects the desired target in the popup (Fig. 2c) (or cancels the selection if an empty space is selected).

Selection is by design more precise because the selecting tap takes place on a magnified view of the area of interest where the targets are large enough to be easily selected with the thumb. TapTap also improves accessibility in screen border areas. Not only does the first tap not need to be performed on the desired target (it must only be performed reasonably close to this target), but also the magnified view pops up in the center of the screen. Targets that are close to the borders in the original view thus appear in a location that is much easier to reach in the magnified view.

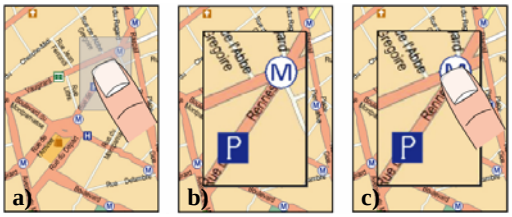


Figure 2. TapTap Design

TapTap is thus based on a temporal multiplexing strategy where the first tap serves to specify the focus area in the original view so that this focus will be displayed at a scale that makes it possible to select the target precisely. Although based on zooming, this strategy has some interesting characteristics that make it different from usual multi-scale approaches. First, there is no interactive control of the zooming factor nor of the amount of XY panning as they are automatically adjusted. Interaction is very fast and works practically like a quasi-mode: the first tap enters the selection mode and makes the zoomed view appear, while the second tap closes this view and leaves the selection mode.

The zooming factor was chosen in order to take into account the size constraints of small touch-screens on mobile devices. Besides, in an attempt to satisfy contradictory constraints, the view and the targets are not zoomed in with the same factor.

On the one hand the focus zone that is selected by the initial tap must be relatively large so that it contains the desired target even if the tap location is (reasonably) far from the target. A size of 80 x 120 pixels was empirically chosen (for a QVGA screen of 240 x 320 pixels). This makes it possible to tap as far as 40 pixels horizontally and 60 pixels vertically from the desired target, a distance that is sufficient to prevent almost all errors in the first tap.

On the other hand, the relatively large size of the focus zone constrains the zooming factor that can be applied in the magnified view because of the small size of the QVGA screen. Moreover, the whole screen real estate can not be used because of the accessibility problem (the areas close to the borders are difficult to reach with the thumb). As a consequence, the focus area is only magnified by a factor of 2 in the pop up (its size is thus 160 x 240 pixels) and placed in the center of the screen (Fig. 2). It is hence located in the most favorable area of the screen for interacting [6].

However, this zooming factor may be insufficient for making common targets large enough to be selected precisely. According to [9] targets should be at least 9,2mm large for making thumb selection easy. But many mobile applications have targets as small as 3 mm [14,12]. In order to ensure sufficient size, targets are zoomed in by a factor of 3 instead of a factor of 2 for the rest of the focus view. Ours observations showed this choice to be effective: users had no difficulties in selecting 9mm targets (i.e. 3mm targets magnified 3 times) and they were not disoriented by this dual zooming factor (in fact none of them noticed this feature).

3.2 MagStick

Dragging techniques are more accurate than Direct Touch but they are significantly slower and do not solve all screen accessibility issue. MagStick solves these problems by providing a telescopic stick that controls a "magnetized" cursor. The telescopic stick can reach any target on the screen while the magnetization of the cursor (which can be seen as form of semantic pointing [8]) speeds up the adjustment of the cursor to the target location. Finally, the offset distance of the cursor is not constant, but dynamically adjusted by the user in a highly predictable way.

MagStick works as follows: 1) when the user presses the screen, he defines a reference point (Fig. 3a); 2) by dragging his thumb he makes a two-part stick appear (Fig. 3b): the two parts emanate from the reference point and end at the current position of the thumb and the location of the cursor; 3) as both parts always have the same length and (initially) the same direction, the user can control the location of the cursor by dragging his thumb continuously on the screen (changing the size of one part of he stick automatically changes the size of its other part); 4) targets *attract* the cursor as if it was "magnetized", with the effect of bending the stick as shown in Fig. 3c); 5) finally the user releases the thumb to select the target that is currently below the cursor (or to cancel if an empty space is selected)

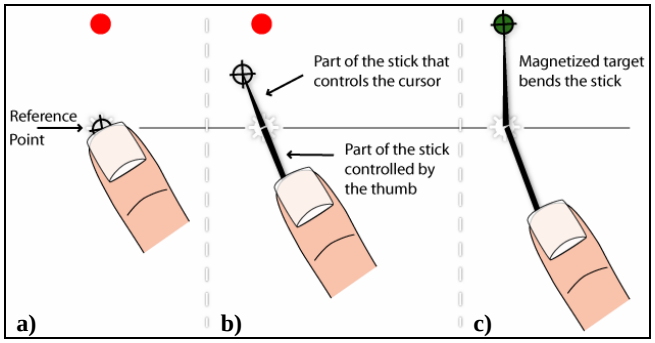


Figure 3. MagStick Design

A key feature of this technique, which was inspired by games such as electronic billiards, is that the cursor moves in the opposite direction of the fingertip. This strategy is especially efficient for avoiding visual occlusions as the thumb must be moved *away* from the desired target: not only the thumb will not hide the target but a large part of its visual context will be made visible.

Another important feature of MagStick is that its symmetrical design allows the user to easily predict the movement to perform. An important drawback of Offset Cursor is that most users, with the exception of very well-trained ones, can not know the exact location of the cursor until they touch the screen. They must thus wait for the cursor to appear before starting to adjust its position finely. Conversely, as the two parts of the stick are of equal length, this problem does not exist with MagStick. The user can predict how far he will have to move his finger *before* touching the screen as this distance is equal to the distance between the target and the reference point.

Magnetization, which derives from Semantic Pointing [2], also contributes to speed up the selection task. Each target has a proximity area that attracts the cursor and "bends" the stick. When the cursor enters a proximity area, it is attracted to the center of the corresponding target. This feature makes fine positioning unnecessary but also avoids "empty selection" errors that would otherwise occur when the user overshoots or undershoots the desired target. Conversely, when the user moves the stick (and the cursor) away from a target area, the magnification effect vanishes and the two parts of the stick become aligned again until the cursor is attracted by another target. A possible refinement would be to assign different attraction powers to targets, as proposed in the original Semantic Pointing technique. It could facilitate the selection of targets that are very frequently used, or, conversely, to prevent the accidental activation of dangerous commands. However, this feature should be carefully tested in the context of thumb interaction where cursor movements are necessarily more imprecise than when using a mouse on a desktop.

4. PROPERTIES OF THE TECHNIQUES

This section compares the properties and the respective advantages of our techniques. In particular, it shows that they provide efficient solutions to the three problems presented in the 'related work' section: target accessibility, visual occlusion and accuracy. We also investigate the compatibility of our techniques with dragging gestures and other target sizes and layouts.

4.1 Target accessibility

TapTap and MagStick can select targets anywhere on the screen although they use different principles. TapTap uses a two-step zooming strategy where the user specifies a focus of interest that is then displayed at a larger scale in the center of the screen. The first tap does not need to be very close to the target and the second tap is always performed in the most favorable area of the screen.

Conversely, MagStick relies on a space-shifting strategy by providing a "telescopic arm" that reaches targets close to the borders. As with TapTap, MagStick makes it possible to perform the dragging gesture in the most favorable area of the screen, but it leaves freedom to the user to interact by following two different strategies. The first one consists in touching the screen very close to the target in order to minimize the length of the dragging gesture. Another strategy is to systematically start the dragging gesture from the center of the screen. Any target can then be selected, either by placing the thumb below the target if it is in the upper part of the screen, or above the target if it is in its lower part. This strategy was in fact used by most of our participants during the evaluations. Another of its advantages is that it allows the user to hold the mobile device firmly with the hand that performs the interaction. The thumb joint is then located in the middle of the right border of the screen (for a right hand user) and the center of gravity of the handheld device is roughly above the center of the hand. This position is safe and convenient because it prevents the risk of dropping the device accidentally. The user then moves his thumb upward or downwards when the target is exactly located beneath the natural position of the thumb joint, but this case seldom occurs and does not require cumbersome hand movements.

4.2 Visual occlusion

The zooming strategy of TapTap prevents visual occlusion by design: as targets are magnified by a x3 ratio, they are large enough not to be completely hidden by the thumb.

The design of MagStick also ensures that visual occlusion can not occur as the thumb moves away from the desired target. Both the target and the focus of attention are clearly visible. It also prevents occlusion in the thumb joint area as shown in Fig. 4 for the same reason as explained in the previous section: the thumb is naturally located in the middle of the screen and can easily be slightly shifted up or down when needed.

Figure 4. No occlusion on the thumb joint with MagStick

4.3 Accuracy without reducing speed

Both techniques attempt to "circumvent" the constraints of the Fitts' Law for a homogeneous 2D space in different manners. TapTap relies on a multi-scale space (that can be seen as a generalization of magnification tools as those proposed in [8,15]). As shown by Guiard et al. [4] multi-scale spaces significantly increase the range of indexes of difficulty that users can handle and Fitts' Law applies uniformly over this range. TapTap makes it possible to decrease the index of difficulty through zooming (that is to say a translation on the scale axis of the space-scale diagram). The two taps required by TapTap are thus performed faster than two "standard" successive taps (this assumption was confirmed by experimental data): The size of the "target" is increased, and the distance between the thumb and the target decreased in both steps of the interaction (the "target" being a zone of interest in the first case, and an actual but magnified and centered target in the second case). This property also increases accuracy and allows the user to view TapTap as a double click with a fast spatial readjustment between the two taps. As detailed in the experiment section, this effect was striking when conducting the evaluation: users did not give the impression that they were performing two successive taps but rather a compound gesture.

Similarly, MagStick relies on Semantic Pointing, a technique that distorts the motor space and thus artificially reduces the pointing distance. This technique also avoids the cursor leaving the target when the thumb is slightly, and involuntarily, moved. As stated above, the input signal provided by current touch-screen technology is somewhat imprecise and instable when interacting with the thumb. Although filtered by a low pass filter to remove outliers and smooth the input curve [13,14] this signal is still far from being perfectly reliable. Besides, the user may also involuntarily move his thumb when he releases it and thus miss the target. Magnetization solves both problems.

Finally, the ability to predict the movement before starting the gesture is probably another key feature for making the selection faster. The property relies on the fact that both parts of the stick always have the same length. Using a variable gain, as in [1], sounds appealing but could decrease performance in our case because this important property would be lost. This was confirmed by preliminary experiments we made when designing MagStick.

4.4 Other properties

Real mobile computer operations are combination of different interaction techniques, such as pointing or dragging. In our experiments, we focus on pointing with small and randomly laid out targets. In this section, we present some other interesting properties of TapTap and MagStick. More precisely, we investigate how our techniques work with different kinds of targets (size and layout) and their compatibility with other interaction styles.

Large targets. Although targets can hardly be much smaller, and still easily visible, than those we considered (3mm, a size found in many mobile applications [14,12]), they can however be much larger. MagStick then operates as Direct Touch: as the cursor appears below the thumb when it is pressed on the screen, the user can just release it without performing any movement to select the target. TapTap can be replaced by Direct Touch for large targets. This can be made explicit by a visual cue. But choosing target sizes in a consistent way may suffice (for instance targets with only 2 or 3 different heights). Selecting targets in two different ways may not be a real problem after some training: a) people do that all the time when using desktops (documents must be double-clicked, while other buttons are, generally, simple-clicked); b) a small inactivation delay could be used in such a way that a second click on a large button (or the view it generates) would have no effect. Hence, a useless second tap would never produce an unexpected result.

Lists and Groups. Aligned or grouped targets are often common in real applications: this case typically occurs in menus, lists, tool boxes, tabbed panes, etc. While TapTap performance is likely to be similarly high whatever the layout, the specific design of MagStick can provide interesting features in this case. It makes it for instance possible to access items organized as lists or trees by moving the thumb away and keeping it approximately at the same location of the screen. This could be very useful for browsing a menu system without having to perform multiple target selections. Besides, as the thumb can be placed rather far away from the target, this would noticeably reduce occlusion and would thus make it possible to display more contextual information.

Dragging gestures. TapTap does not interfere with interaction styles based on dragging gestures as it only requires users to tap the screen. A target can be moved by dragging the thumb on the screen instead of releasing it immediately after the second tap (the popup does not cause visual occlusion because it disappears when the user starts the second tap by pressing the screen). This way of dragging objects is in fact quite similar to the usual one except that the target is not beneath the cursor but remotely controlled by the movements of the thumb. The target moves in the global view according to the movement of the thumb from the position of the second tap. In order to move the target anywhere on the screen, this movement is multiplied by a constant gain of 2. In addition, TapTap also makes it possible to pan the entire view by dragging on its "background". An image, a map or a page could for instance be panned in this way.

MagStick also has interesting properties regarding this criterion. First, it allows an object to be dragged, although in a slightly less usual way than with TapTap. Instead of releasing the thumb immediately when the proper target is reached, the user must wait for a small temporal delay. The target is then implicitly selected and can be moved by dragging the finger.

To sum up, we have seen in this section that TapTap and MagStick address all the issues raise by one-handed interaction, and that they can be applied in different kinds of application without preventing the use of other interaction styles. The next section shows the effectiveness of TapTap and MagStick through a controlled experiment that compares them with the main techniques proposed so far.

5. EXPERIMENTAL EVALUATION

We conducted a controlled experiment to compare TapTap and MagStick with the main techniques published before: Direct Touch, Offset Cursor [11], Thumbspace [6] and Shift [14]. Since the previous techniques principally explored the pointing task, our experimentation focuses on this problem of pointing only. According to the design of our techniques and the properties that were previously described, our hypotheses are that:

***H1**: TapTap and MagStick are the fastest techniques after Direct Touch.*

***H2**: TapTap and MagStick are the techniques with the lowest error rate*

***H3**: TapTap and MagStick are efficient for accessing targets anywhere on the screen.*

5.1 Task

The task consisted in performing series of target selections with the six techniques. Participants were asked to hold the device with their dominant hand and to use their thumb. Several targets were displayed on the screen and one of them was to be selected. The participants were instructed to perform the selection as quickly and accurately as possible. Before each trial, the user presses a "Next trial" button and a city map appears with a set of 16 targets. They are displayed in blue color, except for the one to be selected that is in red. The blue targets are distractors in order to improve the realism of the target acquisition task. The color of the target changes to green when the cursor flies over it (except for tapping techniques such as Direct Touch and TapTap). The trial ends when the user lifts his thumb from the screen, whether he succeeds or not the selection. A sound indicates the result of the acquisition.

5.2 Apparatus and participants

The techniques have been implemented in C# (with the .Net Compact Framework) and operate on the Windows Mobile 5.0 OS. Experiments have been performed on a HTC P3600 PDA-phone with a QVGA (320x240) touch-screen. Twelve volunteers (1 female), ranging in age from 23 to 47 years, were recruited from our institution and received a handful of candies for their participation. All of them were using a mobile device with a touch-screen for the first time. Two subjects were left-handed and we mirrored their results so that each user used their dominant hand to perform the experimentation.

5.3 Experimental conditions

The efficiency of the interaction techniques involved in this experimentation is likely to depend on the location of the target. Karlson et al. took this aspect into account in their experiment [6]. They subdivided the screen into 12 areas arranged as a regular matrix. We used a different subdivision pattern, with 8 zones of the same surface area (Fig. 5a represents the 12 areas for a right-handed person), which provides a clear separation between the areas located at the center of the screen and those close to the borders, which may degrade performance. This analysis of the screen areas is important because it can have strong implications on the design of interactive applications.

To reduce the task time for our participants, we only considered one target size of 3 mm, because this value was reported to be the actual minimal widget size in mobile applications [14,12]. Besides, Vogel also reported in [14] that Direct Touch and Shift outperforms other techniques for targets larger than 18 pixels. The study thus focuses on small targets, as they constitute a more difficult case and are commonly found in mobile applications. The proximity areas for the MagStick magnetize effect measure 10.8mm.

A minor enhancement was made on Offset Cursor because its original design makes it impossible to reach targets on the bottom of the screen. So that this technique is not at disadvantage, the user can make the cursor appear below the thumb position (negative offset mode) by pressing a hardware button before touching the screen. The analysis of the experimental data confirmed that this improvement did not affect the results (the performance is not significantly different in the 'down' area than in "easy to reach" areas such as 'up' and 'Center'). Hence all targets can be selected by using any of the 6 tested techniques.

During the task, Time, errors and thumb movements were recorded. At the end, the subjects answered a questionnaire to give their opinion and satisfaction about all techniques (6 variables were measured on a 5 pt. Likert scale).

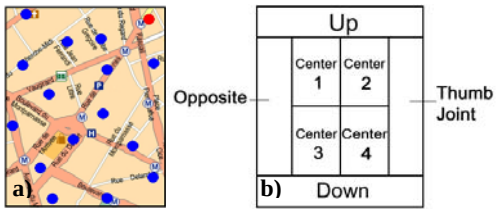


Figure 5. a) Targets layout b) Target Area subdivision

5.4 Experimental design

A repeated measures within-subject design was conducted. The independent variables are *Techniques* (Direct Touch, Offset Cursor, Thumbspace, Shift, TapTap and MagStick) and *Target Area* (8 areas shown in Fig. 4a). The presentation of *Technique* was circularly counterbalanced among participants. All of them performed 16 selections twice in all the 8 *Target Areas*. *Target Areas* were ordered in a sequence circularly counterbalanced for each technique. This sequence aims at balancing the regions that are easy or hard to reach. Finally, at the beginning of each technique, subjects performed 10 practicing trials. In summary, the design was: 6 *Techniques* x 8 *Target Areas* x 2 *blocks* = 96 selections (15-20 minutes) per participant.

5.5 Results

Repeated measures analysis of variance showed that the order of presentation of the techniques had no significant effect on selection time or error rate.

5.5.1 Selection time

Task time was measured from the moment the user released the "Next trial" button to the moment his thumb was lifted up from the screen. Trials with selection errors were excluded from the selection time analysis. We performed a 6 x 8 (*Technique*, *Target Area*) within subject analysis of variance. We found significant main effects for *Technique* ($F_{5,55}$=14.59, p<.001) and *Technique* x *Target Area* interaction ($F_{35,268}$=2.31, p<.001). Post hoc multiple means comparison tests allowed us to rank the techniques as follows: Direct Touch (1177.8 ms) and TapTap (1547.4 ms) (no significant results between them), MagStick (2037.6 ms), Shift (3046 ms) and Offset Cursor (3562.7 ms) (no significant results between them), and the slowest, Thumbspace (3897.3 ms). The results show that: TapTap is about to 2.3 times faster than Offset Cursor, 2 times faster than Shift, and almost 2.5 times faster than Thumbspace; MagStick is about 1.7 times faster than Offset Cursor, 1.5 faster than Shift and 1.9 faster than Thumbspace. These results, illustrated in Fig. 6, are all significant. We found that Direct Touch was the fastest, but (as described in the error result) the quantity of data collected is small compared to the other techniques. We can considerate Direct Touch "out of range" because a technique that produces so many errors is of course very frustrating for users, and can not be compared in this experiment with the other techniques that all provide better results.

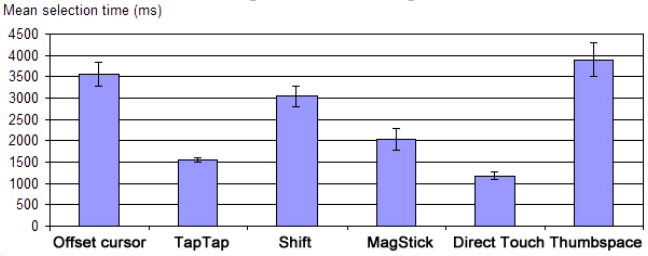


Figure 6. Mean time (ms) for Technique
Bars represent 95% confidence interval.

The analysis of the *Technique* x *Target Area* interaction showed that *Target Area* has no significant effect on selection with TapTap and MagStick. There is a significant effect for Offset Cursor, which is less efficient in the 'joint' area (see Fig.4) (2246.2 ms mean difference) and in the 'opposite' area (1250 ms mean difference) than in other zones. A similar effect was found for Shift in the 'up' (2445.6 ms mean difference) and 'opposite' (936.8 ms mean difference) areas. These results confirm our observations during the experimentation sessions where we noticed that users often hide the target with their thumb in these two areas. Some other significant effects were also found with Thumbspace, which performed better on the borders of the screen than in the center area (1381.9 ms of difference on average). This result corroborates the assumptions of the authors [6].

In summary, without considering Direct Touch, TapTap is the fastest and MagStick the second. The border areas are reached faster with MagStick than with the hybrid and the dragging techniques. Not only is MagStick quite efficient for reaching the edges, but it also does not impair interaction in the center of the screen (as Thumbspace does). TapTap is particularly fast and consistent across screen areas.

5.5.2 Error rate

The error rate measurement aggregates both empty and wrong target selections. We performed a 6 x 8 (*Technique*, *Target Area*) within subject analysis of variance on the aggregated number of errors. Error rate was significantly affected by *Technique* ($F_{5,55}$=45.91, p <.001) and *Technique* x *Target Area* interaction ($F_{35,268}$ = 1.74, p<.001). Post hoc multiple means comparison tests showed that TapTap (6.7%) has the lowest error rate and Direct Touch (59.9%) the highest in comparison to all other techniques (Fig. 6). No significant results were found in comparing the other techniques (i.e. Offset Cursor (16.1%), Shift (17.1), MagStick (10.4%) and Thumbspace (18.7%)). We can notice that the error rate of Direct Touch is considerably high. The error rate of TapTap is about 2.5 (and 1.6 for MagStick) times smaller than for Offset Cursor, Shift and Thumbspace.

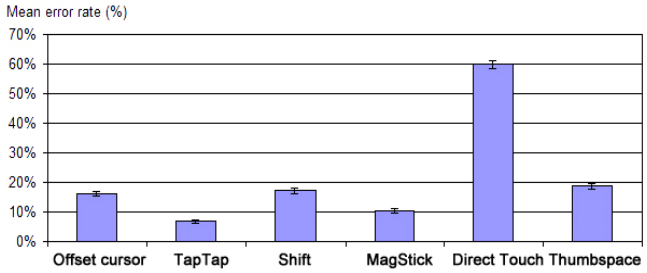


Figure 7. Mean Error rate for *Technique*
Bars represent 95% confidence interval.

The only significant result about *Technique* x *Target Area* interaction is mainly due to Direct Touch. Considering its high error rate and dissatisfaction of our participants with it, we will not discuss on these results. By considering empty and wrong selections separately (they were previously merged), we found that Thumbspace only produces wrong selections while the other techniques induce mostly empty selections. In fact these results are not surprising because by design Thumbspace "tiles" the space. This approach, which could be efficient because the target is then larger in the motor space, have also the disadvantage of causing more wrong errors that are much more costly than empty selections (canceling an action triggered by a wrong selection may be time-consuming and frustrating).

In summary, TapTap has the lowest error rate and Direct Touch the highest. All the "dragging" techniques and Shift have approximately the same error rates, except that Thumbspace errors are only wrong selections.

5.5.3 Subjective preferences

With the post-study questionnaire, participants ranked the six techniques as follows: TapTap, MagStick, Shift, Offset Cursor, Thumbspace and the most disliked Direct Touch. Their opinions about the *speed*, *accuracy*, *pleasantness*, *simplicity*, *learning* and *fun* are illustrated in Fig. 8. TapTap is the most liked technique for all criterions, except for 'fun' where it is placed second. Tapping approaches (TapTap and Direct Touch) are ranked first for the 'speed' assessment and users estimated that TapTap performs faster than Direct Touch, even if quantitative results showed the contrary. Direct Touch is disliked for the 'accuracy', 'pleasantness' and 'fun' criteria. Results for dragging approaches have a similar shape, with MagStick and Offset Cursor generally above Shift and Thumbspace. MagStick is judged slightly inferior for 'learnability' but ranked first for 'fun'.

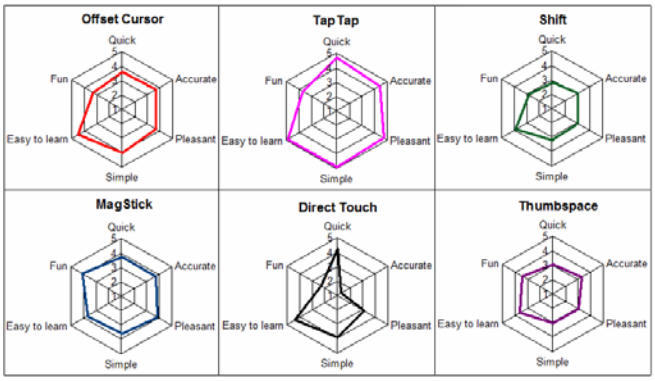


Figure 8. Questionnaire results (means).

5.6 Discussion

The results of our experimentation confirm our hypotheses. TapTap has the lowest error rate (H2) of all techniques and it is the fastest technique after Direct Touch (H1). In fact, it would be even faster than Direct Touch in the case of real usage. As Direct Touch is very error prone, many selections will have to be performed again. The average time needed to select a target is thus significantly higher than the time to correct selections given in the previous section. This average time can be estimated by considering that the selection task will take at least twice as much time in the case of wrong selections as the target must then be selected again (in fact it will take more time because a wrong selection may launch an undesired application that the user will have to close). According to this hypothesis, Direct Touch would require an estimated average time of 2002 ms while TapTap would only need 1676 ms as it produces much fewer errors.

Another interesting point is that the single tap of Direct Touch takes more time (1177.8 ms) than each tap of TapTap (803.3 ms for the first tap and 744.1 ms for the second tap). These results confirm the validity of the design hypotheses presented in section 3. Besides, users seem to perform the second tap slightly faster, an effect that may come from the fact that the magnified area is centred and the target thus pretty close to the natural position of the thumb.

Our results also validate the hypotheses that MagStick is faster (H1) than other techniques (TapTap and Direct Touch except, but Direct Touch is too error prone to be really usable, as stated before) and that it produces fewer errors (H2) than other

techniques (TapTap except again). Another interesting observation is that the time to press the screen is slightly faster for MagStick (844.2 ms) than for other dragging techniques (1140,6 ms for Offset Cursor, 958,7 ms for Shift and 935,8 ms for Thumbspace). This may be explained by the fact that users tend to place their thumb systematically in the centre of the screen without spending time to adjust the position of the thumb. Once they touch the screen (approximately) at its center they then move the thumb for the same distance as the distance between the center and the target. As the execution time of MagStick is also faster than for other dragging techniques, we hypothesize that the users make an estimation of this distance before touching the screen (only one participant among the twelve has made errors due to a wrong positioning of the thumb).

The rapidity of MagStick may also be explained by Semantic pointing. However this mechanism depends on target density, and should be carefully tested in this context. Our first experiments with a high target density (32 instead of 16 targets of 3mm randomly displayed), shows that MagStick performance is then equivalent to those of Shift and Offset Cursor (while TapTap efficiency is almost the same for both densities). To increase the performance of MagStick, we plan to implement a density-dependent approach that dynamically adapts the strength of the magnetizing effect according to the position of the cursor and its local context on the screen.

Our experimentation also shows that the selection time and the number of errors do not depend on screen areas when using TapTap and MagStick (H3). Conversely, Thumbspace is less efficient in central areas (as also demonstrated in [6]), Shift impairs interaction in the top and left corners because of visual occlusion, and Offset Cursor degrades performance in all screen corners. TapTap and MagStick both provide efficient solutions to these issues as they help users to reach any target in a short and constant time, whatever its location on the screen. MagStick performed well in border areas without decreasing efficiency in the center. Finally, MagStick tends to concentrate most thumb movements in the center as shown in Fig. 9. It also provides a comfortable grip for user interaction in mobility conditions and it is well-adapted to thumb morphologic capabilities.

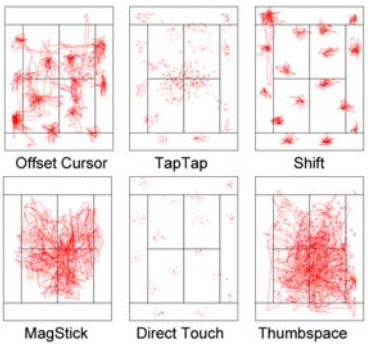


Figure 9. Thumb gesture traces.

6. CONCLUSION AND FUTURE WORK

We have presented TapTap and MagStick, two new interaction techniques that improve target acquisition on small touch-screens for mobile devices. TapTap is based on time-multiplexing through an automatic two-step zooming strategy. MagStick relies on magnetization, a variant of semantic zooming and also makes it possible to predict thumb movements and thus to reduce the net correction distance. Our experiments showed that both techniques are faster and produce fewer errors than the current state of the art. They also cover the other issues raised by thumb interaction on small touch-screens such as visual occlusion and target accessibility in all parts of the screen. They are also both compatible with interaction techniques relying on dragging gestures. Finally, this paper also offers a significant benefit by presenting a thorough analysis of the techniques published so far.

In future work, we plan to adapt our techniques to constraints that depend on the application context (higher target densities, specific target layouts such as lists or trees…) and to perform further evaluations to evaluate their efficiency under these conditions.

7. ACKNOWLEDGMENTS

This work has been done in collaboration with Bruno Aidant, Bruno Legat and Johann Daigremont from Alcatel-Lucent that we thank for their useful advices. We also thank Yves Guiard for his precious help for statistical analysis and all the participants for their pleasant contributions.

8. REFERENCES

1 Albinsson, P. and Zhai, S. 2003. High precision touch screen interaction. *Proc. CHI'03*. 105-112. 2003.

2 Blanch, R., Guiard, Y., Beaudouin-Lafon, M. Semantic pointing: improving target acquisition with control-display ratio adaptation. *Proc. CHI'04*. 519-526. 2004.

3 Guiard, Y., Blanch, R., Beaudouin-Lafon, M. Object pointing: a complement to bitmap pointing in GUIs. *Proc. Graphics interface 2004*.Vol. 62. 9-16. 2004.

4 Guiard, Y., Beaudouin-Lafon, M. (2004). Target Acquisition in Multi-Scale Electronic Worlds. International Journal of Human-Computer Studies, 61, 875-905.

5 Huot, S., Lecolinet, E. Focus+Context Visualization Techniques for Displaying Large Lists with Multiple Points of Interest on Small Tactile Screens. *Proc. Interact'07*.

6 Karlson, A., Bederson, B. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. *Proc. Interact'07*.324-338.2007.

7 Karlson, A., Bederson, B., Contreras-Vidal, J. Understanding on User Interface Design and Evaluation for Mobile Technology, Idea Group, 2007.

8 Mankoff, J., Hudson, S. E., and Abowd, G. Interaction techniques for ambiguity resolution in recognition-based interfaces. *Proc. UIST'00.* 11-20. 2000.

9 Parhi, P., Karlson, A., Bederson, B. Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices. *Proc. MobileHCI'06*. 203-210. 2006.

10 Pascoe, J., Ryan, N., Morse, D. Using while moving: HCI issues in fieldwork environments. *ACM Trans. Comput.-Hum. Interact.* 7(3):417-437. 2000.

11 Potter, R., Weldon, L., Shneiderman, B. Improving the Accuracy of Touchscreens: An Experimental Evaluation of Three Strategies. *Proc. CHI'88*. 27-32. 1988.

12 Ren, X. and Moriya, S. Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks. *ACM TOCHI*. 7(3).384-416. 2000.

13 Sears, A., Shneiderman, B. High precision touchscreens: design strategies and comparisons with a mouse. *Int. J. Man-Mach. Stud.* 34(4):593-613. 1991.

14 Vogel, D. and Baudisch, P. 2007. Shift: a technique for operating pen-based interfaces using touch. *Proc. CHI'07*. 657-666. 2007.

15 Grossman, T., Balakrishnan, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *Proc.CHI'05*. 281-290. 2005.

Combining and Measuring the Benefits of Bimanual Pen and Direct-Touch Interaction on Horizontal Interfaces

Peter Brandl[1,2], Clifton Forlines[1,3], Daniel Wigdor[1,3], Michael Haller[2], Chia Shen[1]

[1]Mitsubishi Electric Research Labs
Cambridge, Massachusetts, USA
forlines | shen@merl.com

[2]Upper Austria University of
Applied Sciences
Hagenberg, Austria
firstname.lastname@fh-hagenberg.at

[3]University of Toronto
Toronto, Ontario, Canada
dwigdor@dgp.toronto.edu

ABSTRACT
Many research projects have demonstrated the benefits of bimanual interaction for a variety of tasks. When choosing bimanual input, system designers must select the input device that each hand will control. In this paper, we argue for the use of *pen and touch* two-handed input, and describe an experiment in which users were faster and committed fewer errors using pen and touch input in comparison to using either *touch and touch* or *pen and pen* input while performing a representative bimanual task. We present design principles and an application in which we applied our design rationale toward the creation of a learnable set of bimanual, pen and touch input commands.

Author Keywords
Bimanual input, pen and touch, self revealing gestures.

ACM Classification Keywords
H.5.2 INFORMATION INTERFACES AND PRESENTATION (e.g., HCI): User Interfaces - Input devices and strategies (e.g., mouse, touchscreen)

1. INTRODUCTION
The benefits of bimanual interaction have been investigated in numerous research projects [4, 7, 8, 9, 14. 32]. By leveraging input from both hands, system designers can increase the input bandwidth from their users and add rich and natural interactions to their applications. When designing for bimanual input, system designers must choose among the many input devices available for each hand. Comparisons among various input devices (such a mice, pucks, stylus, and touch-tables) are plentiful [16, 18, 21, 30]. Taken as a whole, this body of research indicates that individual input devices excel in certain measures and lack in others.

For example, multi-touch interactive surfaces [13, 28, 32] have the strong advantage that no intermediary input device is required. For this reason, this type of direct, "under-the-finger" input device is often called "natural" and "intuitive" when compared to a mouse or stylus. Additionally, by sensing multiple points of contact, these devices allow for complex input [32, 36]. On the other hand, occlusion and finger size hamper accurate touch input in a graphical interface. In contrast, a computer stylus provides a higher-level of input accuracy, but typically only a single point of input. While choosing any *one* particular input device requires weighing these types of tradeoffs, bimanual input allows us to design input commands using different input devices with the dominant and non-dominant hands. Previous work in this area, along with our observations and experimentation, has convinced us that by combining dominant-hand pen input with non-dominant hand touch input, we can effectively harness the benefits of both pen and touch input while avoiding many of their pitfalls.

In this paper, we first survey the related work in the field of bimanual interaction, and then describe a set of design principles based on this work. We further describe our graphical editing application (cf. Figure 1). This application was developed based on the principles described in past work, as well as leveraging the strengths of both pen and touch input. Finally, we present the results of a laboratory experiment, in which the combination of pen and touch input outperforms bimanual pen and bimanual touch input for a representative task.

Figure 1. A designer sketching using our prototype.

2. RELATED WORK
Bimanual interaction has been investigated in numerous research projects and formal studies using a variety of input devices [1, 3, 20, 21, 24, 31]. The previous work on bimanual input can generally been divided into two categories: the first category defines or extends models and frameworks for bimanual input, and the second one applies those models and frameworks.

2.1 Models and Frameworks
Most work in bimanual interaction has been influenced by Guiard's Kinematic Chain model [17], which proposes general principles for asymmetric bimanual actions. During two-handed interaction, both hands have different roles that depend on each other with respect to three rules: the dominant hand (*DH*) moves within the frame of reference defined by the non-dominant hand (*NDH*); the sequence of motion generally sees the NDH setting the reference frame prior to actions with the DH being made within that context; and that the DH works at a higher level of precision than the NDH in both spatial and temporal terms. Follow-up research has extensively investigated different facets of these hypotheses, such as the importance of visual and kinesthetic feedback for bimanual tasks [1, 3] and differences between symmetric [24, 31] and asymmetric [10, 11, 18] bimanual input.

AVI'08, 28-30 May , 2008, Napoli, Italy

Kabbash et al. [21] studied four techniques for performing a compound drawing and color selection task using a unimanual technique, a bimanual technique in which each hand controlled independent tasks, and two bimanual tasks where the DH depended on the NDH. They suggest that asymmetric, dependent tasks are most effectively performed using two hands. Several research projects have sought to apply these findings and to investigate interaction design and input devices for bimanual tasks.

2.2 Interaction Design and Input Devices

Different input devices have been evaluated regarding their suitability for bimanual tasks. The use of bare hands for gestures and self revealing tasks has been studied by Kruger [22] in VIDEOPLACE. Matsushita et al's. Holowall includes bimanual object manipulation [28]. SmartSkin [32] is an interactive surface enabling bimanual interaction for different tasks, such as map panning and zooming. Several projects have explored the benefits of two-handed control [7,14,16,36] using DiamondTouch [13].

Kabbash et al. [21] performed a comparison among a mouse, a trackball, and a stylus for bimanual tasks. Their findings support Guiard's claim that the NDH is best suited for imprecise tasks. Forlines et al. [16] conducted a study that compared bimanual mouse and touch input on interactive tabletops. A combination of a PDA in the NDH and a mouse in the DH was investigated by Myers et al. [30].

Matsushita [27] and Yee at al. [37] implemented mobile devices supporting pen and touch input. In both cases, touch input complemented pen input. Cutler et al. investigated the use of a glove and pen on the Responsive Workbench [12]. They found that the combination of a glove for the NDH and a stylus in the DH worked best for asymmetric tasks that reflected the natural qualities of each input device. The stylus with its thinner tip and more precise point of touch fits better for high precision tasks.

Researchers have explored bimanual interaction for a variety of tasks that can be performed more efficiently compared to a sequential single-handed input. Potential tasks include menu control [4, 6, 23, 35,], desktop interaction such as selecting [8], scrolling [9, 26] and cursor control [5, 14], map navigation [18, 32, 33, 34] and sketching [7, 15, 23, 37].

Hinckley et al. [18] explored the performance of puck and stylus as well as touchpad and *TouchMouse* combinations for bimanual interaction. They found bimanual benefits for map navigation tasks. Formerly sequential actions were chunked by the simultaneous use of two devices were therefore performed more quickly. Kurtenbach et al. [23] tested two-handed interaction with Toolglass menus with a graphics editing program. They used WACOM tablets with two pucks as input devices to evaluate their design approach that aims at maximizing the screen space for application data while providing an increasing quality of input.

The contribution of our work is to leverage insights of previous bimanual-input related research in developing a framework for system designers. This framework is intended to guide the development of interaction using combinations of pen and direct touch input for tabletop interactions. Compared to relatively small surfaces such as Tablet PCs or graphic tablets, the necessity of visually linking the tasks of both hands on a tabletop becomes increasingly important. Switching the attention between left and right hands results in highly sequential performance and neutralizes or reverses the advantage of bimanual interaction [3, 21], a design based on both descriptive principles and predictive models is especially demanded for large surfaces.

3. DESIGN PRINCIPLES

We sought to establish a set of design principles intended to guide developers of bimanual user interfaces. This set is based on an exploration of the possible combinations of bimanual input when each hand may used for either pen or touch input, aided by previous work in bimanual systems. By investigating the possible pairs of input actions, one performed by the DH and one by the NDH, as well as the type of and order in which these commands are performed, we have developed a set of principles that we believe combine the best qualities of pen input and touch input into a single system.

3.1 Input Choices for Both Hands

The general structure we propose for the categorization of input variations is shown in Figure 2. Both, the DH and the NDH have the same set of input possibilities – pen input, touch input, or no input. When using a pen for input (Figure 2, *left branch of tree*), researchers and system designers typically distinguish between *inking* and *command stroking* modes: referring to the usage of a pen for writing or drawing, and for making commands in the second. A barrel-button can be used to delimit these two modes, as can gestural delimiters [18]. Command strokes are interpreted as either point-based interaction (i.e. mouse-like, point-and-click commands) or gestural strokes (e.g. handwriting input).

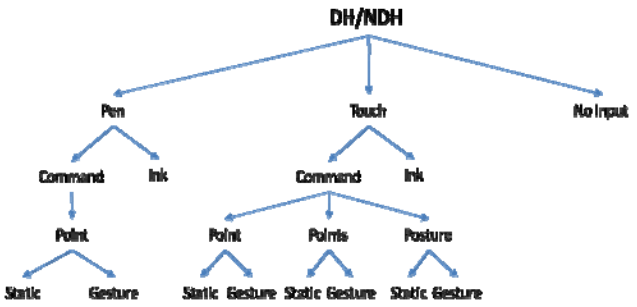


Figure 2. Input categorization. Both the DH and NDH can perform one type of input in a bimanual action.

When using touch input (Figure 2, *center branch*), single-finger commands are often interpreted as point-based interaction (i.e. mouse-like interaction). A benefit of touch input with multi-touch devices is the ability to sense and handle multiple points of input, or even different hand postures. Postures can be recognized as commands themselves or moved over time creating high-bandwidth gestures.

For designing bimanual commands, two different inputs are combined to infer a task or operation. Input possibilities are stated in the tree as leaves. One input comes from the dominant and one from the non-dominant hand.

3.2 Pros and Cons of Pen and Touch Input

We considered the pros and cons of each technique to motivate the assignment of different input combinations for different tasks (cf. Table 1**Errore. L'origine riferimento non è stata trovata.**). We wished to combine the positive qualities of both input mechanisms, while avoiding their pitfalls. According to previous studies on the role of the dominant and non-dominant hands [12, 21], we propose using a pen for precise input with the DH and direct touch for intuitive, high-bandwidth touch commands with the NDH. For example, inking is a typical pen task, as touch input suffers from larger occlusions and lower touch precision. Cutler showed that this use of pen and touch performed better than two-handed touch [12].

They found that, especially for asymmetric tasks, the benefit of a distinguished pen point for fine grain gestures and the intuitive use of coarse hand gestures exactly mirrored the asymmetric distribution of labor described by Guiard.

	PROS	CONS
PEN INPUT	Less accidental input than touch	Only one input point
	Precise touch point / High sensing resolution	Separate device
	Familiar tool that leverages users' experience	Less occlusion emphasizes parallax
	Less occlusion of targets than touch	
TOUCH INPUT	Multiple points of input, high # of degrees of freedom (high bandwidth)	Occlusion by hands and fingers
	Use with low attention	Low touch point preciseness (fat fingers)
	"Natural"	
	No extra input device to manage	

Table 1. Advantages and disadvantages of pen and touch as input devices for tabletops.

3.3 Sequencing of Commands

We have already discussed how each hand can issue one of several input commands and that the combination of direct touch and pen input offers the tantalizing opportunity to take advantage of the strengths of each device. Additionally, we explored the sequencing of the DH and NDH actions. A bimanual task can be started by either hand, and the sequence of the start of the paired input streams can set the context for the further action. Wu et al. [36] refer to this concept as the gesture registration phase that defines the beginning of every gesture operation and therefore sets the context for subsequent interactions. They describe a system in which a stylus can be either treated as a writing device or a pointer depending on the mode set in the gesture registration phase.

When the sequence of bimanual actions is relevant, a pen's input preceding a touch gesture is different to a touch gesture performed prior to a pen point. According to their temporal occurrence, we distinguish three different types of sequences: sequential, overlapping and simultaneous (cf. Figure 3).

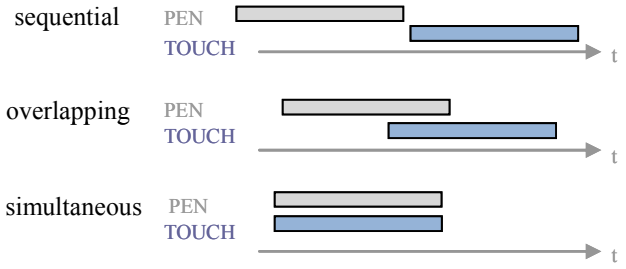


Figure 3. Three causal sequencing of commands.

3.4 Coupling and Decoupling of Interactions

In terms of sequencing, we dynamically and systematically couple and decouple the input of the two hands. For example, a task that can be performed with the NDH could be supported by the DH to extend the functionality or increase the accuracy. The NDH therefore sets the modal reference frame in which the DH will be acting. We add the input of the DH if necessary (couple) and proceed with single NDH input if this is sufficient (decouple). Coarse positioning of an object, for example, can be achieved with the NDH; for a final accurate placement, the DH can be coupled to add high precision information.

3.5 Self Revealing Gestures

Effective feedback is critical in a system that accepts bimanual input consisting of points, postures and gestures. We propose a new type of tooltip that is based on the two level concept of invoking mechanism and consequence. There are different ways of invoking an action: through pen input or with direct touch that allows additional gestures. The user sees these possibilities in the first row of the feedback panel. In the second row, he sees the consequence of each action. Thus he always knows what he can do as a next step and what the consequence will be. We use additional tooltips that are placed near an interactive item in the scene, a button for example. The same tooltip is also shown in the feedback panel. The tooltip in the scene gives additional information about the position where the action shown in the feedback panel has to be invoked.

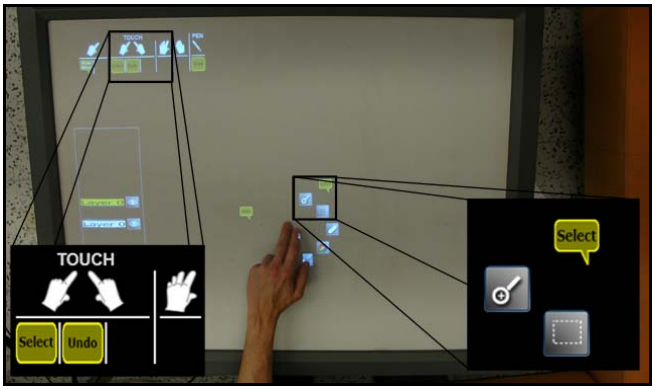


Figure 4. Always visible user feedback shows possible options in each state (left). Tooltips in the scene show positions for command invocation (right).

Our implementation of the mechanism and consequence tooltip concept with position relevant information is depicted in Figure 4. The always visible information panel on top of the screen (cf. Figure 4, *left*) reveals possible actions as a combination of invoking mechanism and consequence. If applicable, balloon shaped tooltips in the scene show the positions for specific commands (cf. Figure 4, *right*). Continuous actions that are currently performed are shown as balloon tooltips with inverted text color in the feedback panel. Matching balloons in the scene will then also show an inverted text.

4. PROTOTYPE APPLICATION

We built a proof-of-concept prototype application, based on our design principles, that wraps Adobe Photoshop, a popular image editing application. Commands are issued through combinations of pen-and-touch input on a digital tabletop. We note that the goal of this prototype is to prove our assumptions in a real world scenario, whereas the concepts address a broader field and could be as well applied to different areas.

4.1 Enabling Technology

To enable direct touch and pen interaction we used a DiamondTouch [13] table to sense touch interaction and Anoto [1] technology for pen input. To achieve a co-incident touch and pen sensing surface, we augmented a DiamondTouch table with a transparent sheet on which we printed an Anoto dot pattern (cf. Figure 5). For our prototype setup, we used multiple Bluetooth streaming pens from Maxell [29]. With this combination of input devices, both the touch-table and the pens provided unique IDs.

The prototype is implemented in C# using the Windows Presentation Foundation (WPF) graphics engine. Our application renders a transparent layer on top of Adobe Photoshop CS3 which we basically use for handling images and sketching tasks. The communication between our application and Photoshop is accomplished through a combination of .NET Automation functions, mouse emulation and generated keyboard shortcuts that invoke the appropriate commands in Photoshop. We hide all Photoshop graphical user interfaces and show how a subset of them can be substituted with our bimanual interaction techniques.

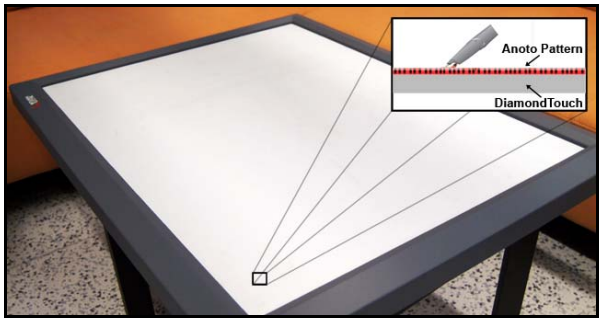


Figure 5. Direct touch and pen tracking surface as a combination of Anoto and DiamondTouch technology.

4.2 Application

Our prototype application shows selected tools that are taking advantage of the strengths of the pen and touch combination and their different roles in bimanual tasks. The functionality covers basic drawing commands in graphics application, like sketching, color picking, brush sizes and eraser tools. Beyond that, further interaction includes scene management like zoom and pan, different kinds of selections and a history tool.

4.2.1 Sketching

In sketching, the precise pen is used as a DH drawing tool while the NDH is free to manage the drawing area through natural panning and zooming touch interaction (cf. Figure 6, *left*). We implemented the zooming feature according to the two point stretch and squeeze technique seen in [19]: touching the surface with two fingers enters the zooming state; the distance between the two fingers defines the level of magnification. More than two fingers touching the surface define the pan state, the drawing area moves with the fingers. As the gestures for entering these states are simple, we support fast scene management without the need to shift the focus from the pen's point of sketching to the control hand.

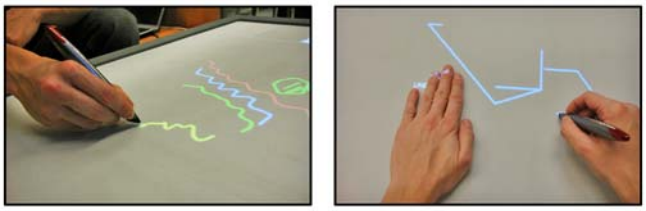


Figure 6. Free-form (left) and straight-line sketching (right).

In addition, we implemented a novel variation of a spring loaded mode for the pen (cf. Figure 6, *right*). If the user touches the tabletop with his flat NDH, the pen remains in drawing mode but is constraint to draw straight lines. The flat hand gesture is easily performed, it can be used anywhere on the surface and directly affects the drawing of the pen. In this sense, it fits well to the coarseness of touch input that sets the mode for the precise drawing tool.

4.2.2 Menu

Using only one finger of the NDH, a menu at the finger's position is shown (Figure 7). Moving the finger drags the menu, once the finger is lifted, the menu fades out. This is an example of a transition from static touch point to point gesture input. Options in the menu can be selected by simply clicking the buttons or stroking over them with another finger of the NDH; this shows a static point to static posture transition. The interaction with the menu can be performed with one hand, the positioning of the menu and selections in the menu are designed for coarse touch input. Moreover, the placement of the menu items is easily learned which aims at the kinesthetic memory and fast repeatability. The layout of the menu is designed to position the buttons left or right of the NDH finger, depending on the handedness of the user.

Figure 7. The menu command. One finger touch shows the menu (left). Dragging the menu (right).

4.2.3 History Tool

The history tool can be used to undo one or more steps depending on the dwelling of the hand's posture. Tapping with the NDH to the opposite side of the menu causes a single undo. The history tool appears for a second and shows one-step-back. This mode can be selected very fast and without even looking at the menu as the whole side serves as responsive area. Holding this hand posture enters the multiple-steps-undo mode and keeps the history tool visible. The history tool can be positioned with the NDH while the pen sets the number of undo steps. This interaction fits well to the high precision DH input of the pen and the gesture action of the NDH touch (cf. Figure 8).

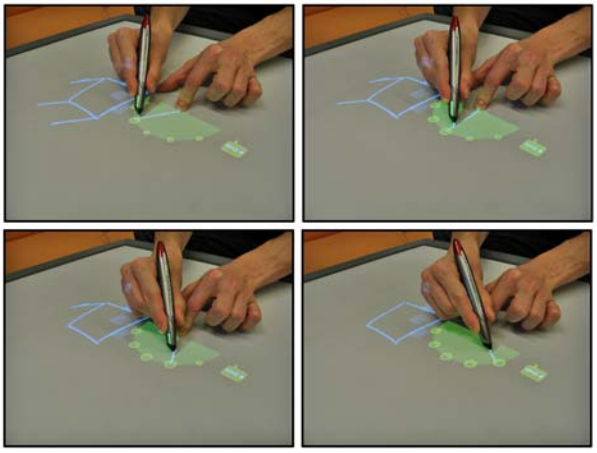


Figure 8. A two finger touch shows the history tool, the precise pen defines the number of undo steps.

4.2.4 Color Picker

The current drawing color can be changed through a HSV color picker (Figure 9). Our implementation handles two simultaneous inputs that control the H and the SV component of the color: the

pen offers precise picking of the H-value, one finger selects the SV-value and two fingers drag the color picker.

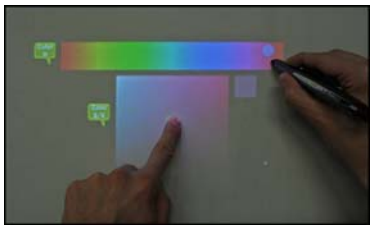


Figure 9. Bimanual color picking. Fine selection with the pen and coarse two-dimensional touch action.

4.2.5 Selection

Our prototype provides three different kinds of selections (rectangular, polygonal and lasso) that illustrate concepts of bimanual sequential dependent interaction. Rectangular selection areas can be defined in two different ways, two-point simultaneous placement or corner placement combined with adjustable dimensions [9]. This choice is made through the sequencing of pen and touch actions. For a two point rectangular selection, the user first defines one corner with his finger. Then he selects the two point option from the menu with the pen. By clicking this button, he sets the diagonal second corner of the rectangle at the pen's position (cf. Figure 10). With this command sequence, one corner for the rectangle can be located first, the option for rectangle selection is chosen from the menu that appears on that location afterwards, and finally the selection rectangle can be set by adjusting the second corner without ever loosing the position of the first corner. In this case, we are not aiming at the different accuracies of pen and touch but rather at the simultaneous use for a bimanual task. Another way of defining a rectangle selection is to choose the option with the pen from the menu without a simultaneous single finger touch on the surface. In this mode, the pen is used to set one corner of the rectangle and subsequently stretch the second corner to define the dimension. Again, the pen is used to perform precise actions.

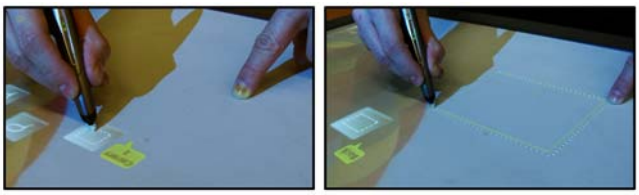


Figure 10. Bimanual rectangular selection with pen and touch. The option is selected from the menu with the pen while the finger already defines one corner position (left). Stretching the rectangle selection with the pen controlling one corner and touch setting the other (right).

For polygonal selections, the points of the polygon shape are defined by the pen while the NDH confirms their position with a single finger touch (cf. Figure 11). With this technique, the points can be placed very accurately with the pen whereas the hand's touch can be performed at any arbitrary position. In contrast, pens with integrated buttons may suffer from a small jitter during a button press. Releasing the pen causes the polygon shape to be closed immediately. The lasso selection is performed solely with the pen.

Figure 11. Polygon selection featuring precise waypoint definition with the pen and intuitive touch confirmation.

4.2.6 Cut/Copy Paste

After a selection is finished, cut and copy actions can be performed on the selected region. Again, we consider the NDH for this task as we can use gestures to provide fast access to all possible modes. Tapping with one finger on the selected region results in a cut action, whereas two finger interaction means copy. Immediately after performing the cut or copy action, the new area can be further positioned with the NDH (cf. Figure 12). Moving the object with the hand is a very intuitive action that can be carried out fast with average preciseness. But when it is required to achieve pixel accurate results, touch is not sufficient in terms of resolution, occlusion and jitter.

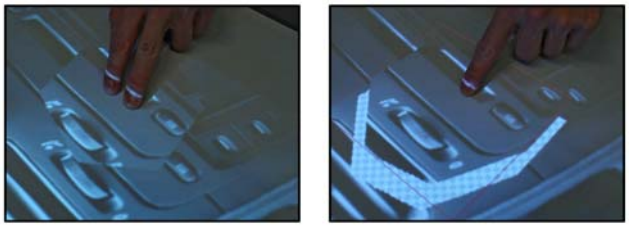


Figure **12**. Copy (left) and cut (right) action.

For this reason, we introduce a novel technique that benefits from the advantages of the pen and touch input devices. We already introduced the underlying concept of dynamically and systematically coupling and decoupling of the two hands. Therefore, we propose the use of the pen in the DH to gain additional preciseness that can be controlled from any position on the table. Once the pen is used simultaneously with the touch to position an object, visual connection between the two hands is shown in form of four lines connecting the pen's point with the corners of the object's bounding box (cf. Figure 13).

The pen's movements are directly applied to the selected object to control subtle transformations; the touch is locked meanwhile to prevent jitter influence. Once the pen is lifted (decoupled), the touch is again controlling the object. We note that according to our design considerations, the pen can be used if necessary to add preciseness, but the action of positioning itself still can be carried out with the NDH touch alone.

Figure 13. Coarse positioning of a selection with the hand (left and center). Pen in second hand allows pixel accurate transformation (right).

5. LABORATORY EXPERIMENT

Previous work has argued for the advantages and disadvantages of pen and touch combinations; however, they have not been investigated in a laboratory experiment. To address this issue, we conducted an experiment that explores the possible assignments of input devices to each of the hands and their effects on efficiency, fluidity, and user preference. Our goal was to understand the differences among the possible input device pairings for a representative task. The experimental task was carefully chosen to tease out the differences between the input device-to-hand pairings, while maintaining ecological validity.

5.1 Participants, Apparatus and Task

Twelve subjects were recruited for our study through an on-line community bulletin board, and paid $20 (USD) for participating. Seven were male and five were female, and their ages ranged from 20 to 50 years old. Eleven of the 12 subjects were right-handed.

Our experimental task consisted of solving and navigating through mazes by drawing a path from a green start marker to a red finish marker (Figure 14). These mazes were designed so that participants had to magnify the maze in order to successfully follow its paths without colliding with the maze walls as well as zoom out in order to plan a path through the maze that would reach the goal. We believe that the maze solving experimental task has a high-level of ecological validity because it matches many graphical editing operations in which a user repeatedly switches back and forth between detailed editing at a high-zoom level and contextual verification of the changes at a low-zoom level (such as when masking a region of a high-resolution image for clipping). In essence, this is a traditional *path following* / *tunneling* task with the added element of route planning.

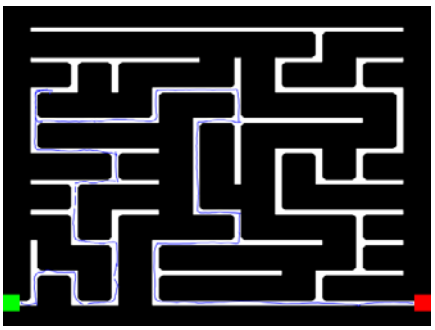


Figure 14. A maze from our experiment with the participant's path stroked through the tunnels.

An error was recorded whenever the participant's stroke intersected with the black walls of the maze. When this collision occurred, a buzzing sound was played, the subject's stroke changed color from blue to red and was stopped. To continue, the participant had to pick a 10 by 10 pixel *continue target* that was displayed at the last valid position before hitting the wall. Upon returning to the white path of the maze, the stroke returned to blue. To complete the maze, the participants had to draw one continuous stroke; each time they lifted the pen or drawing finger, the continue target at the end of the stroke had to be picked to proceed. Each participant controlled the testing application using three different input techniques. Each of the three techniques was a bimanual input technique in which the dominant hand created strokes through the maze and the non-dominant hand zoomed and panned the maze itself. The techniques differed in terms of what input device the dominant and non-dominant hands controlled.

In the first technique, a participant held two pens, one in each hand. While their dominant hand's pen created strokes through the maze, their non-dominant hand controlled a simple marking menu from which they could zoom in/out and pan the maze..The zoom option was selected with a stroke over the right 90 degree region in front of the pen, the pan selection was performed in the left 90 degree region. In the zooming case, a forward motion would zoom into the scene, whereas a backwards motion zoomed out. After learning the left/right assignment for pan/zoom, this marking menu could be used without paying visual attention. We refer to this technique as *Pen/Pen*.

In the second technique, the participants held a pen in their dominant hand, which they used to create strokes in the maze, while they performed two simple gestures for zooming and panning with their non-dominant hand. Two fingers spreading apart or pulling together would zoom in or out respectively, one finger panned the maze. We refer to this technique as *Pen/Touch*. The C/D gain for zooming was the same for the pen's marking menu option and the direct touch stretching gesture. The mapping coefficient was multiplied by a fixed value to achieve a larger zooming effect with less motion.

Our third and final input technique, *Touch/Touch*, combined the non-dominant hand gestures for pan and zoom from the *Pen/Touch* condition with index-finger stroking performed with the participant's dominant hand.

5.2 Hypotheses

Our hypotheses, the confirmation of which will validate our hand / input device pairings, were as follows:

H1: Participants will complete the mazes in less time while using the Pen/Touch technique than when use the Pen/Pen or Touch/Touch techniques.

H2: Participants will commit fewer errors while using the Pen/Touch technique than when use the Pen/Pen or Touch/Touch techniques.

H3: Participants will prefer the Pen/Touch condition over the other conditions.

5.3 Design

We used a within-participant, repeated measures design for our study, with each subject completing 10 mazes using each of the 3 input techniques. The order of the three techniques was balanced between participants. All participants completed the same 30 unique mazes, and maze / technique pairings were balanced. Participants were given instructions before using each technique, and were asked to practice the technique on two practice mazes before starting the experimental trials. In short, our design was:

12 participants x 3 Input Techniques x 10 mazes = 360 trials

5.4 Results

5.4.1 Time Analysis

The time of a trial was recorded as the time between the participant's click of the start button that was shown before each maze and their successful crossing through the "finish" rectangle at the end of the maze. A repeated-measures ANOVA shows that there was a significant difference among the three input techniques ($F_{1,11} = 10.70$, $p < 0.01$), thus confirming hypothesis H1. On average, our participants successfully completed each maze in 42.6s, 36.5s, and 52.7s for the Pen/Pen, Pen/Touch, and Touch/Touch conditions respectively.

5.4.2 Error Rate Analysis

In our study, an error was recorded whenever the participant's stroke collided with one of the walls of the maze.

When this occurred, an error sound was played, the color of the participant's stroke changed from blue to red and could not be continued until the small recover rectangle at the last valid stroke position was picked. Upon reentering the white path of the maze, the sound would stop and the stroke color would return to blue.

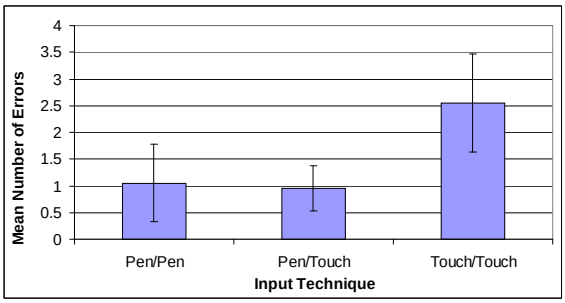


Figure 15. The mean number of errors committed during each maze for each of the three input techniques. Error bars represent 95% confidence interval.

A repeated-measures ANOVA suggests that there is a significant difference among the average number of errors committed by our participants while using each of the three input techniques ($F_{1,11}$ = 11.6, $p < 0.01$). On average, participants committed 1.05, 0.95, and 2.55 errors per maze for the Pen/Pen, Pen/Touch, and Touch/Touch conditions respectively. A post-hoc comparison of means shows a significant difference between the Touch/Touch and both of the other two input conditions in respect to error rate. Figure 15 shows the average number of errors per maze for each input technique.

5.4.3 Preferential Results

At the end of each session, we asked our participants to rank the three techniques in terms of ease of use, accuracy, and overall preference. Table 2 shows the mean rank and standard deviations for each of the three techniques for each of the three measurements (lower numbers indicate a higher level of preference). These results support hypothesis H3, in that our participants seemed to indicate a strong preference for Pen/Touch input over the other two techniques, with 10 of our 12 participants ranking *Pen/Touch* as highest in terms of overall preference.

Table 2. Mean (StDev) rankings for each input techniques.

	Touch/Touch	***Pen/Pen***	***Pen/Touch***
Overall Preference	2.50 (0.67)	2.33 (0.65)	1.17 (0.39)
Ease of Use	2.50 (0.67)	2.25 (0.75)	1.25 (0.45)
Accuracy	2.83 (0.39)	1.92 (0.67)	1.25 (0.45)

5.5 Discussion

In need of investigation is an accounting of the observed differences in task time between our three input techniques. While the number of errors committed certainly accounts for some of the difference in trial times, they do not fully explain it. In addition to recording the trial time and number of errors committed during each trial, our testing application also recorded the number of *zoom* and *pan* operations as well as the number of times that a participant lifted the pen (in *Pen* conditions) or index finger of their drawing hand (*Touch/Touch*) from the table. These numbers provide details allowing us to provide additional insights from the observed differences in completion time.

An examination of the number of zoom operations provides further insights. An ANOVA shows that each of the input techniques had a significantly different number of zooms ($F_{2,22}$ = 23.0, $p < 0.001$). On average, participants zoomed 1.62, 1.89, and 7.91 times per maze for Pen/Pen, Pen/Touch, and Touch/Touch respectively (Figure 16). The much larger number of zooms in the Touch/Touch condition is explained by the lack of precision of the finger for drawing input: participants zoomed in to draw, then back out to gain context in navigation. At the other extreme, participants zoomed significantly less often in the Pen/Pen than in the Pen/Touch condition, despite the identical drawing device. We attribute this difference to the increased awkwardness of using the pen-based menu versus gestures.

Additional timing information can be deduced by examining the number of panning operations during each trial (Figure 16). The mean number of pans was significantly different in each of the input conditions: 0.93, 2.14, and 11.37 for Pen/Pen, Pen/Touch, and Touch/Touch ($F_{2,22}$ = 23.8, $p < 0.001$). Panning is positively correlated with zooming, since zoomed-in mazes require more pans to traverse the space, while requiring frequent zooms in and out to gain context and to draw strokes. As with the mean number of zooms, we see a reduction in the number of pans in the Pen/Pen condition as compared with the Pen/Touch condition. The reason for this result lies in the behavior of the participants, who tried to avoid using the marking menu in the Pen/Pen condition, while hesitating less to perform gestures for zooming in the Pen/Touch condition. Although the marking menu offered only two options to select from and the selection gesture could be learnt after the first usage, we observed a constant focus shift when the participants used the menu. This behavior was not found in the case when they used touch gestures to zoom and pan.

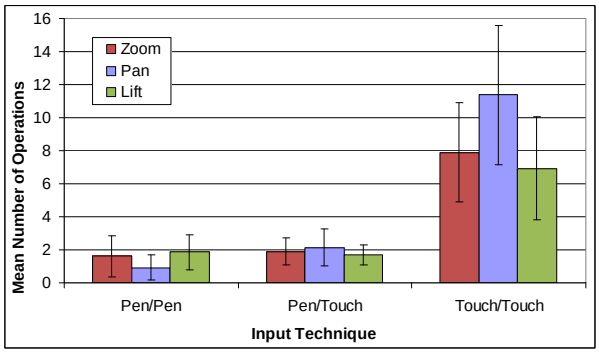


Figure 16. Mean pan, lift, and zoom actions per trial.

The final measurement that helps to explain the observed differences in task time among the input techniques is the number of times that a participant lifted their dominant hand from the tabletop. Again, we see a significant difference among the input techniques ($F_{2,22}$ = 15.38, $p < 0.001$), with a large difference between the Touch/Touch input technique (6.93 lifts/maze), and both the Pen/Pen (1.86 lifts/maze) and Pen/Touch (1.68 lifts/maze) techniques (Figure 16). The fact that a significant higher number of lifts occurred with the Touch/Touch technique seems to be caused by two reasons. First, during the zoom and pan operation, most participants lifted the pen or drawing finger. They could have left the finger or pen on the last drawing position while zooming without causing an error. During panning, this would have resulted in a similar effect of dragging a sheet of paper under a pen. Nevertheless, they felt more comfortable to lift the finger or pen during these actions. Second, due to larger occlusion areas in the Touch/Touch scenario, the stroke was frequently interrupted for hand positioning reasons. Taken together, the clear evidence in support of our hypotheses and these additional details provide strong validation for our assignment of input devices to the hands.

6. CONCLUSION

In this paper, we presented a survey of prior work on bimanual input which led us to a set of principles for the design of two-handed input techniques. These principals included the assignment of pen and touch when considering bimanual input on a horizontal display. To justify this guideline, we conducted an experiment in which three different input combinations for two-handed interaction on horizontal surfaces were tested: *touch and touch*, *pen and pen* and *pen and touch*. The results of this experiment suggest that *pen and touch* input is superior in terms of speed, accuracy, and user preference. As a further validation of our design principals, we implemented a prototype graphical editing application, which includes a new method of teaching bimanual gestures. Considering early feedback we gathered about our prototype implementation, we are confident in the application of our design principles for the creation of future two-handed interactions.

7. FUTURE WORK

We would like to further investigate the learnability of our system in repeated sessions with the same users. We are excited about the opportunities of gathering valuable insights from a user customizable system; therefore a next step would be to enforce the development towards a more flexible application. Accordingly, we would like to explore the mechanism and benefits of an adjustable feedback concept, extending our proposed solution to better accommodate users' requirements.

8. REFERENCES

1. Anoto. http://www.anoto.com
2. Balakrishnan, R. and Hinckley, K. (1999). The role of kinesthetic reference frames in two-handed input performance. *UIST 1999*.p. 171-178.
3. Balakrishnan, R. and Hinckley, K. (2000). Symmetric bimanual interaction. *CHI 2000*. p. 33-40.
4. Balakrishnan, R., Patel, P. (1998). The PadMouse: facilitating selection and spatial positioning for the non-dominant hand. *CHI 1998*. p. 9-16.
5. Benko, H., Wilson, A., and Baudisch, P. (2006). Precise selection techniques for multi-touch screens. *CHI 2006*. p. 1263-1272.
6. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. (1993). Toolglass and magic lenses: the see-through interface. *SIGGRAPH '93*. p. 73-80.
7. Butler, C.G., Amant, R.S. (2004). HabilisDraw DT: A Bimanual Tool-Based Direct Manipulation Drawing Environment. *CHI 2004*. p. 1301-1304.
8. Buxton, W., and Myers, B. (1986). A study in two-handed input. *CHI 1986*. p. 321-326.
9. Casalta, D., Guiard, Y., Beaudouin-Lafon, M. (1999). Evaluating two-handed input techniques: Rectangle editing and navigation. *CHI 1999*. p. 236-237.
10. Chatty, S. (1994). Extending a graphical toolkit for two-handed interaction. *UIST 1994*. p. 195-204.
11. Chatty, S. (1994). Issues and experience in designing two-handed interaction. *CHI 1994*. p. 253-354.
12. Cutler, L.D., Frohlich, B., Hanrahan, P. (1997). Two-handed direct manipulation on the responsive workbench. *I-3D*. p. 107-114.
13. Dietz, P. Leigh, D. 2001. DiamondTouch: a multi-user touch technology. *UIST '01*. p. 219-226.
14. Esenther, A., Ryall, K. (2006). Fluid DTMouse: Better mouse support for touch based interactions. *AVI 2006*. p. 112-115.
15. Flider, M. J., Bailey, B. P. (2004). An evaluation of techniques for controlling focus+context screens. *GI 2004*. p. 135-144.
16. Forlines, C., Wigdor, D., Shen, C., Balakrishnan, R. (2007). Direct-Touch vs. Mouse Input for Tabletop Displays. *CHI 2007*. p. 647.
17. Guiard, Y. (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19(4), 486-517.
18. Hinckley, K., Baudisch, P., Ramos, G., Guimbretiere, F. (2005). Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scriboli. *CHI 2005*. 451-460.
19. Hinckley, K., Czerwinski, M., and Sinclair, M. (1998). Interaction and modeling techniques for desktop two-handed input. *UIST 1998*. p. 49-58.
20. Hinckley, K., Pausch, R., Proffitt, D., Patten, J., and Kassell, N. (1997). Cooperative bimanual action. *CHI 1997*. p. 27-34.
21. Kabbash, P., MacKenzie, I.S. & Buxton, W. (1993). Human performance using computer input devices in the preferred and non-preferred hands. *InterCHI 1993*. p. 474-481.
22. Krueger, M., VIDEOPLACE and the interface of the future. *The art of human computer interface design*, B. Laurel, Editor. 1991, Addison Wesley: Menlo Park, CA. p. 417-422.
23. Kurtenbach, G., Fitzmaurice, G., Baudel, T., Buxton, B. (1997). The design of a GUI paradigm based on tablets, two-hands, and transparency. *CHI 1997*. p. 35-42.
24. Latulipe, C., Mann, S., Kaplan, C., Clarke, C. (2006). SymSpline: symmetric two-handed spline manipulation. *CHI 2006*. p. 349-358.
25. Leganchuk, A., Zhai, S., & Buxton, W. (1998). Manual and cognitive benefits of two-handed input: an experimental study. *ToCHI*, 5 (4). p. 326-359.
26. MacKenzie, I. S., Guiard, Y. (2001). The two-handed desktop interface: are we there yet?. *CHI 2001I*. p. 351–352.
27. Matsushita, N., Ayatsuka, Y., Rekimoto, J. (2000). Dual touch: A two-handed interface for pen-based PDAs. *UIST 2000*. p. 211-212.
28. Matsushita, N., Rekimoto, J. (1997). Holo Wall: Designing a Finger, Hand, Body, and Object Sensitive Wall. *UIST 1997*. p. 209-210.
29. Maxell PenIT. http://www.maxell.co.jp/e/products/industrial/digitalpen/
30. Myers, B. A., Lie, K. P., Yang, B-C. (2000). Two-Handed Input Using a PDA and a Mouse. *CHI 2000*, 41-48.
31. Owen, R., Kurtenbach, G., Fitzmaurice, G., Baudel, T., and Buxton, W. (2005). When it gets more difficult, use both hands: exploring bimanual curve manipulation. *GI 2005*. p. 17-24.
32. Rekimoto, J. (2002). SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. *CHI 2002*. p. 113-120.
33. Shoemaker, G., Gutwin, C. (2007). Supporting Multi-Point Interaction in Visual Workspaces. *CHI 2007*. p. 999-1008.
34. Ullmer, B., & Ishii, H. (1997). The metaDESK: Models and prototypes for tangible user interfaces. *UIST 1997*. p. 223-232.
35. Wu, M. and Balakrishnan, R. (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *UIST 2003*. p. 193-202.
36. Wu, M., Shen, C., Ryall, K., Forlines, C., Balakrishnan, R. (2006). Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. *IEEE Tabletop 2006*.
37. Yee, K.-P. (2004) Two-handed interaction on a tablet display. *CHI 2004*. p. 1493-1496.

Semantics - Based Applications

Semiotic Engineering in Practice: Redesigning the CoScripter Interface

Clarisse Sieckenius de Souza
SERG - Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente 225, Gávea
22453-900 – Rio de Janeiro, RJ - Brazil
+55 21 3527-1500 ext. 4344
clarisse@inf.puc-rio.br

Allen Cypher
IBM Almaden Research Center
650 Harry Rd. MS: B2-422
San Jose, CA 95120
+01 408 927-2513
acypher@us.ibm.com

ABSTRACT
Semiotic Engineering uses semiotic theories to characterize human-computer interaction and support research and development of interactive systems. In order to show the value of Semiotic Engineering in design, we illustrate how semiotic concepts have been used in the analysis and generation of redesign alternatives for a web browser-based program called CoScripter. We also discuss how specific perspectives and expectations about the design process can increase the benefit from Semiotic Engineering in design activities, and describe our future steps in this research.

Categories and Subject Descriptors
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Human Factors.

Keywords
Semiotic engineering, graphical user interface design, UI design methodology, end user programming.

1. INTRODUCTION
Contemporary practitioners of user interface design are familiar with user-centered design, which focuses on how a user interacts with a software program. In this paper, we instead focus on how a software designer communicates with a user through the software's interface. This shift in perspective results in valuable insights that can assist designers in making design tradeoffs. We exemplify this value by analyzing the interface of one specific program, named CoScripter.

CoScripter is a system for sharing "how-to" knowledge about activities in a web browser. It consists of an extension to the Firefox web browser (Figure 1) and an associated wiki website. The extension enables users to record their actions in the browser as scripts: when the 'record' feature is turned on, CoScripter records a script of every action performed in Firefox – every click, every command, and any text that is typed. CoScripter can then replay this script, performing all of the clicks, commands, and typing automatically. These recorded scripts are stored on a public wiki, so that all users can replay scripts created by others.

Scripts can be used to automate many tasks, such as filling in forms. Scripts can also be used as a kind of "how-to" knowledge: as CoScripter replays actions, it highlights each action on the screen, turning buttons and links green as it clicks on them, and typing green text into text boxes. That way, users who have never performed a particular task can learn by watching the script as it executes. In Figure 1 we see the instruction for entering the user's area code ('650') into the *Area Code:* text box. On the web page, green highlighting around the text box and the green *650* text show the user what CoScripter is doing.

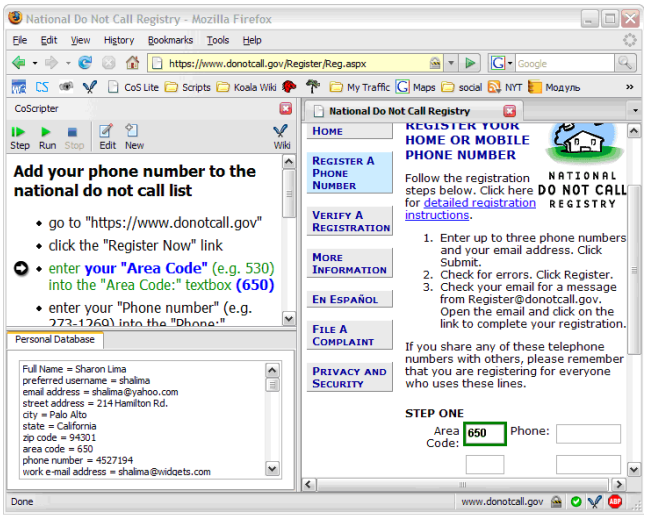


Figure 1: CoScripter performing the third step in a script.

Semiotic Engineering [3] and other analytical frameworks (such as Activity Theory [8] and Distributed Cognition [7]) face the challenge of showing how they can or should be used in design. Although the valuable insights that Semiotic Engineering produces in analytic tasks have been recognized [1], there is the potential that the theory will also lead to design methods that can be demonstrated to improve the quality of interactive systems.

In this paper, we show how Semiotic Engineering concepts can help designers *name and frame* the elements and structures

AVI'2008, May 28–30, 2008, Naples, Italy.

involved in the design problem space [11] and how these bring about solution paths and alternatives that may not arise from user-centered design methods. Concrete illustrations from redesigning the CoScripter sidebar interface language provide support for the points we make.

We begin with a brief overview of Semiotic Engineering. We then present elements of CoScripter's interface design and analyze them in terms of Semiotic Engineering. Next, we take the results of our analysis as input, and show how a designer can apply Semiotic Engineering concepts to generate interface alternatives for CoScripter. Finally, we summarize how the insights gained from a semiotic perspective can be practically useful in HCI design, discuss how specific design perspectives can boost the benefits of our approach, and point to future work in this research project.

2. A BRIEF OVERVIEW OF SEMIOTIC ENGINEERING

A prime goal of semiotic investigation is to understand how people use signs to communicate [6]. Semiotic Engineering provides a semiotic account of human-computer interaction, stressing the fact that designers of interactive software communicate their design vision to users through their systems. They express their message through the signs in the interface – its words, icons, graphical layout, sounds, and widgets (*e.g.* buttons, links, and dropdown lists). Users then discover and interpret this message as they interact with the system. Since the designer can't personally be present when a user interacts with the software, the signs in the interface, along with their dynamic behavior, are the sole means available to the designer to get the user to understand what the software does, and how to use it.

It is important to realize that although systems' interfaces share a number of interactive patterns (*e.g.* many systems use pop-up dialog boxes when communicating about opening and saving files), every system has a unique interactive language whose semantics is determined by the system's unique semantic model. This language, which must be learned by users in a process that resembles second-language acquisition, is also delivered through the interface, as an important part of the designer's message to users. In other words, the message contains the very code in which it has been signified (by the designer) and that must be employed (by users) to communicate back with the system. Therefore, designing and delivering this message is a very complex task in itself.

Semiotic Engineering focuses on communicative rather than cognitive aspects of HCI analysis and design. It is important to realize, however, that it is not taking the stance that the whole purpose of an interface is to let the designer communicate with the user. The interface of course is mainly there to let the user perform actions. But the interface also serves this other crucial function of communication, without which users can't even begin to achieve their goals: The interface delivers the designer's message about how (*i.e.* using which signs) to use the system, and why (*i.e.* to what effects and advantage).

As is the case with all human communication, the designer's messages can be interpreted by users in ways that were not *meant* by the designer. Some such *misinterpretations* will lead users to errors as soon as they are generated. Others. however, may take longer to do so. Yet, some *misinterpretations* may occasionally prove to work just fine for the users. In fact, some of these may be instances of *repurposing* the signs in the designer's message. A popular example of this phenomenon is how most microwave oven users repurpose its original design meaning – a home appliance to *cook* food – into one that can be used to *warm and heat up* food. This example is important because it also shows that repurposing is not necessarily a sign of design *failure*. It can in fact be interpreted as a participatory redesign activity (*i.e.* one in which users are involved) *at use time* [4].

This example also demonstrates an important concept in semiotics. In natural settings, *meaning* is an evolving and unpredictable process, rather than a static abstract end point that we can or should eventually reach in the process of interpretation. Rather, we generate meanings that are continually 'revised' and 'elaborated' as a result of our encounters with them throughout life. At any given point in time, if asked what these meanings *are*, we provide our *current account*, which may turn into something else as soon as these signs co-occur with other signs that show them to be incomplete, incorrect, or simply related to some other sign that we had not been aware of before. For instance, most GUI users know that  refers to 'close window'. However, some users may take this to *mean* 'exit program', or even 'cancel / undo the action that took me here (to this window)'. Such a reinterpretation may work just fine in some situations, but it may also put the user into trouble in others (for example, the window may disappear without the program terminating and without the action being cancelled).

The process of constantly generating and revising meanings – called *semiosis* – has been formally defined by Peirce [10] as a particular kind of logical reasoning, called *abduction*. To Peirce, abduction is the primary sense making ability that we all share. It permeates human intelligent behavior, from simple common sense reasoning to sophisticated knowledge discovery in science and philosophy. Thus, in this perspective, users will *surely* generate use meanings that differ from design meanings. And this moves the design target from encoding the *correct* range of users' meanings to encoding meanings that communicate and achieve goals that are useful and enjoyable to users, in both anticipated and unanticipated situations. The designer's best choice is then to design a system of signs that is *easy to learn* (the cognitive dictum), but also *efficient and effective in communicating meanings* back and forth along the designer-system-user path (the semiotic dictum).

To facilitate the users' learning of interface sign systems, designers must cue the interpretations they expect from users by introducing signs that have the potential to trigger consistent abductions in the users' minds. Cueing expected interpretations is what all communicators do in a wide variety of contexts, such as teaching, politics and marketing, to name but a few. Therefore, the semiotic engineering of interface sign systems (or languages, in a computational sense), involves the use of rhetorical strategies that *lead to* certain cognitive effects, but are not *cognitive* in essence.

An important strategy in communicating the design vision to users, and concomitantly the interface language in which it is expressed and must be put to use, is *elaboration*. Elaboration amounts to communicating further details like circumstances, explanations and relations (*e.g.* analogies and contrast). Another

widely used strategy is *redundancy*, which amounts to repeating the same meanings (in whole or part) in various related situations. This achieves the cognitive effect of reinforcing certain sense-making patterns that will be useful to get the designer's message across.

Finally, there are dozens of classes and sub-classes of signs that can be manipulated to help communicators achieve their intent. Three classes of signs – *symbols*, *indices* and *icons* – will probably sound familiar to an HCI audience. These terms have lost most of their original theoretical tenets when they crossed the disciplinary boundary between Semiotics and HCI. But because they will play a role in our redesign of CoScripter's sidebar language, we will briefly retrieve some aspects of their original definition by Peirce [10].

This classification applies to how the interpretation of a sign is cued by its representation. *Icons* are signs whose interpretation is cued by representations that resemble (*i.e.* that reproduce the perceptual experience caused by the presence of) what they mean. So, for example,  can be classified as an icon in all contexts where it is used to mean 'a folder'. *Indices* are signs whose interpretation is cued by representations that have a logic, conceptual, causal, or other relation with what they mean. For example,  can be classified as an index in all contexts where it is used to mean 'cut'. Scissors are instruments for cutting.  is, of course, an *iconic* representation of 'scissors', but an *indexical* representation of 'cut' (which is itself an indexical representation of what this popular editing function really is – it represents this function's meaning *by analogy*). *Symbols*, at last, are signs whose interpretation is cued by representations that are associated to certain meanings *by convention*. Thus,  is a symbol that by convention *means* 'close current window'.

3. REDESIGNING COSCRIPTER

In this section we will show how Semiotic Engineering concepts have been used first to analyze the CoScripter *s*idebar *i*nterface *l*anguage (SIL), and then to design an alternative interface language (SIL*). SIL* has not been implemented in a prototype or tested experimentally. As a result, we only include an analytic evaluation of the communicative qualities of SIL* compared to SIL. This evaluation, nonetheless, is sufficient to highlight the main contributions of Semiotic Engineering in design tasks.

3.1 The Sidebar Interface Language

A redesign of the CoScripter sidebar is motivated by the fact that users have been confused by the current interface language. The first step in the semiotic engineering of an alternative interface is to retrieve the gist of what the designers want to communicate to CoScripter users. The second step is to analyze the sign classes and sign system structures that they are using to convey their message. The third step is to produce alternative classes and structures – based on analytic properties – that should make the sidebar language easier to learn and more efficient and effective in communicating the designers' intent.

3.1.1 Messages from the Designers

There are two important methods for deriving the designers' message to users. One, of course, is to *ask* them to say what the message is and to discuss how they are expressing it through the interface. The other is to inspect the interface for signs that emerge with interaction, to contrast them with the system documentation and help, and *reconstruct* the designers' message [5]. In this study we have used both methods, and concluded that the gist of what CoScripter's designers intend to communicate to users can be summarized in 10 messages, where 'we' means 'we, the designers' and 'you' means 'you, the users':

a) We don't have much space, so we will try to communicate as much meaning with as few signs as possible.

b) CoScripter is a system you can use to record and run (play back) Web scripts, and share them with a community of users. The scripts' content is represented in English-like language to make it easier for you to follow the actions they execute.

c) Scripts can be edited and saved (including saved as another script, with a different name).

d) Scripts can be public (i.e. shared with the whole community of CoScripter's users) or private (i.e. saved for your usage only). We think you will be willing to share most of your scripts with others.

e) When editing a script, new or not, take advantage of the automatic recording facilities in CoScripter. You can, however, type script commands directly in the scripting language.

f) If automatic recording is not 'on', you can switch it 'on' and then 'off' at any time while editing your scripts.

g) You can run your saved scripts in two modes: 'non-stop' or 'step-by-step'.

h) You can stop recording and playback at any time.

i) You can abandon script recording and editing at any time, and you can also delete saved scripts.

j) More information from designers and from the community of users can be found in CoScripter's Wiki and Forum.

From (a) to (j) we see different categories of content, some of them present in more messages than others. The first category is the system's **purpose** – to record and run Web scripts, and share them with a community of users. (b) is an explicit message in this category; (d) and (j) communicate implicit messages in this category, too. The second category is the **system's functions and operations**. Messages (c), (f), (g), (h), and (i) communicate just this kind of content. Message (j) also fits in this category, as does the second part of messages (b) and (e), regarding the use of the script language and text. Finally, the third category of messages is the **designers' intent, values and expectations**. Messages (a), (d) and (e) express explicitly the designers' attitude and values, marked by the presence of 'we' (in (a) and (d)) and by the word 'advantage' (in (e)). In passing, we should note that the first two categories are basically *objective*, in that they are talking about the object of design, whereas the third is basically *subjective*, in that it is talking about the subject of design. Compared to user-centered design approaches, this is an important contrast.

There are, of course, other ways to phrase and organize the content expressed in the messages above. For instance, one way would be to constrain message phrasing so that each message conveys just one content unit, and fits into just one of the three proposed categories. In this way, there would be no *overloading*

of communicative expression, with more than one unit of content and/or intent per message. However, there are basically two advantages in allowing for such *overloading*. One is that this kind of phenomenon is pervasive (and arguably unavoidable) in natural communication, and thus it could and should be explored in computer-mediated communication as well. The second is that this helps us address from the very start of the design cycle an important constraint in CoScripter – very little screen space for communicating all 10 messages. *Overloading* is a possible strategy that can be used to solve communicability problems.

Each message can be analyzed further in terms of traditional semantic categories borrowed from natural languages and used in many representation languages, including programming and interface languages. These are: **actor**, **action**, **object**, and **modifier**. So, for example, 'script' is an **object**, and there are two **modifiers** that can be applied to it: 'public' and 'private'. 'Run' is an **action**, to which two other **modifiers** can be applied: 'non-stop' or 'step-by-step'.

Just as in natural language, however, these categories are sometimes recursive. For example, the **action** 'edit' accepts two **modifiers** that, unlike in the previous examples, are not atomic concepts (*e.g.* 'private', 'non-stop') but complex structures in themselves: 'by **actor**=CoScripter **action**=record **object**=script' and 'by **actor**=User **action**=typing **object**=commands'. Moreover, all communication with CoScripter is determined by **context**. For instance, 'stop' and 'cancel' are not valid content elements in contexts where there is no action taking place. Likewise, message contents can be used to achieve different purposes in different contexts: 'stop recording' means something different in an *execution* context from what it means in an *explanation* content.

So, although CoScripter's communicative domain is highly restricted, it encompasses thoroughly complex communicative structures that must be effectively and efficiently expressed in SIL. Additionally, since screen space is small, communication cannot be verbose, and – as already noted above – single signs in the signification system will often communicate multiple message contents. As a consequence, the design of such signification system (SIL) must be done so that users will be able to learn and express these complex communicative structures as quickly and easily as possible. We can expect the 'cognitive' loads associated with the interface language acquisition process to increase as the underlying grammars move from regular to context-free, context-sensitive, and unrestricted [2]. But, especially because non-expert users (to whom CoScripter is targeted, by the way) may not be familiar with the processing constraints of computer languages, users may wish SIL to express far more than easier-to-learn languages (*e.g.* regular or context-free) can express [13]. In short, lower language acquisition loads run contrary to higher language expressive power, a design tradeoff that can best be formulated and framed when designers think beyond 'usability' (highly influenced by cognitive criteria) and into 'communicability' (highly influenced by semiotic criteria).

3.1.2 Signs and grammar of CoScripter's SIL

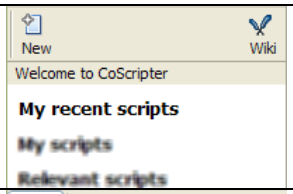
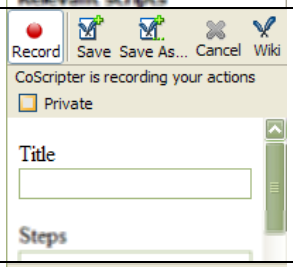
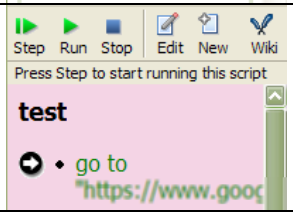
CoScripter's sidebar appears in three alternative basic forms (see Table 1). Each form has its own sub-variations depending on local context values, as will be discussed later. Signs in SIL can be visual (like ), textual (like 'Welcome to CoScripter'), or hybrid textual and visual (like 

).

As a rule, hybrid signs express actions, some of which have modifiers expressing states like 'enabled'/'disabled' or 'active/inactive'. Although related, these states are not synonymous. For instance, the 'Run' button is disabled when there is no script in the sidebar, since there is nothing to run. When a script is present, the 'Run' button is enabled, and it is 'inactive' when CoScripter is not currently running that script, and 'active' when CoScripter is running the script. Given this distinction, both pairs of modifiers must be signified in SIL.

Learning an interface language is facilitated when designers adopt *patterns* from similar or related languages. Thus, the expression of the modifier pair 'enabled/disabled' can be more easily learned if it is encoded using the gray color for 'disabled' (and other color(s) for 'enabled'). There is also a common pattern for expressing the 'active/inactive' modifier pair, namely a *toggle* element. A toggle button for instance is 'active' when pressed down, and 'inactive' when up.

Table 1. CoScripter's Sidebar Interface Language

Basic Sidebar Contexts	Description
	**Start**: Sidebar when user first *calls* (or opens) CoScripter
	**Edit**: Sidebar when user opens a 'New' script for editing
	**Run**: Sidebar when user runs a script

Because context is such an important dimension in CoScripter's design, a context-sensitive grammar naturally comes to mind when designing SIL. However, because context-sensitive grammars also impose higher cognitive loads on learners than context-free and regular grammars, the designers of a signification system must be especially attentive to regularity in their underlying grammar. For example, in row 2 of Table 1, designers are communicating to users a number of things:

- That CoScripter is recording the user's actions in the web browser (the 'record' button is toggled down).
- That the recording can be stopped (if the user toggles the 'record' button up).

- That the designers think it is advantageous to start editing by recording actions (the default state of the 'record' button is active, without the user's telling the system to record anything).

If the user gets all of these messages right, then she is likely to infer (by *abduction*, as discussed in the previous section) that actions can be stopped or initiated by pressing toggle buttons corresponding to them. However, if she tests this inference when *running* scripts, she will find out that this is not the case, because the grammatical pattern for *running* differs from the grammatical pattern for *recording*. That is, in row 3 of Table 1 we see that there is a 'stop' button in the interface. Instead of using a single toggle button, *running* is instead implemented with one button to start running, and a different button to stop running. Therefore, users must not only learn another rule to express the same communicative type of active/inactive, but they must also learn which rule is used in which context. The absence of a 'stop' button in the editing interface facilitates the inference that in order to stop recording, the user must press the 'record' toggle button. But the presence of the 'stop' button in the execution interface might lead to other kinds of equally valid inferences. For example, the user might suppose that by pressing 'stop', she would quit the execution mode and go back to the initial state of the system (why not?). So, it would make perfect sense to her to try to stop execution (non-stop or step-by-step) by *toggling* the 'step' and the 'run' buttons. But this inference is not consistent with the underlying SIL grammar. As a consequence, the user must incur the additional cognitive cost of learning grammar rule exceptions and specificities by trial and error.

Context is the most difficult dimension to master in SIL. The user views context in terms of high-level conceptual tasks, which are termed *modes.* The main modes for the CoScripter user are *starting* CoScripter, *editing* a script (by recording or typing), and *running* a script. The actual program that implements CoScripter has a corresponding context, termed *states*. Unfortunately, despite being similar, CoScripter's states are not identical to the user's modes, and this shift in ontology may give rise to a lot of confusion.

Consider a user who wants to 'cancel' her activity in one mode and return to the previous mode. For example, suppose the user starts CoScripter (Table 1 row 1) and then transitions to creating a new script (Table 1 row 2) by clicking the 'New' button. The user immediately decides to cancel this activity and clicks on the 'cancel' button. This change of mode coincides nicely with CoScripter's states, and causes a transition back to the start *mode* (Table 1 row 1).

Now consider the example of a user who wants to 'cancel' her activity in the run mode and return to the edit mode. The user is editing a script, saves it, runs it, and sees that something went wrong. There is no 'cancel' button in the context shown in Table 1 row 3, so she cannot tell the system to 'cancel' run mode and return to the previous edit mode.. Instead, the interface only offers an 'edit' button, whose semantics implies not "returning to" the previous edit mode, but rather "progressing to" a new edit mode. At this point she may already wonder why 'cancel' is not uniform, and this introduces the potential for wrong abductions in sense making.

3.2 Semiotic Engineering in design activities

In the previous section we showed how semiotic concepts can be used to analyze the cognitive and communicative tradeoffs associated with various design choices in the signification system that conveys the designers' messages. Can Semiotic Engineering help design a better signification system for CoScripter?

An answer to this question actually involves more than just elaborating a different SIL. It involves choosing the appropriate evaluation criteria that will help designers (i) *anticipate* that they have found a better solution, and (ii) *verify* that their anticipation is actually correct. Whereas we can think of analytic criteria that can adequately account for anticipation, only user studies can adequately account for verification. Because the research reported in this paper is still in progress, we will concentrate on anticipation and save verification for further stages in the project.

Going back to the ten messages that designers want to communicate with SIL (cf. (a) to (j) in subsection 3.1.1), we see that SIL communicates different content categories in different ways. **System's functions and operations** are communicated directly with hybrid signs composed of a visual form and a textual label. Because these functions and operations are associated to specific modifiers and contexts, the signs that express them have different form *inflections* (*e.g.* a sign may appear in gray color to express 'disabled', or in full color to express 'enabled') and different *distributions of occurrence* (*e.g.* the 'cancel' sign co-occurs with the 'save' sign, but not with the 'run' sign). *Inflection* and *distribution of occurrence* are used to communicate the **designers' intent, values and expectations.** Together, they are components of indexical representations of important meanings. For example, *distribution of occurrence* is used to express the designers' concern about optimizing screen space use in SIL. Visibility and invisibility of representation (which signal distribution) signify lack of space. Also, *inflection* is used to express the designers' belief/expectation that certain default values will call the users' attention to the advantages of using CoScripter in one way, rather than the other. Here, a mix of iconic and symbolic signs is used to convey the intended message (*e.g.* button states resemble those in most electronic appliances, and many of their control conventions are adopted in CoScripter).

The use of *distribution* to resolve design tradeoffs, however, comes at the expense of communicating the **system's purpose** more directly. For example, instead of communicating all the main purposes of the system simultaneously (recording, running, and sharing scripts), designers of SIL chose to distribute communication across different interactive modes. Modes, however, constitute a cognitive challenge for interface language acquisition. Can Semiotic Engineering improve SIL in view of these givens and findings, and reduce cognitive load in the process?

3.2.1 A new context-sensitive grammar for SIL

In this subsection we present only the sign types and rule types that can improve the designer-to-user meta-communication and facilitate the users' learning. A full-fledged grammar specification is not only tedious, but also beside the point of this paper. All improvements proposed for SIL are meant to eliminate ambiguities, facilitate learning, and preserve the communication of the designers' intent, values and expectations.

3.2.2 *Using inflection and distribution to express context transitions*

Semiotic Engineering's focus on signification systems and communication processes directs designers to consider their sign choices in terms of what they communicate (and mean) *to users*, what users may mean by them, and what users take them to communicate back *to the system*. In redesigning SIL we can first choose sign inflections and sign distribution to be *indexical representations* of context. Next, by building a consistent set of rules to control the use (and computational interpretation) of such representations, we can produce *conventions* (or *symbolic representations*) that mean and communicate the designers' intent.

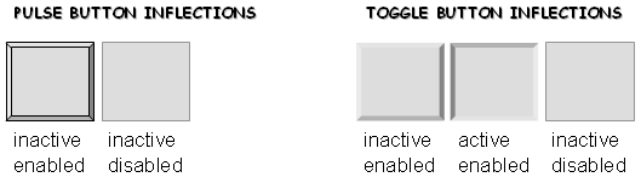


Figure 2: Classes of buttons and their inflection in SIL*

As shown in Figure 2, there are two classes of buttons in SIL* (as we call this redesigned language): pulse buttons and toggle buttons. Each class has inflections. Because pulse buttons express instant actions that have no duration over time, 'active/inactive' distinctions are unnecessary. Inflection must only account for the 'enabled/disabled' distinction. Toggle buttons, however, must communicate 'active/inactive' distinctions in order to account for duration aspects of certain actions over time.

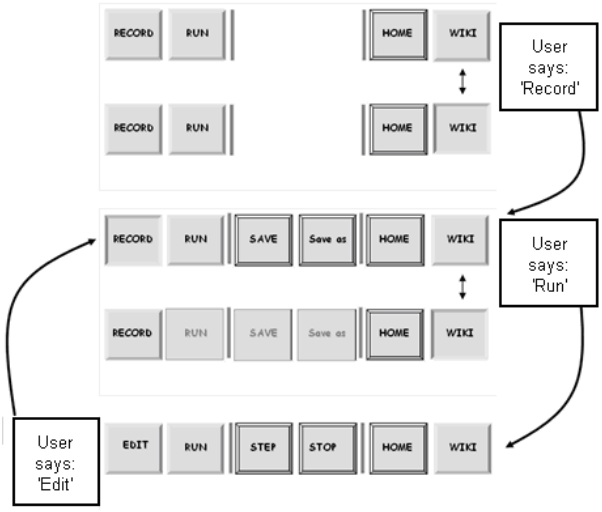


Figure 3: The five states of SIL*, showing transitions when the user communicates to the system.

Context can also be signified by distribution, as in the original SIL design. In SIL* distribution plays a major role in communicating, indexically, the system's mode (and the *interactive* state – the user's current state of discourse or conversation). Figure 3 illustrates how button inflection and distribution combine to express complex contextual dependencies in SIL*. Note that there are only three signs expressed as toggle buttons: 'record', 'run' and 'wiki'. The 'edit' sign is a synonym of 'record', which will be discussed in the next section. Note, also, that button inflection is being used to communicate a previous state of the system. When the 'record' function is active and the user chooses to activate the 'wiki' function, the system can preserve the interactive context prior to entering CoScripter's wiki. This feature is not currently implemented in the system, but the example serves to show how a relatively simple set of signification patterns can account for fairly elaborate human-computer interaction. The 'home' button is there to facilitate access to the system's initial state, where users can choose to 'record' a new script, or 'run' an existing script. Again, this latter feature is not implemented in the current system. In many interactive contexts, in order to run an existing script users must follow a long and complicated communicative path, through the wiki, with many opportunities for error. SIL* gives faster access to such scripts, and helps prevent wiki navigation errors.

3.2.3 *Some analytic indicators of improvement*

The SIL* grammar is presented here with impoverished visual elements, compared to the signs actually used in SIL (see Figure 1 in the Introduction). On the one hand, this helps the reader focus on the crucial semiotic elements of the interface. It does not mean, however, that 'run', 'record', 'stop', and the like cannot or should not be expressed as images, or as a combination of image and text. On the other hand, the introduction of visual signs is in itself a fairly complex Semiotic Engineering task. What images should be used? How should they be rendered in inflected button forms? These issues will not be discussed here for sake of brevity and clarity.

Although SIL* is still a context-sensitive grammar – hence more difficult to learn than a context-free or a regular grammar – it explores inflection and distribution regularities to help users 'infer' how to interpret and express communication in context. For example, guessing the effect of communicating 'Run' when the user is in the context represented by the top pair of sidebar representations in Figure 3 is not difficult. Neither is it difficult to infer the effect of communicating 'Wiki' when the user is viewing a sidebar like the one at the bottom of Figure 3. Note that the redesign alternative builds a convention for how indexical signs are used in the interface. Note also that if we add images, it is likely that visual representations will introduce redundancy into this communication and reinforce the design messages.

In its current form, nonetheless, SIL* sorts out some of the confusion between *mode* and *state*, mentioned in 3.1, but it requires the implementation of additional semantics compared to SIL. Therefore, designers must trade off superior semiotic and cognitive quality, against increased programming costs.

Another analytic indicator of improvement in SIL* is the introduction of a visual grammar – a conventional sign (or *symbol*) – in the sidebar layout. There are constantly three different segments in the sidebar, separated by '|'. The leftmost segment constantly communicates the two main operations in CoScripter: *record* and *run* scripts. Both are toggle buttons, constantly enabled to signal a switch of context. The 'record' button is aliased to 'edit' in the context of 'run'. This choice breaks the visual and lexical regularity of other signs in SIL*. However, this is a conscious design choice, an *index* intentionally introduced to tease users into wondering whether there is more than 'recording' involved in building and refining scripts. And

indeed there is, as messages (b) and (e) in 3.1.1 tell us. Also, the suggested distribution of synonyms across contexts preserves the designers' intent of promoting recording as the prime strategy for generating scripts.

The rightmost segment has buttons of different classes: a pulse and a toggle button. However, both share a particular content feature that is important and not well-resolved in SIL: stepping out of the current context by big horizontal or vertical leaps. A big horizontal leap in SIL is to go back to the system's initial state. One of this paper's authors actually uses the side effects of other actions to retrieve the initial state (*e.g.* closing the sidebar and opening it again from the interface browser). In SIL* this leap is always available to users, who can simply press the 'home' button. A big vertical leap is opening the CoScripter wiki in the web browser, for help or script harvesting. The design solution in SIL is very similar to the one proposed in SIL*. However, because control of the current context is not as consistent in SIL, the user may be confused when trying to anticipate the differences between pressing the 'run' button in the sidebar or the 'run' button in the browser's window in Figure 4.

Figure 4: Running scripts from the wiki

Finally, the middle segment in the sidebar in Figure 3 is associated to context-dependent actions and modifiers. Note that in 'record' mode, the buttons in the area are 'save' and 'save as', whereas in 'run' mode, they are 'step' and 'stop'. Only pulse buttons are used, to facilitate interpretation, communication and learning. 'Step' might be designed as a sub-mode of 'run', but in addition to the semiotic and cognitive complexity associated with this choice, more buttons would be required to control the triggering of each step. More controls also mean more space required, which could eventually cause some of the buttons to disappear from the interface – a strategy that was adopted in SIL and may cause confusion for novices. Therefore the pulse button choice for stop and step is a more promising choice.

Notice that SIL* does not require more space than SIL. A comparison between Figures 3 and 4 shows that both designs use six button slots. SIL* expresses the gist of the designers' messages (see (a)-(j) in 3.1) with important communicative advantages over the design in SIL, which have the potential to improve learnability of the sidebar interface language. The analytic criteria supporting this potential refer to the properties of signs, themselves, and the properties of sign systems. The sign properties of inflection and distribution have been systematically explored in SIL* to communicate contextual cues in a concise manner. Additionally, regularities of form within a context-sensitive grammar – sign system properties – have been maximized to help designers communicate (and teach) the interface language that users must acquire to interact successfully with CoScripter

With respect to these analytic criteria, we can state that SIL*'s design is *analytically* superior to SIL's in terms of communicability and usability. However, this analysis is decidedly not *empirical*. While only user studies can empirically verify that SIL* actually improves the whole process of computer-mediated communication taking place in HCI, Semiotic Engineering adds a valuable tool for the design process by enabling designers to conclude analytically that one design has a simpler and more consistent signification system than another.

4. CONCLUSION

This paper reports on work in progress, and we are focusing exclusively on a specific component of the CoScripter interface – the sidebar. The whole interface actually involves other languages, such as the scripting language, the wiki interface language, the browser interface language (especially in playback mode, when recorded actions are signified in the browser's interface), and natural language (present in a considerable extent of web material and throughout the CoScripter wiki and users' forum). Nevertheless, even in this constrained context, we demonstrate Semiotic Engineering concepts and show how they can concretely impact design decisions.

The Semiotic Engineering approach to redesigning SIL is not meant to be taken as a design *method.* There are two main reasons for this. First, new design methods require extensive and laborious field research from which we can derive the recurrent conceptual and procedural steps that constitute the method's original contribution to design practice, and that distinguish it from other existing design methods. Second, this theory *can* clearly be used in combination with a well-known design approach and associated methods – Schön's *reflection in action* [11].

Schön's approach stresses the fact that every design is a unique problem, whose first solution step is actually to name and frame the elements that, according to the designer's interpretation, are part of the unique situation that they are dealing with. This view is in sharp contrast to other design traditions in which, for example, design problems are seen as instances of general problem types that can be solved by a systematic application of pre-established problem-solving procedures [12].

When we look at the theoretical roots of Semiotic Engineering, it is clear that viewing *meaning* as a constantly evolving sign-producing interpretive process is at odds with the idea that design methods can be used to *ensure* that the product will be interpreted in one way or another. Thus, Semiotic Engineering should not be used to *predict* how users will interpret the designers' message. In fact, there is no theory that can do this.

The advantage of using Semiotic Engineering in combination with *reflection in action* is that both emphasize the role and the value of knowledge generation in the design process. Schön believes that the most important requirement in design is what he calls 'an epistemology of practice'. By this he means a set of practical tools to help designers produce and evaluate design *knowledge* ('episteme' in Greek) that continuously arises and intervenes in the design process. Semiotic analysis and concepts help designers organize this knowledge according to communicative and interpretive categories, and to explicitly formulate what they *mean by* (*i.e.* how they name and how they frame) design

elements in an attempt to bring up certain features and effects when the design product is used.

Because Semiotic Engineering has a clear-cut characterization of HCI (as a designer-to-user communication about how to communicate with a system in order to achieve a certain range of effects) and provides an ontology for analyzing the elements involved in this process (*e.g.* sign classes, signification systems, and communication processes) it can boost the advantages of adopting Schön's perspective in design. Conversely, Semiotic Engineering has a great impedance with design methods that are expected to *predict* how users will react to the presence of certain signs in the interface, and with those that seek to generate universally applicable solutions.

Nevertheless, even in the absence of predictive statements, Semiotic Engineering can be used to improve design choices in important ways. In section 3.2 we have shown how knowledge about sign classes and properties of signification systems helps designers understand and (re)formulate the role and the function of interface languages as communication tools that are crucially important to achieve even the most traditional usability goals (like ease of learning and ease of use, for example).

In talking about Semiotic Engineering, we have mentioned certain communicative strategies, such as elaboration and redundancy, that communicators can effectively use to bring about the desired interpretations in the interpreters' minds. Throughout the paper we have provided instances of the use of redundancy in designing and redesigning SIL, but not of elaboration. Elaboration requires further communication, and given the spatial constraints of CoScripter, elaboration *is* a major challenge for the designers. SIL's hybrid signs (a composite of visual and textual material), as well as the tool tips that users view when they position the mouse on certain interface elements, are instances of elaboration. Signs of a given signification system are being *elaborated* (explained or described) by signs of another. The signs in the first system are invitations to the user to explore the interface and learn how to use it. However, this is only a timid attempt at meta-communication – communication *about* communication. So, one of the items in our future work agenda is to explore the design space for elaboration messages, in the form of online help, explanatory question-answering, parallel signification systems, and the like.

Another important step is to include other CoScripter interface languages in our analysis and redesign efforts. Among such languages the main ones are: the scripting language, in which recorded steps are codified in the sidebar, and which users can edit to introduce or modify certain steps in execution; and the playback preview language, in which CoScripter signals the interactive steps that are being performed in the browser's interface as users *run* a recorded script. Together, the scripting language, the playback language, the sidebar interface language and even natural language (used to elaborate on the messages conveyed by all of the other languages) must be consistent and cohesive with each other. Such is the complex semiotic engineering challenge of a relatively small system like CoScripter, which we have used to demonstrate and improve this new theory of HCI.

5. ACKNOWLEDGMENTS

The authors thank Laura Haas for establishing the Visiting Researcher Program at IBM Almaden Research Center, which provided the means to carry on the studies reported in this paper. Clarisse de Souza thanks CNPq for sustained support of her research, and her graduate students at PUC-Rio for providing numerous examples of interface challenges in CoScripter.

6. REFERENCES

[1] Blackwell, A. F. (2006) The reification of metaphor as a design tool. *ACM Transactions on Computer-Human Interaction* 13, 4, 490-530. DOI= http://doi.acm.org/10.1145/1188816.1188820

[2] Chomsky, N. (1959). "On certain formal properties of grammars". *Information and Contro*l (2): 137-167.

[3] de Souza, C. S. (2005) *The semiotic engineering of human-computer interaction.* Cambridge, Mass. The MIT Press.

[4] de Souza, C. S. and Barbosa, S. D. J. (2006) A semiotic framing for end-user development. In: Henry Lieberman; Fabio Paternò; Volker Wulf. (Org.). *End User Development: Empowering people to flexibly employ Advanced Information and Communication Technology.* New York: Springer. pp. 401-426

[5] de Souza, C. S., Leitão, C. F., Prates, R. O., and da Silva, E. J. (2006). The semiotic inspection method. In *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems* (Natal, RN, Brazil, November 19 - 22, 2006). IHC '06. ACM, New York, NY, 148-157. DOI= http://doi.acm.org/10.1145/1298023.1298044

[6] Eco, U. (1976) *A Theory of Semiotics*. Bloomington, IN. Indiana University Press.

[7] Hollan, J., Hutchins, E., and Kirsh, D. (2000) Distributed cognition: toward a new foundation for human-computer interaction research. ACM Trans. Comput.-Hum. Interact. 7, 2, 174-196. DOI= http://doi.acm.org/10.1145/353485.353487

[8] Nardi, B. A. (1996) *Context and Consciousness: Activity Theory and Human-Computer Interaction.* Cambridge, Mass. The MIT Press.

[9] Norman, D. A. (2007) *The design of future things.* New York, NY. Basic Books.

[10] Peirce, C. S. (1992, 1998) *The essential Peirce: Selected Philosophical Writings.* Vols. I , II. N. Houser and C. J. W. Kloesel (Eds.). Bloomington, IN. Indiana University Press

[11] Schön, D. (1983) *The Reflective Practitioner, How Professionals Think in Action*. New York, NY. Basic Books.

[12] Simon, H. A. (1996) *The Sciences of the Artificial (3rd Ed.).* Cambridge, Mass. The MIT Press.

[13] Stenning and Oberlander (1995) A Cognitive Theory of Graphical and Linguistic Reasoning: Logic and Implementation. *Cognitive Science*, 19, 1, pp. 97-140.

Affective Geographies: Toward A Richer Cartographic Semantics for the Geospatial Web

Elisa Giaccardi
Department of Computer Science
University of Colorado
Boulder, Colorado, USA
+1 303 492 4147
elisa.giaccardi@colorado.edu

Daniela Fogli
Dipartimento di Elettronica per l'Automazione
Università di Brescia
Brescia, Italy
+39 030 3715 666
fogli@ing.unibs.it

ABSTRACT
Due to the increasing sophistication in web technologies, maps can easily be created, modified, and shared. This possibility has popularized the power of maps by enabling people to add and share cartographic content, giving rise to the geospatial web. People are increasingly using web maps to connect with each other and with the urban and natural environment in ways no one had predicted. As a result, web maps are growing into a venue in which knowledge and meanings can be traced and visualized. However, the cartographic semantics of current web mapping services are not designed to elicit and visualize what we call affective meaning. Contributing a new perspective for the geospatial web, the authors argue for affective geographies capable of allowing richer and multiple readings of the same territory. This paper illustrates the cartographic semantics developed by the authors and discusses it through a case study in natural heritage interpretation.

Categories and Subject Descriptors
H.5.m [**Information interfaces and presentation (e.g., HCI)**]: *Miscellaneous*

General Terms
Design, experimentation, human factors

Keywords
Collaborative web mapping, information visualization, map-based interaction, web cartography

1. INTRODUCTION
The possibility to "read-and-write" online maps has given rise to the geospatial web. Web services such as Google Maps have popularized and democratized the power of maps by enabling people to add and share cartographic content. People are now irreversibly cutting their bonds to the desktop and using computing to connect with each other and with their urban and natural environment in ways no one had predicted [23].

Maps are increasingly the venue where knowledge and meanings can be traced and visualized. Current web mapping services provide features for users to contribute location-based content. However, their cartographic semantics are not designed to elicit and visualize what we call "affective meaning." By affective meaning we refer to the perceptions, interpretations, and expectations one ascribes to a specific physical and social setting ("affective" in the sense of showing how we are "affected" by environmental settings, and in turn "affecting" the way in which we experience and interpret the mapped environmental settings). The cartographic semantics of current web mapping services show the where and when of information, but they do not visually relate that information to one's perceptions, interpretations, and expectations—they are not designed to show the personal meaning that one ascribes to specific locations. In a society where "computing means connecting" [23], being able to capture and visualize affective meaning is vital to enhance our perception of space, deepen our connections with the urban and natural environment, and stimulate reflection and discussion about the places in which we live and that we share.

We believe that the future of the geospatial web requires richer cartographic semantics, and we propose the idea of "affective geographies." By "affective geographies," we mean the digital representation of space and place that is enabled by cartographic semantics capable to elicit and visualize affective meaning in collaborative web maps. Affective geographies can be powerful tools to link experience, interpretation, and management of the places in which we live and that we share by allowing us to visualize what really matters personally.

Opening up new perspectives for the geospatial web, the paper aims to frame the idea of affective geographies and illustrates the cartographic semantics the authors have developed to elicit and visualize affective meaning in collaborative web maps. The paper discusses the usage and impact of the cartographic semantics used through a case study in natural heritage interpretation and preservation.

2. RELATED WORK
The term "geospatial web" or "geoweb" has been coined to denote a new infrastructural paradigm to access and explore data on the web—one that permits users "to navigate, access, and

AVI'08, 28-30 May , 2008, Napoli, Italy

visualize georeferenced data as they would in a physical world" [19]. With computers bifurcating database and visualization [1], the geospatial web is offering directly to users the possibility to easily create, modify, and share online maps. Google Maps [15] and Google Earth [14], for example, enable users to create personalized 2D and 3D maps and share them with relatives and friends. With Google Maps, users can create their own maps by using place markers, shapes, and lines to define a location, an entire area, or a path. Cartographic content can then be annotated with text, photographs, and videos. Furthermore, through Google Maps API, several mashups have been created to provide map-based representations using the same Google Maps' cartographic semantics of place markers, shapes, and lines. One example is Chicago Crime [7], which visualizes information about crime concentrations in the Chicago area. Another example is Hurricane Digital Memory Bank [17], which allows users to contribute and share their stories on the aftermath of the hurricanes Katrina and Rita, and visualize the location of such contributions on the map.

The possibilities offered by the geospatial web in conjunction with mobile computing are also inspiring new metaphors for collaborative mapping and the description of experiences in geographic spaces. For example, mobility data [21] and sensor data [4] are increasingly used to obtain different kinds of "geovisualizations." Artists and researchers are using these data to visualize information flows and to display, for example, the real-time dynamics of pollution phenomena, or users' galvanic skin response in conjunction with specific geographical locations [3], or users' personal routes within the city [2]. In particular, participatory approaches [4][13] emphasize the role of users as knowledge authors and stress the importance of easily enabling them to intentionally gather, analyze, and share location-based knowledge. According to Girardin and colleagues [13], uploading, tagging, and disclosing location-based information can be interpreted as an act of communication rather than a purely implicit history of physical presence. Their goal is to use explicitly disclosed location information to enrich the quantitative understanding of the city that is provided by the spatio-temporal patterns of mobility data (i.e., latitude, longitude, and timestamp). In general, the assumption is that "geovisualizations" based on mobility data or sensor data support social navigation, in that people's past interactions with the environment can be read as "recommendations," and may impact others' behaviors within the same space. However, even when users disclose location-based information explicitly, the communicative function of such information has to be extrapolated from the map exclusively through the visualization of its spatio-temporal patterns.

Other projects, such as Social Tapestries [24][18], promote a stronger participatory approach to data collection, exploring the potential benefits and costs of collaborative web maps generated by means of public authoring systems. Framed within map-based community practices revolving around ideas of place and identity, these systems enable community members to participate and contribute their experiences. In Social Tapestries, knowledge mapping and sharing is pursued through various themes, community interventions, and contributed cartographic content. However, once again, even participatory approaches appear to be lacking an investigation of how web mapping and visualization may support qualitative readings and foster reflection and discussion. The Snout map of London [Figure 1], for example, is a collaborative web map created as part of the Social Tapestries project. It visualizes air pollution data collected by the community through different kinds of environmental wearable sensors. The map combines markers, colors, and text to represent different types of air pollution. For each data point collected, the map provides only location and taxonomy (i.e., carbon dioxide versus organic solvent vapor), using balloons that are difficult to read and interpret.

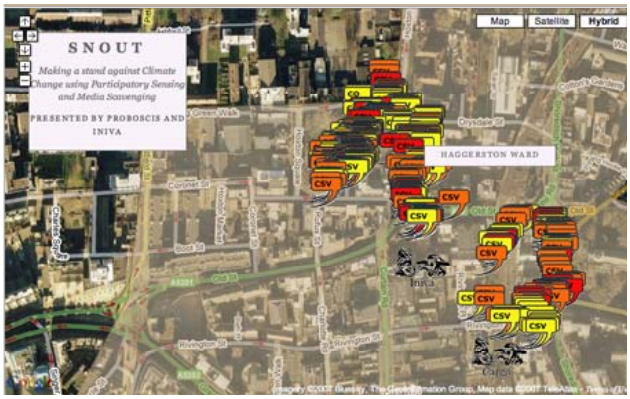


Figure 1 The Snout map (http://socialtapestries.net/snout/).

Independent of the methodology used to collect and contribute cartographic content (whether "sensed" or "user-generated"), we argue that the main limit of current approaches to web mapping and visualization is in the cartographic semantics. It displays only where a specific information is located and, even when enriched by users' comments, photographs, or other multimedia content, it does not visually conveys individual meanings or "social moods" [6] at first glance.

3. AFFECTIVE GEOGRAPHIES OF SPACE AND PLACE

Affective geographies, then, are maps that elicit and visualize "affective meaning"—the perceptions, interpretations, and expectations one ascribes to a specific topological and social setting. We call these "affective" because they reveal how we are "affected" by environmental settings, and in turn "affect" the way in which we experience and interpret the environmental settings mapped. They are maps that display cartographic content contributed by users as well as their personal readings of such content.

3.1 Mapping Space

The widespread tendency in the last centuries of European history has been to think of space and place in terms that reduce space to topological construct and place to mere location or position within an extended space [5]. In this way of thinking, space turns out to be a given entity, and place a somewhat arbitrary or constructed notion. Place is often considered to be identical with either the position of a body in space or with an area of the kind that can be identified by using physical markers in a space (e.g., in the geospatial web, pushpin-like place markers). An alternative tendency has been to think of place as a "significant locale," that is, as a space to which human meaning is attached.

Harrison and Dourish [16] initially distinguished two aspects of spatially organized environments: "space" is concerned with those material and geometrical properties (such as relational orientation, proximity, partitioning) that enable certain forms of movement and

interaction; "place" has to do with the ways in which human activity and social practices can occur within a space. Ten years later, Dourish revised this view by arguing that space should be seen as "a social product just as much as place" [8]. Grounding his position in anthropology and human geography, he argues that geography is the product of a particular kind of social practice that gives us an account of space.

In line with this later approach, affective geographies allow users to visually define space by enabling them to choose what to map according to their own knowledge and practices. Through contributed cartographic content, the resulting geography provides a living account of space as a social product of individual embedded knowledge, daily practices, and concerns.

3.2 Giving Meaning to Place

However, "there is no there there" [25], as Gertrude Stein would say, to be visualized. In other words, contributed cartographic content alone does not allow visualizing the personal meaning associated to the mapped territory. Such a map would not refer us to any particular site or locale that has a special significance. As a result, the space is mapped, but no meaningful context is visually provided for that mapping. A "sense of place" is missing from the map in that there is neither a sense of the character or identity that belongs to certain places or locales, nor a sense of our own identity as shaped in relation to those places [20]. We might say that although the space represented is the account of individual knowledge, practices, and concerns, no sense of place is represented because no account is *visually* provided for such knowledge, practices, and concerns. What would be visible on the map is only the "where," or at best the "when," of information.

The cartographic semantics of affective geographies provide a visual account of how space and place relate to each other. Visual mechanisms are provided so that users' actions (e.g., their own particular decisions about collecting and annotating cartographic content) not only stimulate reflection on personal experience, but also encourage reflection about others' experiences that may in turn inform subsequent action.

Mobile devices and "technologies of spatiality," such as global positioning system (GPS) tools and maps, can create new opportunities for social interaction and help people remember and "re-encounter" everyday space [8]. By weaving affective meaning in geospatial mapping and visualization, affective geographies provide a new way of thinking and exploring the social relationship between space and place. Their role can then be understood as one of assisting in the remembering, reconstructing, and representing our geography of space and sense of place.

In summary, affective geographies enable users to define space by choosing what to map, and at the same time to give meaning to the place by providing a personal reading of the mapped territory. Mapping and visualization in affective geographies reveal individual emotions, concerns, and values. But in order to foster reflection and discussion, it is fundamental for affective geographies to visually correlate these unique perceptions with places to which people collectively ascribe a similar meaning and spaces that people map and recognize as belonging to the same territory but without strong feelings or expectations about them. Affective geographies must reveal individual as well as collective patterns of perception and interpretation in relation to the same territory. Only in this way can they display aspects of the environment that lie beyond our usual perception and allow multiple readings of the same territory.

Making the geospatial web a richer tool means revealing and eliciting the affective meaning that is associated to a mapped territory. This requires affective geographies that evolve according to the social perception and interpretation of the individual meanings and values that one ascribes to specific topological and social settings. It requires maps that can be easily created and modified by users, so users not only contribute information but also are able to annotate and qualify this information by expressing their feelings and concerns in relation to it. Affective geographies enable users to easily visualize and read on the map the compound of cartographic content and affective meaning that defines one's geography of space and sense of place.

4. WEAVING AFFECTIVE MEANING IN GEOSPATIAL MAPPING AND VISUALIZATION

Affective geographies are enabled by cartographic semantics that elicit and visualize affective meaning. To this end, we have developed a cartographic semantics based on the principles of the Abaque de Régnier method.

4.1 The Abaque de Régnier Method

The Abaque de Régnier [22] is a method used today in areas as diverse as human resources, regional planning, and sustainable development to help people express themselves and build shared understanding. The method uses a color-coded scale by which to provide answers to specific questions. The colors are suggestive of the traffic light, whose codification is the same in most countries (even though the same color may have different names). The colors are green, yellow, and red and, in addition, light green and light red. This scale moves from the most favorable position (green) to the most unfavorable (red). Additionally, white and black are also used to indicate that the respondent does not have any opinion (white) or refuses to answer (black). According to the method's terminology, colors are called "transparencies" and white and black are called "opacities." By combining logical and statistical representation, the method converts colors into numerical values and produces visual matrices where raw data are permutated.

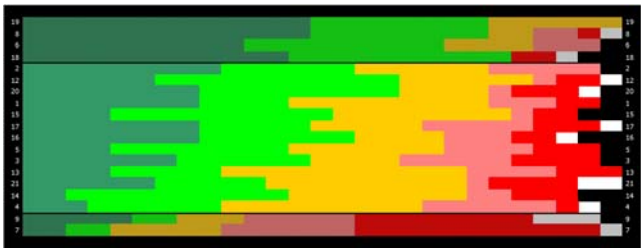


Figure 2 Abaque de Régnier: Visualization of favorable agreement (highlighted, top) and unfavorable agreement (highlighted, bottom).

Tendencies toward "favorable agreement" are located at the top of the matrix (majority of greens), and tendencies toward "unfavorable agreement" are located at the bottom (majority of reds). Problems are located in the middle where there is a significant diversity of colors ("disagreement"). "Areas of

uncertainty" are revealed by the cross section of yellows, and weighted according to its width [Figure 2]. "Anomalous positions" (e.g., isolated red dots or isolated green dots) are revealed by further permutation [Figure 3].

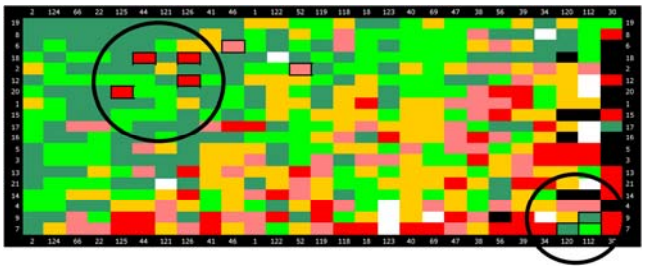


Figure 3 Abaque de Régnier: Visualization of anomalous positions (highlighted).

Representation of values by colors is immediately recognized, and the color-coded scale enables an exploration of subjective perception at three different levels: local, regional, and global. The individual (local) level is represented by the cell at the intersection of a column and a row. It shows the opinion an individual holds about an item. Columns or rows represent the regional level and show the overall positions of all participants on a single item or of a single individual on all the items. The global level is represented by all the colored positions on all the items, and is expressed by the whole matrix. Evaluations and varied adoptions of the method over the past 30 years have demonstrated that this visualization successfully elicits reflection and discussion. In particular, recursive cycles of data collection, visualization, and discussion have proven successful in providing mutual understanding within groups sharing the same problems.

4.2 Toward a Cartographic Semantics for Affective Geographies

The strength of the Régnier method is its ability to map subjective perceptions by means of colors, visualize patterns of judgment at different levels, and visually foster reflection and discussion.

We have translated Abaque de Régnier principles to collaborative web mapping and adapted them to provide a cartographic display of individual meanings and social relations that might provoke reflection and discussion about the places where we live and those that we share. We have created a visual notation for location-based information based on the Régnier colors: dark green, light green, yellow, light red, and dark red. White is used exclusively to indicate content not yet annotated and therefore not public. On our map, colored dots serve the function of both locating information and representing the affective meaning one has ascribed to that information. The dots visually compose an affective geography that defines both the mapped space and its subjective quality as place. Color dots assume different sizes according to the zoom level: the higher the zoom level, the smaller the size of the dot (from a cloud of small dots to the view of a clickable single dot). At a bird's-eye view, clouds of dots identify color patterns and visualize areas of positive and negative agreement, dissension, uncertainty, and anomalous positions [Figure 4]. These patterns can be correlated with the characteristics of a specific location, and show which places elicit strong feelings (prevalence of red or green dots) or no particular expectations (prevalence of yellow dots). Patterns can be explored at different levels, from the local level of the individual user (single dot) to the global level of the community (clusters of dots).

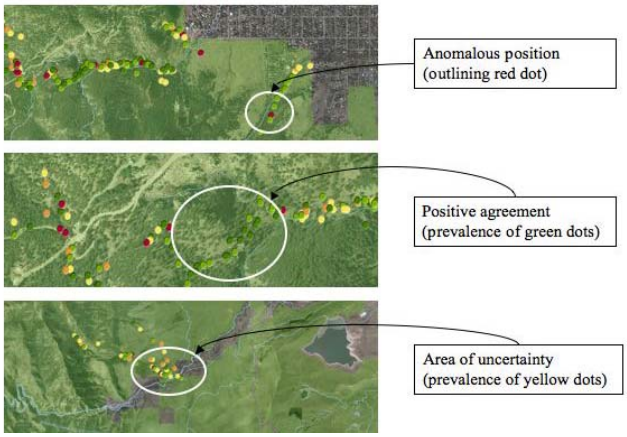


Figure 4 Color patterns in the cartographic semantics.

Our cartographic semantics combine this visual notation with the capability to preview cartographic content (such as an image or a sound) by mousing over the dot, and to access additional verbal descriptors (such as associated tags and a personal journal) by clicking on the dot.

The web application we have created [Figure 5] enables users to choose and collect cartographic content through mobile devices. GPS data locate this content in space and time on an interactive map created through an open source Geographic Information System (GIS). Once cartographic content is uploaded, users can access, manage, and interpret it by visually associating Régnier colors and annotating it with tags and narratives. In this way, cartographic content as well as users' personal interpretations become publicly available at the immediate level of the visual notation and then incrementally through map-based interaction.

More specifically, the web application provides an Edit mode and an Explore mode. In the Edit mode, registered users can privately visualize on the map their collected content and distinguish between content they have already annotated (colored dots) and content they have not yet annotated (white dots). Mousing over the dot enables users to preview the content, whereas clicking on the dot selects the content and automatically opens the window, enabling color rating and textual annotation. The Explore mode allow both registered and unregistered users to navigate the map and filter the cartographic output according to several criteria such as color, and tags of interest.

Through the web application, users create and share cartographic content that visualizes their daily practices and personal perspectives. The resulting affective geography provides multiple readings of the same territory at different levels: from the local level of the individual (single content and single color), to the regional level of social patterns (local clusters of content and colors), to the global level of the community (overall trends of content and colors). These readings, in turn, can be conducted at the local level of a specific site, the regional level of a specific topological area, or the global level of a community's self-defined geography. Additionally, the different filtering capabilities provided by the web application allow the user to define and operate "permutations" of the cartographic content and multiply such readings according to the user's specific interests.

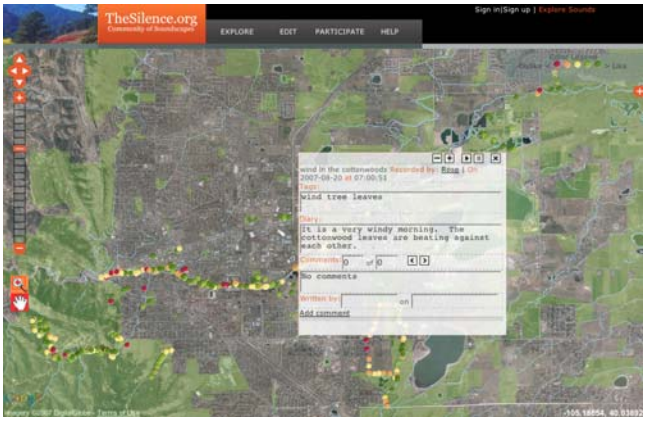


Figure 5 The collaborative web mapping application (http://thesilence.f-dat.org/).

In summary, we have developed a cartographic semantics for collaborative web mapping by which visual notation provides an immediate visualization of both individual and collective affective meanings, while the content to which meanings are associated is provided incrementally through map-based interaction. Mousing over the dot enables the user to preview the cartographic content, and clicking on the dot provides more detailed information about both content and meaning. In this way, the cartographic semantics proposed smoothly overlays location, meaning, and content, starting from an immediate and intuitive visualization.

5. CASE STUDY: COMMUNITY OF SOUNDSCAPES—TOWARD AN AFFECTIVE GEOGRAPHY OF SILENCE

Community of Soundscapes is part of a long-term project called "The Silence of the Lands," a socio-technical environment using sounds to raise environmental awareness and promote the active and constructive role of local communities in the interpretation and management of their urban and natural environment [9][10][11]. The project was initiated by Giaccardi at the University of Colorado, Boulder, in 2005, and currently involves an international collaboration among the CU-Boulder's Center for LifeLong Learning & Design (USA), the University of Brescia (IT), and the University of Plymouth's Institute of Digital Art and Technology (UK). Based on the belief that sounds are an important and personal element of the natural environment, the project's goal is to encourage a focused and engaged way of "listening to the land." In doing so, the project sustains a narrative mode of social production of natural heritage aimed at fostering environmental awareness and eventually supporting new forms of sustainable development.

This goal is accomplished by allowing people to capture and map their sonic experiences and then annotate and share the soundscape of the environmental settings where the sounds were recorded. Users record sounds by using a mobile device outfitted with GPS mapping hardware and software, called Sound Camera. Recorded sounds are then uploaded on the web through the collaborative mapping application, where they are associated with their owner and placed on the map [11]. Geographic position, time, and date are entered automatically. Then, through the web application, users are able to add and share descriptions of the sounds they heard, indicate by means of colors whether they liked or disliked those sounds, and comment on other people's sounds. The result is an "affective geography of silence," as we call it, where understandings and encounters with space and place evolve according to how users' experiences and interpretations of the sonic environment are mapped, visualized, and in this case audio-streamed in the form of an interactive soundscape.

5.1 Pilot Study

In collaboration with the City of Boulder Open Space and Mountain Parks Department and Water Quality Department, we engaged the local community of Boulder, Colorado, in capturing and sharing sonic experiences for a period of six weeks. Contextualized within the City of Boulder nature programs and public hikes, *Community of Soundscapes* enrolled a group of community members representative of different age populations and professional backgrounds [Figure 6]. From July 2007 to September 2007, participants engaged in sound walks and workshops, mapping and sharing more than 1300 sounds [12].

Figure 6 Participants from the Boulder community using the Sound Camera (CU Photography/Larry Harwood).

The goal of the pilot was to evaluate the use, impact, and real-world application of our sound mapping technology in the context of natural heritage interpretation and preservation. Because sounds hold a strong affective meaning in relation to our experience of space and place, we were interested in investigating how sound mapping can encourage people to reflect on their perception and interpretation of the environment, facilitate looking at each other's experiences and connecting with each other's perceptions, and finally help unfold new understanding of the environmental settings in which people live and that they share. The findings presented in this paper offer data relevant to evaluate aspects of the cartographic semantics proposed and discuss implications of the evaluation to the visualization strategy.

5.2 Methodology

A sample of 20 volunteers (4 males and 16 females) participated in the pilot study. Their ages ranged from 20 to 62 years. They all held a higher education degree and represented varied professional backgrounds. They included writers, engineers, scientists, managers, designers, educators, therapists, musicians, and college students.

Participants were asked to capture their sonic experiences by using the Sound Camera and to upload sounds on the web application, where they could annotate them and share them with other participants. They were asked to take at least three sound walks: one on Flagstaff Mountain, one along the Boulder Creek Path, and a third one of their choosing. A total of 1338 sounds

were recorded by using the Sound Camera, and 567 sounds were selected and made available on the web application.

We triangulated qualitative data collected through: (a) three focus groups at the beginning of each organized workshop, (b) two questionnaires (a pre-questionnaire and post-questionnaire), (c) unstructured interviews and direct observations conducted during participants' activities, and (d) participant's narratives associated with sounds. Quantitative data derived from database queries and web analytics have not yet been integrated in the evaluation.

5.3 Findings

5.3.1 Expressing Affective Meaning

The first theme that emerges from the data concerns whether people felt able to express affective meaning by means of the cartographic semantics designed for the web application. One question in the post-questionnaire explicitly asked: "Did you feel able to express and share your perceptions and values through the technology provided? Can you give an example?"

Answers to this question, corroborated in the focus groups and by our observations as well, indicated that participants felt able to express and share their perceptions and values through the technology provided, in particular by being able to "rate" a sound (as they often referred to the use of Régnier colors). One participant answered:

"I never liked the sound of small aircraft that seem so prevalent in Boulder, and especially when I go on a walk or hike. When I recorded these sounds and was able to rate them, I was able to convey my strong dislike of these sounds."

More clearly, another participant explained:

"Yes [I felt able to express and share my perceptions and values, authors' contextualization*]. First of all through the choices of what to record and keep in the web application. Second through descriptions of sounds and comments on sounds of others."*

Another participant wrote:

*"Rating sounds make me think about *good* sounds vs. noise + how it differs for me depending on my mood."*

These and other similar answers give us material to sustain that being able to annotate sounds and particularly "rate" them through Régnier colors seemed to enable and encourage participants to reflect on their own experiences and to express their impressions and interpretations of the space encountered during designated hikes (i.e., Flagstaff Mountain and Boulder Creek Path) or their daily practices (for locations of their own choosing).

This is confirmed by some of the narratives provided to annotate sounds. One participant, for example, "rated" the sound of small aircraft as a pleasant sound:

"There are always airplane sounds at Sawhill Ponds. Right now there are two overhead. One is a cute little red bi-plane."

Contrary to the reaction of the participant who generally dislikes the sound of small aircraft, this participant expresses and reveals a different set of experiences in relation to the expected identity of a familiar location. Interestingly, unfavorable patterns of judgment (red dots) toward the sound of aircraft appear in locations expected to be pristine (e.g., Flagstaff Mountain), whereas anomalous positions, such as the one recorded at Sawhill Ponds, appear in locations whose identity is more strongly tied to an individual's personal experiences. Anomalous positions visualize occasional events and users' idiosyncrasies (including moods) with respect to one's unique experience of a specific place. This information is visualized and easily singled out at the global level, and has proven to be a particularly useful strategy to stir curiosity and foster reflection in map-based interaction (see Sections 5.3.2 and 5.3.3).

5.3.2 Exploring Other People's Experiences

Another theme that emerges from the data concerns whether people felt able to explore and understand other people's experiences through the cartographic semantics.

Answers provided to the post-questionnaire's direct question: "Did you find it interesting to listen to other people's sounds? Can you give an example?" are particularly useful. Generally speaking, participants appreciated the possibility of enjoying sounds collected by other participants. A couple of them, for example, commented:

"I loved the sounds from the Boulder Farmer's Market. I could listen to the sound and visualize the setting without being there."

"Yes, I liked hearing the more random sounds from crowds in downtown Boulder."

Other participants emphasized the differences in perceptions and interpretations that emerged within the community and stressed the enrichment they gained from these differences. For instance, some interesting answers to the same question include:

"I learned from their trials. For example, there are not many animal sounds in the heat of the day, I noticed, so I planned to 'walk' later in the day."

"I was curious about what others chose to record. Many were like my choices; some were totally different (a trash can lid)"

Once again, the adopted visualization strategy appeared useful to participants. Through colors, participants were able to notice differences, and in general find their own way through map exploration and interpretation. Participants demonstrated an awareness of the dependency between an individual, that individual's color "rating," and the context of the recording. Because of that they were curious about other people's sounds when looking at colors on the map; in particular, when looking at the extreme ones (dark reds and dark greens), typical answers to the question: "Did colors trigger specific behaviors in your exploration of the map? Can you give an example?" included:

"Dark green and red (both ends of spectrum) were ones I checked out first."

"I liked going through the red ones, to see what people classified as negative sounds."

Participants' differences stirred quite a lot of reflection and discussion also in the focus groups, keeping participation high, and motivating participants to more recordings.

5.3.3 Reading and Understanding the Map

With regard to the general visualization strategy, another relevant theme that emerged from the data concerns what aspects the cartographic semantics allowed participants to judge. The following answers provide a comprehensive account of how participants tended to use and read the map, and to what they paid more attention:

"Extremes (extreme likes/dislikes). I could also see places that I might like to visit (lots of dark green dots)."

"What areas were louder, more contaminated with traffic, and which were quieter."

"I noticed areas that I was supposed to visit, e.g., the east end of the Boulder Creek Path, that had sounds other people liked."

Overall, the colors painted a general impression of an area, guiding participants in their map-based interactions and explorations. As the pilot study suggests, colors also influenced participants' reflective processes, learning, and behaviors. What emerges from the data collected is that the adopted visualization strategy—based on the principles of the Régnier color schema—plays an important role in people's reading and understanding of the map, and also in supporting subsequent actions in the real world as a result of these readings. In the next section, we discuss such impacts.

6. DISCUSSION

The results of the pilot seem to suggest that the cartographic semantics proposed provide an effective mode of reflection and discussion about the individual and collective perceptions, interpretations, and expectations that relate to a specific location and its environmental setting. The resulting affective geography of *Community of Soundscapes* seems to produce a new mode of interaction with the environment and with other members of the community that is responsible for several perceived benefits. Based on participants' feedback, these benefits can be categorized as an enhanced perception of the environment, deepened social and environmental connections, increased environmental awareness and reflexivity, and behavioral change. We are aware that further studies are needed to reveal the co-dependency between the use of the cartographic semantics and sounds, and to help isolate the specific benefits and limits of the cartographic semantics. We discuss here our initial set of results.

6.1 Enhancing Perception

Enhanced perception of the environment seems to be the first and more immediate benefit perceived by participants:

"Nature sounds have always been a favorite background while I'm working, but now I'm also more curious of the outdoors and I want to trace sounds."

"I find myself saying 'that would be a cool sound to capture' such as a bird call, coyote howl. I'm also much more interested in the man-made sounds, such as the ding of the bus."

"[I am] more perceptive, or at least more open to listening for sounds—went on a night hike and sat and listened to intense duet between insects and the hum of the city—wished I had brought the Sound Camera."

6.2 Deepening Connections

Participants also reported a deepened connection to the environment and an increased sense of place. One participant, for example, explained:

"[I have] more appreciation for how rare it is to be away from human sounds. Also, it really made me feel bad for wild animals that have to deal with human sounds, must mess with their instincts."

A few participants asserted that sharing sonic experiences and listening to other people's sounds have somehow changed their sense of belonging to the community. One participant wrote:

"I do admire some of the sounds the other volunteers found. It is an interesting way to connect with others."

6.3 Fostering Reflection and Awareness

From participants' feedback emerged the feeling that the possibility of color rating and annotating sounds was an effective mechanism to provoke reflection and stimulate environmental awareness. Some participants perceived this benefit at the personal level. Some, for example, wrote:

"I have a greater awareness and appreciation for the ability of some sounds to have a negative affect on my mood."

"Increased awareness of sound. Enhanced experience of life."

For other participants, this awareness assumed a different scale. For example, one participant said:

"Awareness of other 'life' that we share space with—disappointment of not being able to escape man-made sound (i.e., cars). Even when you get far enough away, city noises and airplanes disrupt the natural sounds every few minutes."

6.4 Supporting Behavioral Change

Enhanced perception, deepened social and environmental connections, and increased environmental awareness and reflexivity seemed to encourage participants to spend more time in the outdoors and learn more about their environment and the community in which they live. Some participants, for example, wrote:

"I learned to pay attention and be aware of the sound environment. The main benefits were the immediate ones, going out and spending time in nature, and the longer-term awareness of the sounds around me."

"I'm more interested in learning to recognize specific bird calls. Also, I am more attentive to sounds, whereas before I mostly got lost in my mind while walking."

In general, participants perceived these benefits as so meaningful to them that the only limitations they reported concerned the usability and robustness of the system: they liked what they were doing and wanted to be able to do it faster and more reliably. Participants also suggested new features to be added to the application, such as the possibility of switching directly from the explore mode to the edit mode when accessing information related to their own recorded sounds. Despite technical limitations, though, the kind of experience and interaction provided by *Community of Soundscapes* motivated half of the participants to request continuation of the project over the entire year. To this end, the web application is being improved to both overcome the current technical limitations and provide participants with new interaction possibilities. New visualization strategies are also being discussed to allow users to manage multiple readings of the same location through the filtering mechanisms.

7. CONCLUSIONS

The research activities we have described here are motivated by the desire to address the need for affective geographies as a central issue for the geospatial web. We have defined affective geographies as web maps that reveal how we are "affected" by

environmental settings, and that in turn "affect" the way in which we experience and interpret the environmental setting mapped.

Other researchers have attempted to create geovisualizations of subjective content. Their maps, however, even when enriched with users' comments, photographs, or other multimedia content, appear difficult to read and hardly convey some kind of individual and/or social meaning at first glance.

Attention is shifting to these new concerns, due not in the least to increasing sophistication in web mapping technologies and mobile computing, and to the increasing role that web maps play as venues where knowledge and meanings can be traced and visualized. The goals, of course, are challenging. What this attention to web mapping and visualization as well as map-based interaction needs is additional design thinking about some of the core concerns presented here, including how to elicit and visualize the social system of experiences, interpretations, and expectations that contribute to one's geography of space and sense of place.

We have argued that, by weaving affective meaning in geospatial mapping and visualization, affective geographies provide a new way of thinking and exploring the social relationship between space and place: they enable users to define space by choosing what to map, and at the same time to give meaning to place by providing a personal reading of the mapped territory.

We have proposed a cartographic semantics for affective geographies capable of providing immediate and spontaneous readings of the same territory at multiple levels (local, regional, and global), and we have illustrated its viability through a case study. Initial positive results suggest that the proposed cartographic semantics foster reflection, discussion, and behavioral change: users' actions (e.g., their own particular decisions about collecting and annotating cartographic content by means of the semantics provided) not only stimulate reflection on personal experience, but also encourage reflection about others' experiences that may in turn inform subsequent action.

8. ACKNOWLEDGMENTS

The authors thank: Ilaria Gelsomini, Francesca Pedrazzi, Guido Pollini, Gianluca Sabena, and Chris Speed for their contributions to the development of the collaborative web mapping application. This research was supported by: (1) the National Science Foundation grant IIS-0613638, (2) the CU-Boulder's Outreach Committee grant 2006-2007, (3) the Università degli Studi di Brescia, and (4) the Arts Council England.

9. REFERENCES

[1] Abrams, J., and Hall, P. (eds) *Else/Where: Mapping—New Cartographies of Networks and Territories*, Minneapolis: University of Minnesota, 2006.

[2] *Amsterdam RealTime*, http://realtime.waag.org/.

[3] *Bio Mapping*, http://www.biomapping.net/.

[4] Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., and Srivastava, M.B., Participatory Sensing, paper presented at *ACM Sensys 2006*, Boulder, Colorado, 2006.

[5] Casey, E. S. *The Fate of Place*, Berkeley: University of California Press, 1997.

[6] Celentano, A., Mussio, P., and Pittarello, F. The Map is the Net—Towards a Geography of Social Relationships, *Workshop on Map Based Interaction in Social Networks (MapsInNet07)*, INTERACT 2007, Rio de Janeiro, Brazil.

[7] *Chicago Crime*, http://www.chicagocrime.org

[8] Dourish, P. Re-Space-ing Place: "Place" and "Space" Ten Years On, *Proc. CSCW'06*, New York: ACM Press, 299-308.

[9] Giaccardi, E., Cross-media Interaction for the Virtual Museum: Reconnecting to Natural Heritage in Boulder, Colorado. In Y. Kalay, T. Kvan, and J. Affleck (eds.), *New Heritage: New Media and Cultural Heritage*, London: Routledge, 2007, 112-131.

[10] Giaccardi, E., and Palen, L. The Social Production of Heritage through Cross-Media Interaction: Making Place for Place-Making, *International Journal of Heritage Studies*, 2008, 14(3), in press.

[11] Giaccardi, E., Fogli, D., Gelsomini, I., Pedrazzi, F., Sabena, G., and Speed, C. Acoustic Cartographies: Supporting Interpretative Experience of the Natural Heritage through Collaborative Mapping, *Proc. ICA2007*, Madrid, Spain, September 2007 (CD-ROM).

[12] Giaccardi, E., Freeston, J., and Matlock, D. *Community of Soundscapes: Results and Evaluation*, Internal Report, University of Colorado at Boulder, September 2007.

[13] Girardin, F., Blat, J., and Nova, N. Tracing the Visitor's Eye: Using Explicitly Disclosed Location Information for Urban Analysis, *IEEE Pervasive Computing*, 2007, 6(3): 55.

[14] *Google Earth*, http://earth.google.com/.

[15] *Google Maps*, http://maps.google.com/.

[16] Harrison, S., and Dourish, P. Re-Place-ing Space: The Roles of Place and Space in Collaborative Systems, *Proc. CSCW'96*, New York: ACM Press, 67-76.

[17] *Hurricane Digital Memory Bank*, http://hurricanearchive.org/map/.

[18] Lane, G. Urban Tapestries: Wireless Networking, Public Authoring and Social Knowledge, *Personal and Ubiquitous Computing*, 2003, 7(3-4): 169-175.

[19] Leclerc, Y. G., Reddy, M., Iverson, L., and Eriksen, M. The GeoWeb—A New Paradigm for Finding Data on the Web, *Proc. ICC2001*, Beijing, China, August 2001.

[20] Malpas, J. *Place and Experience*, Cambridge, UK: Cambridge University Press, 1999.

[21] Ratti C., Pulselli R. M., Williams S., and Frenchman D. Mobile Landscapes: Using Location Data from Cell-Phones for Urban Analysis, *Environment and Planning B: Planning and Design*, 2006, 33(5): 727–748.

[22] Régnier, F. *L'Enterprise Annonce la Couleur: Gerer les Divergences Leviers d'Efficacite Creatrice*. Paris: Les Editions d'Organisation, 1984.

[23] Roush, W. Social Machines: Computing Means Connecting, *Technology Review*, August 2005, 108(8): 44.

[24] *Social Tapestries*, http://socialtapestries.net/.

[25] Stein, G. *Everybody's Autobiography*. New York: Random House, 1937, p. 289.

Recognition and Processing of Hand-Drawn Diagrams Using Syntactic and Semantic Analysis

Florian Brieler
florian.brieler@unibw.de

Mark Minas
mark.minas@unibw.de

Institute for Software Technology
Computer Science Department
Universität der Bundeswehr München
85577 Neubiberg, Germany

ABSTRACT

We present an approach to the processing of hand-drawn diagrams. Hand drawing is inherently imprecise; we rely on syntactical and semantical analysis to resolve the inevitable ambiguities arising from this impreciseness. Based on the specification of a diagram language (containing aspects like concrete and abstract syntax, grammar rules for a parser, and attributes for semantics), editors supporting free hand drawing are generated. Since the generation process relies on the specifications only, our approach is fully generic. In this paper the overall architecture and concepts of our approach are explained and discussed. The user-drawn strokes (forming the diagram) are transformed into a number of independent models. The drawn components are recognized in these models, directed by the specification. Then the set of all components is analyzed to find the interpretation that best fits the whole diagram. We build upon *DiaGen*, a generic diagram editor generator enabling syntax and semantic analysis for diagrams, and extend it to support hand drawing. Case studies (done with a fully working implementation in Java) confirm the strength and applicability of our approach.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Sketching, Hand Drawing, DiaGen, Model, Ambiguity Resolution, Recognition

1. INTRODUCTION

Nowadays, GUIs have almost completely replaced textual user interfaces from which they have evolved. Still emerging in this field is the use of a pen or stylus as input device, which may often fully replace a mouse. This requires a paradigm shift, since using a stylus for input differs from using a mouse [16]; it feels more natural, and offers a different way to enter commands.

One interesting application area where this shift may take place are editors for diagrams (e.g., class diagrams and all other diagram types defined by the UML, Petri nets, or Nassi-Shneiderman diagrams). Using a traditional WIMP interface (*Windows, Icons, Menus, Pointer*), diagrams are typically created using a designated application: the various diagram components are selected from some graphical widget like a list, and placed on the canvas with one or more mouse clicks. Using a stylus we can now change this way and let the user simply draw the diagram components as with pen and paper. This is by far more natural, and in some cases more preferable. For example, in early stages of software design (where many diagrams are typically used) designers often dislike traditional software but tend to use pen and paper instead [19]. One may argue that a mouse could also be used to draw the components; this is true, but requires much training. Even drawing a straight line is far from easy, let alone complete components (e.g., an actor symbol from UML diagrams). In the following, we assume that a stylus is used as input device, although a mouse would also be conceivable.

Imitating the pen and paper approach with a computer gives the user the best from both worlds: a very natural input mechanism plus strong editing capabilities like copy and paste, and saving and loading diagrams to and from disk. The problem, however, is to have the computer understand the information contained in a diagram for further processing. A precise diagram created by selecting and placing ideal components on the canvas can be much more easily processed than an imprecise, sloppily drawn diagram. This is the very challenge of dealing with hand-drawn diagrams.

In this paper we present an approach to generating diagram editors capable of supporting hand drawing. The main characteristics of our approach are: (i) there are as little restrictions to drawing components as possible, (ii) syntactic and semantic information is used to resolve ambiguities which have occurred in the recognition process, and (iii) the approach is completely generic. A detailed discussion of these aspects, along with further requirements and design goals, is given in Sec. 2. Because of (ii) we have decided to base our approach on *DiaGen* [21]. It is a generic, powerful tool for generating diagram editors based on user-defined specifications, and fits our purpose very well. We have a fully working prototype implementation of our approach in

AVI '08, 28-30 May , 2008, Napoli, Italy.

Java.

Some basic terminology: each *diagram* has a certain *(diagram) type* and consists of several *(diagram) components*, i.e., its visual entities. The components available for a diagram obviously depend on its type. The process of drawing a diagram by hand is also referred to as *sketching*. An important aspect of sketching is to identify all components in a hand-drawn diagram, which is called *recognition*.

For a running example we use simple *Petri nets*. Diagrams of this type consist of only four different component types: places, transitions, arrows and tokens. An arrow connects one place to one transition, or the other way round, but never two places or two transitions. Additionally, places may contain one or more tokens. In order to ease drawing, places are depicted as circles, transitions are depicted as rectangles, and tokens are depicted as small circles.

The remainder of this paper is as follows. Sec. 2 describes and discusses goals and requirements that have guided the design of our approach. Sec. 3 describes the steps taking place to recognize components from a hand-drawn diagram. Sec. 4 explains the basic concepts of DiaGen and the modifications necessary to take the difficulties and peculiarities of sketched components into account. Sec. 5 describes two case studies we have used to assess the behavior and the performance of our approach. Sec. 6 briefly introduces related work. Finally, Sec. 7 summarizes this paper and gives an outlook on future work.

2. DESIGN GOALS

A stylus allows for a natural input of sketched data, like with pen and paper. Following [26, 10], we do not want to diminish this important advantage by enforcing artificial restrictions on the user, as many other related approaches do. Restrictions may come in very different ways. Some are related to the drawing of single components, e.g., drawing a circle always in one stroke, starting at 12 o'clock, going clockwise. Other restrictions require the user to indicate if he has finished drawing a single component, which also implies that any two components must be drawn one after another without any temporal overlapping. Completion of a component may be indicated by waiting some amount of time [2], for example.

Such restrictions are the consequence of insufficient processing capabilities, as they force the user (while drawing the diagram) to deliver some extra information which is required by the underlying system to understand the diagram components actually drawn.

In order to avoid restrictions we have decided to employ a model-based approach, i.e., all strokes drawn by the user are converted to an internal model first and then processing only takes place within this internal model (actually there is more than one model, see next section). This way, restrictions like the ones mentioned above are effectively excluded, since the model does not contain any information about *how* it was constructed. The two main alternatives to model-based approaches are image-based approaches like [18] or feature-based approaches like [12], often employed for processing hand-written text [25]. The characteristics of the model we employ are discussed in Sec. 3.

Due to the sloppy user input, ambiguities naturally arise during recognition. Fig. 1 gives an example of a problematic case of a Petri net: the component to the left, labeled with a question mark, could be either a circle or a (rounded) rectangle, i.e., a place or a transition. An easy way to resolve such ambiguities is to employ some *mediation* technique, e.g., providing the user with a list of possible alternatives and let him decide what it actually is (cf. [20]). Instead of eliciting such artificial user input, it would be better if we had the system automatically decide for one representation. Our approach can do so by the use of the syntactic structure and semantic information contained in diagrams. In the depicted case, the arrow can only be placed in the diagram if the component in question is found to be a circle, i.e., a place, as the arrow cannot connect two transitions in a Petri net.

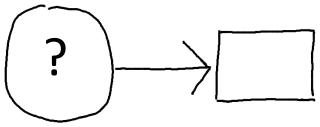


Figure 1: The component to the left is ambiguous.

As mentioned before, we rely on DiaGen, which comes with powerful support for syntax checking and semantics deduction. We have designed our approach to work with DiaGen to utilize exactly these features (cf. Sec. 4). Another important characteristic of DiaGen is its performance. Even large diagrams may be parsed and processed very fast. By adding support for hand drawing, obviously some extra time has to be spent. Nevertheless, it has shown that these extra costs do not have a critical performance impact, although for large diagrams they are noticeable (cf. Sec. 5).

Our final design goal is to have a generic approach which can be customized to arbitrary diagram languages by some language specification, something already achieved in DiaGen. For our approach it is (besides specifications required by DiaGen) necessary to specify the visuals of the components in a manner suitable for the underlying recognizer to identify components from the drawing. We use XML specifications which are conceptually similar to SkG [9] or Ladder [15], but somewhat simpler. Basically, components are defined by their visual primitives, i.e., straight lines and arcs. For instance, a rectangle (a transition) consists of two horizontal and two vertical lines, connected to each other at the corners.

3. THE RECOGNITION PROCESS

Obviously, a line drawing always may be seen as a set of strokes, where a stroke is a portion of ink digitized between putting the stylus on the canvas and the next lifting of the stylus. More formally, it is a list of tuples each containing a pair of coordinates and the time elapsed since the beginning of the stroke. This representation of a stroke is the natural consequence of the stream of events generated by the stylus, and is assumed by virtually any other approach as well. Since we currently do not use any time information, we regard a stroke as a list of (x,y) pairs.

Before diving into the details of the recognition process we start by explaining our overall system architecture, as shown in Fig. 2. Data structures are denoted as rounded rectangles, while processing units are denoted as regular rectangles. Using a drawing tool the user paints strokes on the canvas. The strokes are passed to a number of transformers, each processing the strokes according to their own criteria, and adding

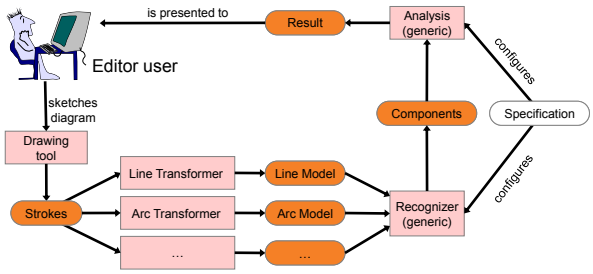


Figure 2: System architecture of a generated sketching editor.

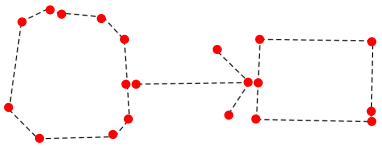


Figure 3: Line model of the drawing from Fig. 1.

the resulting information to its corresponding model. The recognizer then identifies all components (e.g., all places, transitions, arrows and tokens for Petri nets) in the drawing based on the information contained in the models. These components are then passed to DiaGen for further analysis (see next section). Then, the final result can be presented to the user. Recognizer and analysis are tailored to a diagram type by a specification.

The transformers work independently of each other, i.e., each stroke is processed without relying on each other. Consequently, the resulting models are independent, too. This makes it easy to add new models, and simplifies creation of models, as they are strictly self-contained. We currently have two models, one for straight lines and one for arcs (other models are conceivable, e.g., for filled or hatched areas). Each stroke is transformed into these models, one stroke at a time, immediately after it is completely drawn. Using models has two important consequences. First, when transforming the strokes into the models, the information contained in the strokes can be filtered (or *abstracted*), i.e., unnecessary information can be omitted and the remaining amount of data gets smaller, thus speeding up subsequent processing. Second, the actual search takes place in the models *only*, and does not consider the original strokes any more. Therefore, the transformer must not suppress any relevant information.

For our approach both models are undirected graphs with some additional attributes at the vertices and the edges. Each vertex has assigned a pair of coordinates reflecting its original position, so that spatial information is retained. In the *line model*, each edge represents an actual drawn line which has no bends. If a line has bends, it is split into several edges. The original location of the line is given by the coordinates of the two vertices connected by the edge. Each edge has an attribute which tells about its direction, and has one of four values (horizontal, vertical or one of the two diagonals). This attribute speeds up the recognition process, as it allows for easy filtering of lines. See Fig. 3 for the line model of the drawing given in Fig. 1. Vertices are depicted as dots, edges are depicted as dashed lines. Attributes are not shown. It can be clearly seen that the ambiguous component, while drawn in a single stroke, consists of nine edges in the model.

In the *arc model*, each edge represents an actual drawn arc with a smooth turning. Again, if an arc does not have a smooth turning, it is split into several edges. If an arc (or part of it) is a straight line, it is left out from the arc model. Each edge has an attribute which tells about its curvature (the spanned angle).

The algorithms for transforming the strokes into the models basically work as follows. From three consecutive tuples p_1, p_2, p_3 of a stroke, p_2 is discarded in the line model if the angle $\angle p_1 p_2 p_3$ is nearly equal to 180°, i.e., p_1, p_2, p_3 roughly lie on a straight line. We use a threshold of 20° here. From four consecutive tuples p_1, p_2, p_3, p_4 of a stroke, p_2 is discarded in the arc model if the angle $\angle p_1 p_2 p_3$ is nearly equal to the angle $\angle p_2 p_3 p_4$, and if it is not too small (according to a threshold value), i.e., p_1, p_2, p_3, p_4 roughly lie on a smooth arc. The threshold avoids approximation of very small angles (corners) by arcs; we use 50°. Finally, each tuple not discarded from a stroke is then used as a vertex in the model, and the vertices from each two consecutive tuples are connected by an edge. Practical results show that the number of vertices in both models usually is less than 5% of the number of pairs from all strokes, which means that the amount of data is clearly reduced. However, this figure depends strongly on the values of various thresholds we use. Statistical data is not available.

Obviously, the same stroke can be represented in several models. This is necessary because we do not know yet to which component the stroke will eventually belong, so we cannot decide on its correct representation as yet. For example, if the ambiguous component in Fig. 1 is a rectangle, then it is found in the line model; if it is a circle, then it is found in the arc model (of course, one component can consist of edges from different models, but this is not the case for Petri nets). The recognizer will find both representations.

Various further reduction mechanisms, applied to both of our models, ensure that the models contain as little clutter as possible. One example is vertex removal; if a new vertex has to be added to a model, there must not be any other vertex within some small distance. Otherwise both vertices are merged. Similarly, edges shorter than a certain threshold (measured as the Euclidean distance between the two vertices it connects) are discarded.

Before the actual recognition takes place, some extra information is generated into each model to simplify the recognition process. First, all edges are split at intersections with other edges (even if the point of intersection is not part of, but close to the edges, cf. the arrow head and the left line of the transition in Fig. 1; the same applies for the arrow tail). Second, each vertex is attached with a list of all other vertices close to but not connected to it. This is necessary because the drawings may contain gaps which must be jumped over, cf. the lower right hand corner of the transition in Fig. 1.

The actual recognition process then attempts to identify all components in the models. Training data is not required, the specification is sufficient as component description. For the recognition we have enhanced the procedure

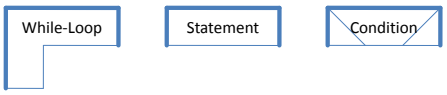


Figure 4: Components of Nassi-Shneiderman diagrams and their common visual primitives (thick lines).

given in [4][1]. In order to identify a component, one of its visual primitives (lines and arcs) is searched. For each occurrence, the next primitive is searched, which must be directly connected to the first one (we consider only connected components here). This process continues until either no further primitive can be found, or until all primitives are found, which means that a component is identified. In order to speed up this process, a *search plan* is computed which allows for searching for similar components concurrently. The search plan first looks for common primitives, appearing in different components, and then branches search for the individual rest of each component. An example is given by Nassi-Shneiderman diagrams (NSD), whose components are shown in Fig. 4. Common primitives are indicated by thicker lines. Note that the actual choice of similarities is not unique in general. We apply some heuristics to gain a result with a possibly high number of common visual primitives. For NSDs the recognizer first tries to find the common three lines, and then tries to identify the remainder of each component. For Petri Nets, the recognizer identifies places and tokens at the same time, as they have the same primitives. Only in the final analysis step, tokens and places are distinguished according to their position relative to other components, i.e., their context.

The result of the recognizer is a set of all diagram components that could be identified in the models. Usually there are more components identified than there are actually drawn components, due to tolerances in the recognition process. Each identified component is provided with a *rating*, i.e., a positive real number indicating the quality how well the type of the component is represented. The rating depends on the complexity of the component, i.e., the number of lines and arcs it consists of, and on how precisely it is drawn. The rating serves two purposes: it can be used to compare different components of the same type, since they have the same complexity (but may not be drawn with the same precision), and it can be used to compare components of different types, whose complexities are likely to differ. Comparing components with each other is necessary for the integration into DiaGen (see next section).

4. ANALYSIS WITH DIAGEN

DiaGen is a generic generator for diagram editors. Editor generation is based on language specifications. So far, the editors generated by DiaGen did not support sketching. By combining our approach with the DiaGen generator not only sketching-enabled editors can be generated. Also, recognition of sketched components greatly benefits from the facilities included in DiaGen. In the following, we describe DiaGen with its support for regular editors without sketchy input first, and then the modifications necessary for sketching.

[1]unlike the present paper, [4] treats only the recognition process.

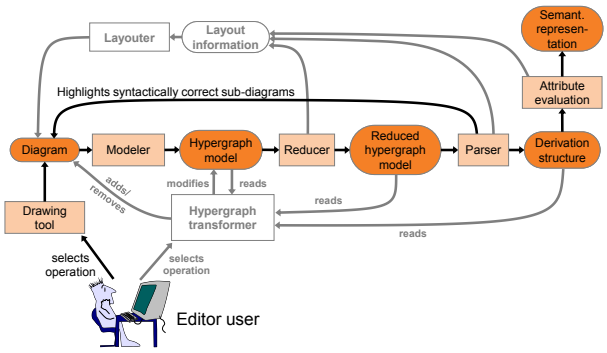


Figure 5: Architecture of a generated DiaGen editor. Aspects not relevant to sketching are grayed out.

The architecture of a generated editor is shown in Fig. 5. Again, data structures are depicted as rounded rectangles, while processing units are depicted as regular rectangles. The grayed out parts relate to the layout of diagrams and to operations (predefined modifications to an existing diagram, also known as *structured editing*), that are not relevant in the context of sketching.

The user creates diagrams with the *Drawing tool*, which provides the GUI for the actual editor. The result is a diagram, which is just a set of unrelated diagram components. Analysis then proceeds in the following steps:

- the *modeler* searches the set of all components for relationships. Relationships describe how two components are related to each other. In the case of Petri nets, relationships are required to attach the head and tail of an arrow to a transition or place, and to relate tokens and places. The modeler produces a *hypergraph model* (HM).
- then, the *reducer* applies the reduction rules. This roughly corresponds to a lexer for textual parsers; from the components and its relationships, terminals are generated. The reducer yields a *reduced hypergraph model* (RHM).
- finally, the *parser* tries to deduce the start symbol of the grammar in a bottom-up fashion. If this is successful, semantics can be computed based on attributes in the derivation structure.

The language specification for DiaGen consists of descriptions of the diagram components, their relations, the reduction rules, the terminals and nonterminals (with attributes), the grammar productions, and attribute evaluation rules. Internally, DiaGen employs hypergraphs [3] to represent its model and its reduced model. Components and relationships are represented as hyperedges. Therefore, the grammar used for parsing is a hypergraph grammar [11]. However, such technical aspects are not considered here.

For sketching support, we have replaced the original DiaGen drawing tool with the editor shown in Fig. 2. The set of all components is generated by the recognizer. The original modeler, reducer and parser have been modified accordingly. This way, ambiguities as the one shown in Fig. 1, where the component with the question mark may be a place, but not

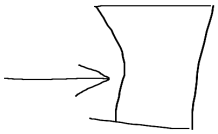


Figure 6: A very deformed transition and an arrow which is related to the transition despite its deformation.

a transition, can be dealt with. With regular editors without sketching support, such kind of ambiguity cannot occur. Of course, a component may be syntactically misplaced, for instance, and DiaGen is equipped with some error handling capabilities, but they cannot help for resolving the described ambiguity.

Modifications to the Modeler. To detect relationships between components, the attachment areas of all components are checked pairwise to see whether they overlap. We add some threshold which allows for the detection of relationships even if the related components miss each other by some distance (which is *very* common for hand drawing, cf. the arrow head in Fig. 1, which does not perfectly touch the transition). Furthermore, relationships are also detected if the related components are heavily deformed (cf. Fig. 6), something which cannot happen in DiaGen, since all components are drawn perfectly. The modeler also handles overlapping components. For example, if the shaft of an arrow is too long and overlaps with a transition, the shaft is cut at the appropriate position to perfectly relate the two components to each other.

Modifications to the Reducer and the Parser. The modifications of the reducer and the parser depend on each other. What the reducer basically does is to transform the HM by repeatedly applying the reduction rules (which are, in fact, graph transformation rules). It is very common that there is a 1:1 mapping between terminals (abstract syntax) and components (concrete syntax), like with Petri nets, but our system also supports diagram languages where this is not the case. Rule application can be restricted by (among other things) *negative application conditions (NAC)*. A NAC is a (connected) set of components which must not be present in order for the reduction rule to be applicable. In Petri nets, for example, a transition from the HM is reduced to a transition in the RHM. This may take place only if the transition does not overlap with other transitions or places. This restriction can be expressed as an NAC. Using NACs, however, introduces another big issue. Again consider Fig. 1, where the recognizer identifies a place and a transition for the ambiguous component, and they surely overlap. With the reduction rule mentioned above (and a second corresponding rule for places), neither the place nor the transition would be reduced. Hence, no terminals would be generated, and the parser would not be able to take these components into account. The solution is to temporarily ignore NACs when applying reduction rules [5]. Because we do not want to change the semantics of the reducer and the parser, we do some bookkeeping: for each terminal edge in the RHM we store the components conflicting with it due to NACs (if they had been used). In the example, the reduced place will thus store the transition as a conflict, and the reduced transition will store the place as a conflict. The parser then uses the terminals to apply the productions from the grammar. It must now make sure that it does not use two terminals which are conflicting with each other on the right hand side of the same production rule. In this case, the rule must not be applied. Accordingly, information about conflicts is not only stored in terminals, but also in nonterminals. In the end, when the start symbol is reached, we have established that no two terminals are used in its derivation structure which do not fit together. This means that, while the semantics of the parser has not changed, diagrams can now be successfully parsed which otherwise could not.

The general idea behind the modified reducer and parser is to postpone the final decision for some representation (place or transition) as long as possible, until it *must* be made in order to proceed with processing. This approach ensures that as much contextual information is present as possible. The drawback is that in general the bookkeeping mentioned above is complicated and memory consuming. Furthermore, the reducer is required not to miss any component (i.e., a false negative), as it cannot be identified otherwise and would be lost forever. For this reason, the reducer is very tolerant, thus generating a lot of false positives (which is no problem, since the parser reliably discards them).

In general, the parser will find more than one start symbol, i.e., more than one syntactically correct (sub)diagram. Among the corresponding derivation structures we discard those whose semantics cannot be computed for some reason. The remaining derivation structures are rated. Various ratings are conceivable, but a very simple approach proved to work very well: we simply add up all the individual ratings for all components used to reduce the terminals in a derivation structure. Finally the structure with the highest rating is chosen to represent the drawn diagram.

5. CASE STUDIES

Regarding syntax, Costagliola et al. distinguish two different categories of visual languages: *connection-based* and *geometric-based* [8]. The former covers all kinds of diagrams with a graph-like structure, like Petri nets, while the latter covers diagrams where components are related by their spatial arrangement, e.g., Nassi-Shneiderman-Diagrams.

Following this distinction, in two case studies we have examined diagrams of both types in order to show the applicability of our approach and to identify characteristic behavior. Their underlying graph grammars are very different and together cover the full range of possible production rules, which makes them excellent test cases. All tests were performed on a PC with an Intel dual core CPU with 2.4 GHz, 2 GB RAM. As input device an LCD tablet (a Wacom DTU-710) was used. Note that the correct diagram interpretation was found in every case.

We wanted to find out how long it takes to fully analyze Petri nets of various sizes. We have drawn the simple Petri net shown in Fig. 7 (top) consisting of three places, three transitions and seven arrows, making a total of 13 components. We have taken 20 measurements of the time necessary to recognize the components in this diagram and parse them. We have repeated this procedure ten times, each time copying the original 13 components next to the right-most component on the canvas, thus linearly increasing the input size. The result is depicted in Fig. 7 (bottom). The x-axis depicts the number of components (a multiple of 13); the y-axis depicts the time in milliseconds. The lines represent

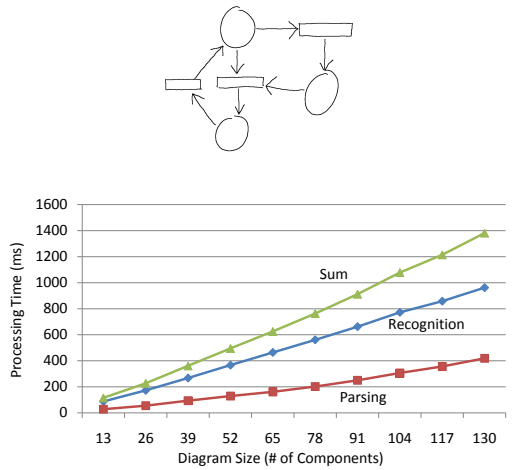


Figure 7: The Petri net used for the case study (top), and the results from the performance test (bottom).

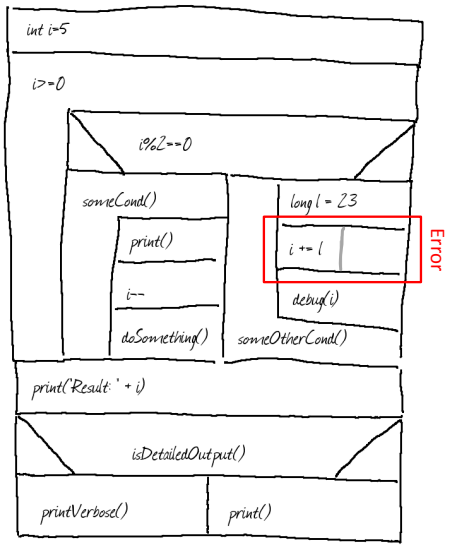


Figure 8: A larger example of a sketched Nassi-Shneiderman-Diagram which is processed correctly. The gray line in the marked statement is erroneous.

the minima for the 20 test runs. The lower line in the figure shows the time required for parsing the diagram, the middle line shows the time for recognizing the components, and the upper line shows the sum of recognition and parsing, i.e., the full time required for diagram analysis.

Note that the average time to analyze a Petri net with, for example, 39 components is about 360ms, which is still suitable for a real-time application. Larger diagrams require more time, which may become cumbersome in a real-time environment. This is why we have decided to require the user to explicitly invoke analysis by pressing a button (this behavior is referred to as *recognize on demand*; it has been shown to have some advantages over immediate interpretation [17]). Actual time consumption grows faster than linear. The parser and some methods from the recognizer have a higher complexity. Parsing complexity, e.g., depends very much on the specified grammar.

Unlike Petri nets, Nassi-Shneiderman-Diagrams have two characteristics which severely degrade performance of diagram analysis. First, the recognizer identifies every two consecutive statements as a third statement where the horizontal line separating the two statements is just ignored. For a very simple NSD consisting of four consecutive statements drawn by the user, the recognizer will thus identify ten statements. In general, for n consecutive statements, the recognizer identifies $O(n^2)$ statements which are passed to the parser. Additionally, the grammar accepts each single statement as a correct NSD, or the composition of two consecutive NSDs. In the example with four consecutive statements, this means ten correct NSDs for each single statement identified by the recognizer, and further 16 for combinations of these, resulting in a total of 26 NSDs (the exact number is $2^{n+1}-n-2$ for n statements, which means an exponential complexity). Loops and conditions have a similar behavior.

A larger example of a NSD is depicted in Fig. 8. The recognizer identifies 56 potential components (compared to the 15 that have actually been drawn), resulting in 4649 potential diagrams returned by the parser. From these, the correct representation was selected correctly due to the rating of the derivation structures. However, the negative effect on time and memory consumption is severe and must be discussed. While the recognizer took around 80ms, the parser took around 700ms to compute its result. This is a big contrast to the Petri nets, where most of the time is consumed by the recognizer. Moreover, with so many derivation structures, the system quickly runs out of memory.

As a solution we added an option to the recognizer which allows for suppressing components if they completely cover other components. The drawback is the reduced tolerance of the system, e.g., consider the marked statement in Fig. 8. The gray line is erroneous. It breaks the statement into two next to each other. Using the mentioned option, the system no longer detects this error, because the original statement (ignoring the vertical line) is suppressed. Without the suppression, the correct diagram is recognized at the cost of heavy time and memory analysis.

Using the recognizer option mentioned above, for a series of practical examples we have obtained acceptable performance results. In a real-time setting where a sketch is analyzed immediately after each change, our current implementation does not suffice any more, because sketches are analyzed from scratch each time and only the models are built incrementally. We, therefore, expect substantial performance gains from incremental recognition and parsing which we are going to implement as future work.

6. RELATED WORK

Starting with the famous *Sketchpad* [27] in 1963, a large body of work has been published on gesturing, sketching and related issues over the last two decades. In this section we briefly describe some of the more recent, related publications.

Alvarado et al. present an approach which is not generic, but limited to mechanical devices. Special emphasis is put on ambiguity resolution [1]. The architecture of their system is similar to ours: a recognizer detects primitive shapes like bodies, springs or pin joints, by assigning the respective

interpretation to each stroke. Then, reasoning about these shapes takes place, i.e., they are scored by some basic rules (this is, in fact, the ambiguity resolution; in case of ambiguity, hard-coded rules decide for the interpretation). The final stage, resolution, selects the final interpretation for the sketch.

Hammond et al. describe an approach limited to the recognition of UML class diagrams [14]. Strokes can be drawn in any order. They are not fed into a data model, but kept separately from each other. Recognition takes various combinations of strokes and tries to interpret them. If the result is ambiguous, decision is deferred to an *identification* stage, where several heuristics are applied to select the final interpretation. Recognizers for the components available in class diagrams are hard-coded. Unlike our approach, editing gestures (single strokes which are immediately assigned a meaning) can also be drawn, thus seamlessly integrating into the drawing process. We require the user to click a button on the stylus to explicitly indicate the editing mode, because we do not want to distinguish editing gestures and drawing strokes based on a heuristics.

A generic framework, *InkKit*, is contributed by Plimmer et al. [7, 24]. Based on the Microsoft Tablet SDK, editors can be built with comparatively little effort. Recognition is done by a modified gesture recognizer based on Rubine [25], which also supports multi-stroke symbols. The framework is also capable of automatic separation of drawing and text. The authors report very good results for various diagram types. Resolution from ambiguity is quite different to our approach: among all potential interpretations, the most probable ones are taken that cover the whole drawing.

An approach by Costagliola et al. [10] addresses ambiguities as the central problem. Like our approach, the context of a component is used for resolution. However, the concepts are different. Three levels of recognition are employed: at the lowest level, primitives like lines and arcs are constructed from the input strokes by the *SATIN* toolkit [16], a predecessor of InkKit. At the next level, the primitives are grouped into (partial) symbols, each having an *importance rate*. Based on this rate and the context, at the highest level it is finally decided for interpretations, and conflicting partial symbols may be pruned. The full system works incrementally. The reported recognition rates typically exceed 90%, with some exceptions. As a comparison, recognition rates are also taken without disambiguitation, which clearly reduces recognition rates.

Casella et al. [6] propose a conceptually interesting framework for sketch understanding based on agents. Arbitrary symbol recognizers can be added to SRAs (*symbol recognition agents*), which manage the recognizers and mediate between them. The SRAs may exchange context information. Conflicts are solved by a central agent which has the full overview over all SRAs. There are no experimental results provided, since there is currently no working implementation available.

The approach of Grundy and Hosking, *MaramaSketch* [13], builds upon existing frameworks and libraries, like Eclipse, and the HHReco toolkit for recognition, thus focusing on more high-level aspects of sketching. Their tool supports (among other things) lazy and eager recognition, informal and formal representation, and explicit user interaction for ambiguity resolution. On the contrary, we decided to develop every component of our system by ourselves to have full control over every single aspect, and we explicitly want to avoid user interaction for ambiguity resolution.

7. CONCLUSION AND FUTURE WORK

In this paper we have presented a generic approach to include hand drawing in diagram editors. Recognition of diagram components uses internal models, one for straight lines and one for arcs. Ambiguity resolution is done by syntax and semantic analysis. Two case studies have shown the practicability of our approach. Processing times for typical diagrams of a reasonable size usually are less than a second, which should be acceptable. Ignoring user input for ambiguity resolution works well in the shown case studies, although we cannot guarantee that the system always makes the right decisions, i.e. infers the meaning intented by the user.

Although all examples discussed in this paper are recognized and processed correctly, our approach does not show a recognition rate of 100%. By testing we have found out that the recognizer is the critical part, as it occasionally fails to identify a correctly drawn component. Especially arcs and circles are problematic. As future work, we will improve our algorithms and perform extensive field studies. Furthermore, our recognizer is still limited in terms of visual primitives: support for hatched or filled areas is missing, and components cannot consist of several unconnected pieces. It is important to assess the impact of these limitations and find ways to overcome them.

Recognition of text has not been discussed yet. Intuitively one expects to put text anywhere on the canvas just like diagram components and let the computer decide what is what. The challenge is then to separate text from drawing; [13] defines *text areas* where all entered strokes are automatically treated as text, for example. In our current implementation we require the user to make a double tap on the canvas to explicitly indicate that text is to be entered. Then a window pops up and allows the text input. Although this procedure is somewhat artificial, it brings two benefits: (i) a broader range of text input approaches can be used and (ii) it is possible to edit previously entered text when tapping on it. We have included six different text input approaches: plain keyboard input, a virtual keyboard displayed on the screen, the text recognition engine of Windows XP Tablet Edition, and reimplementations of Palm's Graffiti, Quikwriting [23] and a single stroke recognizer loosely based on Rubine's work [25]. So far, we have not performed any user tests on the practicability of these text input approaches.

Furthermore, we plan to investigate how *DiaMeta* [22], a similar approach to DiaGen, which uses metamodelling for specifying a diagram language instead of graph grammars, can be used to support hand drawing as well. Experience tells us that most people find it more convenient to specify a diagram language by a metamodel, so DiaMeta can be a valuable option. Further benefits and drawbacks will be evaluated in comparison with the present approach.

8. REFERENCES

[1] C. Alvarado and R. Davis. Resolving ambiguities to create a natural computer-based sketching environment. In *Proc. IJCAI '01*, pp. 1365–1374, 2001.

[2] A. Apte, V. Vo, and T. D. Kimura. Recognizing multistroke geometric shapes: an experimental evaluation. In *Proc. UIST '93*, pp. 121–128, 1993. ACM.

[3] C. Berge. *Graphs and Hypergraphs*. North Holland, Amsterdam, 1973.

[4] F. Brieler and M. Minas. A new approach to flexible, trainingless sketching. In *Proc. VMSIS '05*, pp. 43–50, 2005.

[5] F. Brieler and M. Minas. Ambiguity resolution for sketched diagrams by syntax analysis based on graph grammars. In *Proc. GT-VMT '08*, 2008.

[6] G. Casella, G. Costagliola, V. Deufemia, M. Martelli, and V. Mascardi. An agent-based framework for context-driven interpretation of symbols in diagrammatic sketches. In *Proc. VL/HCC '06*, pp. 73–80, 2006. IEEE.

[7] R. Chung, P. Mirica, and B. Plimmer. Inkkit: a generic design tool for the tablet pc. In *Proc. CHINZ '05*, pp. 29–30, 2005. ACM.

[8] G. Costagliola, A. Delucia, S. Orefice, and G. Polese. A classification framework to support the design of visual languages. *Journal of Vis. Lang. Comput.*, 13(6):573–600, 2002.

[9] G. Costagliola, V. Deufemia, and M. Risi. A trainable system for recognizing diagrammatic sketch languages. In *Proc. VL/HCC '05*, pp. 281–283, 2005. IEEE.

[10] G. Costagliola, V. Deufemia, and M. Risi. A multi-layer parsing strategy for on-line recognition of hand-drawn diagrams. In *Proc. VL/HCC '06*, pp. 103–110, 2006. IEEE.

[11] F. Drewes, A. Habel, and H.-J. Kreowski. Hyperedge replacement graph grammars. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation. Vol. I: Foundations*, chapter 2, pp. 95–162. World Scientific, 1997.

[12] M. D. Gross and E. Y.-L. Do. Drawing on the back of an envelope: a framework for interacting with application programs by freehand drawing. *Computers & Graphics*, 24(6):835–849. Elsevier, 2000.

[13] J. Grundy and J. Hosking. Supporting generic sketching-based input of diagrams in a domain-specific visual language meta-tool. In *Proc. ICSE '07*, pp. 282–291, 2007. IEEE.

[14] T. Hammond and R. Davis. Tahuti: A geometrical sketch recognition system for uml class diagrams. *Papers from the 2002 AAAI Spring Symposium on Sketch Understanding*, pp. 59–68, 2002.

[15] T. Hammond and R. Davis. Ladder, a sketching language for user interface developers. *Computers & Graphics*, 29(4):518–532. Elsevier, 2005.

[16] J. I. Hong and J. A. Landay. Satin: a toolkit for informal ink-based applications. In *Proc. UIST '00*, pp. 63–72, 2000. ACM.

[17] L. B. Kara and T. F. Stahovich. Hierarchical parsing and recognition of hand-sketched diagrams. In *Proc. UIST '04*, pp. 13–22, 2004. ACM.

[18] L. B. Kara and T. F. Stahovich. An image-based, trainable symbol recognizer for hand-drawn sketches. *Computers & Graphics*, 29(4):501–517. Elsevier, 2005.

[19] J. Lin, M. W. Newman, J. I. Hong, and J. A. Landay. Denim: finding a tighter fit between tools and practice for web site design. In *Proc. CHI '00*, pp. 510–517, 2000. ACM.

[20] J. Mankoff, S. E. Hudson, and G. D. Abowd. Interaction techniques for ambiguity resolution in recognition-based interfaces. In *Proc. UIST '00*, pp. 11–20, 2000. ACM.

[21] M. Minas. Concepts and realization of a diagram editor generator based on hypergraph transformation. *Sci. Comput. Program.*, 44(2):157–180, 2002.

[22] M. Minas. Generating meta-model-based freehand editors. In *Proc. GraBaTs '06*, 2006.

[23] K. Perlin. Quikwriting: continuous stylus-based text entry. In *Proc. UIST '98*, pp. 215–216, 1998. ACM.

[24] B. Plimmer and I. Freeman. A toolkit approach to sketched diagram recognition. In *Proc. HCI '07*. Springer, 2007.

[25] D. Rubine. Specifying gestures by example. *SIGGRAPH Comput. Graph.*, 25(4):329–337, 1991.

[26] T. M. Sezgin, T. Stahovich, and R. Davis. Sketch based interfaces: early processing for sketch understanding. In *Proc. PUI '01*, pp. 1–8, 2001. ACM.

[27] I. E. Sutherland. *Sketchpad: A man-Machine graphical Communication System*. PhD thesis, MIT, 1963.

User Studies on Visualization

Exploring Video Streams using Slit-Tear Visualizations

Anthony Tang[1], Saul Greenberg[2], Sidney Fels[1]

[1]Human Communication Technologies Laboratory
University of British Columbia
Vancouver, BC, Canada V6T 1ZN
+1 604 822 4583
{tonyt, ssfels}@ece.ubc.ca

[2]Department of Computer Science
University of Calgary
Calgary, Alberta, Canada T2N 1N4
+1 403 220 6087
saul.greenberg@ucalgary.ca

ABSTRACT
Video slicing—a variant of slit scanning in photography—extracts a scan line from a video frame and successively adds that line to a composite image over time. The composite image becomes a time line, where its visual patterns reflect changes in a particular area of the video stream. We extend this idea of video slicing by allowing users to draw marks anywhere on the source video to capture areas of interest. These marks, which we call *slit-tears*, are used in place of a scan line, and the resulting composite timeline image provides a much richer visualization of the video data. Depending on how tears are placed, they can accentuate motion, small changes, directional movement, and relational patterns.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces.

General Terms
. Algorithms, Human Factors.

Keywords
Information visualization, video analysis, video history, timelines.

1. INTRODUCTION
Digital video cameras are now commonplace and used in numerous settings. We use them to film events, as in a home or commercial video. We use them to transmit real time video for others to look at, as in teleconferencing. We also use cameras to record a fixed scene for later review, as in security or data capture situations. We define this last type of video as *stationary video scenes,* where the footage is typically captured by a strategically positioned fixed-mount camera. Such scenes typically comprise a fixed background and one or more objects that change or move within that scene, e.g., people moving in and out of view, the operations of machines, and so on. In this paper, we introduce *slit-tears,* which support the rapid exploration of stationary video scenes for events and patterns of interest.

A naive and time-consuming way to review video for these kinds of events and patterns is to simply replay it. A somewhat better way is to use *video scrubbing* (found in non-linear video editors), where moving the cursor over the video timeline plays the underlying footage at a speed corresponding to the motion of the mouse. This allows one to rapidly review or replay video footage at slower speeds. While better than replay, scrubbing can still be tedious, and error-prone: events are easily missed if they are brief or of relatively small in size in the video scene.

Automated techniques can help accentuate footage of interest. With *keyframe previews*, a single frame is selected from video scenes in a way that typifies that sequence, e.g., one that is the most visually similar to all others [14]. Multiple keyframes representing a longer sequence can be presented as a storyboard or slide show [8]. With *change detection*, video footage is analyzed and marked if it deviates from the background scene or from surrounding frames (e.g. D-Link's securicam's surveillance software www.dlink.com). Such automated methods are at their best for drawing a person's attention to single points in time, or for marking events of potential interest. Yet they do not reveal patterns as they appear in the video through spans of time and space. For example, consider a camera capturing traffic through an intersection. The above methods may show when a car passes through that intersection, but they will not easily reveal how many and at what rate they pass through that intersection during rush hour compared to a non-peak hour.

Consequently, we designed a real-time interactive application that supports the rapid exploration of a video record—not only to find events of interest, but also to see patterns between those events. To foreshadow what is to come, our method allows a person to draw marks, which we call *slit-tears*, anywhere on the source video to represent areas of interest. For every frame, the pixels under these marks are concatenated to a composite image, thereby creating a rich visual timeline of the selected video data. Depending on how tears are placed, the visualization can reveal things like object motion, small changes, directional movement, and relational patterns in the video scene.

We first detail how slit-tears builds upon previous methods of slit scanning and video slicing, and then explain our algorithm and interface. We then show by example how this new visualization allows us to create and inspect scene visualizations at two analytic levels:

- *event level,* where change events—even if temporally brief or spatially "small" —can be made more salient;
- *pattern analysis level,* where periodicity is revealed, patterns can be compared and correlated over time, and directionality and velocity of movement can be gleaned.

AVI'08, 28-30 May , 2008, Napoli, Italy

2. RELATED WORK

After briefly showing how videos are displayed in traditional timelines, we summarize *slit scanning* as done in photography. We then show several different approaches for visualizing video data as a timeline that have evolved from slit scanning.

Traditional timelines. Most non-linear editors offer a visual timeline as the primary way for editors to view and compose video sequences; Figure 1 gives an example. Clips in the timeline are identified by a thumbnail of the first frame in the clip. To see clip details, a person scrubs over the video by moving the timeline control. Such timelines are largely unsuitable for analyzing video data for small events (in time or space) or for identifying patterns.

Slit scanning in photography passes a panoramic film strip rapidly across a single vertical slit, which exposes film to only a narrow slit from a scene (indeed this is how most digital scanners work). Objects moving in a stationary scene *over time* are seen as motion captured *over space* [3]. Consider Figure 2a, where the slit was positioned over the finish line. The horses are captured over time as they move across the slit / the finish line, allowing for easy and accurate judging. Similarly, Figure 3b illustrates the use of slit scanning to capture the motion of a hammer thrower [13].

Slit scanning in video, sometimes known as video slicing, achieves a similar effect. Here, a slit is placed over a video frame, and the pixels of successive video frames under the slit are captured and added to a composite image. Levin [9] catalogues many examples of slit scanning in video. Figure 3 illustrates our own TimeLine [11] system, designed to facilitate temporal awareness of a remote collaborator. The viewer positions the slit atop the live video frame by moving the red vertical line, and the views in the four rows are updated dynamically to reveal what has happened over different time [1]. The top row is the last minute (we see he has not moved much). The 2nd row is the last hour (he arrived about 20 minutes ago, although people have passed in front of the slit briefly before that). The 3rd and 4th row show the last day and the last week (we now see rhythmic pagers over daytime and nighttime). What sets TimeLine apart from other slit scanning video systems is that it provides users with a dynamic interface to the video slicing mechanism. People can interactively change the location of the slit, immediately updating the entire visualization. People can also scrub the underlying video if they see a pattern of interest. Finally, they can retrieve detailed video of events in the distant past by selecting a past point in time. For example, if a frame is selected from several days back, the day, hour, and minute view are all updated to show the surrounding frames at fine-grained resolution. Taken together, these features make TimeLine effective in allowing users to actively explore temporal patterns of behavior visualized in a 2D timeline.

Other authors have explored the use of these slit-scans for automated scene change detection ([10][2][7]). These authors are primarily interested in automated segmentation of videos based on scene changes. Various methods, involving Markov models [10], statistical methods [2], and discrete geometry tools [7] have all been applied to this problem. While these works are similar by way of the visualization they produce, their spirit and intent is different: the present work applies this visualization for *human* consumption rather than for automated analysis.

Video volumes. We can also consider a video clip as a volume, where successive video frames are stacked atop one another, i.e., as a video cube [5]. Chung et al. illustrate several methods of rendering this cube, and show how rendering successive frame differences can be a compelling summarization of the video data [2]. Fels et al. provide another visualization by slicing through this cube using a cut plane thereby crossing both time and space (Figure 4). Slit-scanning can be viewed a subset of this method, as it realizes the specific case of an intersecting plane being placed perpendicular to the face of the cube. Video volumes are more general, as different effects can be achieved by using other geometric slicing shapes and other slicing positions.

Figure 1. Adobe Premiere Pro CS3 Timeline

Figure 2. Slit-scanning. (a) judging photo finishes, from [4] (b) George Silk Hammer Thrower, from [13].

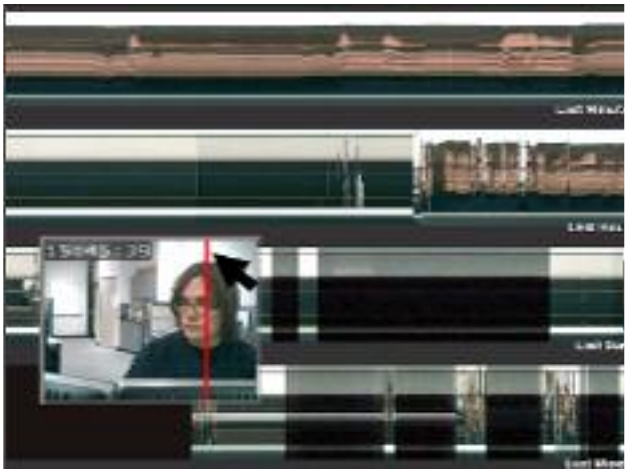


Figure 3. TimeLine, showing several days of webcam capture.

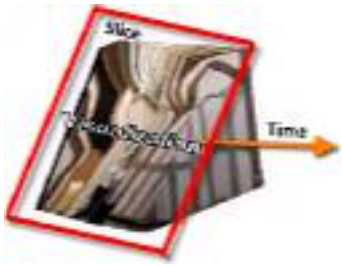


Figure 4. A sliced video cube, modified from [6]

While the non-traditional timelines above all produce somewhat abstract visualizations, in practice, people quickly learn how to read them. For example, Nunes et. al. report that people can quickly discover events of interest and patterns of activity in a telecommuter setting—to the point that significant privacy concerns are raised [11]. Yet slit scanned video is still used mostly for artistic purposes [9].

We believe that slit scan video techniques can be extended and practically applied to video data analysis. In particular, the following section introduces the idea of *slit-tears,* and later sections will show how it can:

- support post-hoc analytic exploration of video data,
- visualize spatial and temporal patterns,
- draw attention to brief or spatially small events,
- accentuate motion,
- indicate directional movement, and
- show relational patterns.

3. Slit-tears

The method. We have already seen how slit scan photography (and later slit scan video) captures a linear—usually vertical—slice of a frame's area, and portrays these as instances in time in a visualization. Yet, while photography is technically limited to a single linear slit, there is no need to impose this arbitrary limitation to digital video. Instead, we can capture a moment in time as multiple *slit-tears* in the video scene that are concatenated together to form a single column in the visualization.

With tear-based video slicing, we first allow the end user to draw multiple slits atop a frame's surface. Each slit is an arbitrary stroke—a straight line, curve, or scribble. For each frame, the system then captures the pixels under each slit in the order and direction that each stroke was drawn, and aligns these pixels into a single vertical column. It then appends this pixel column to the right side of the visualization.

Consider the sketch in Figure 5. The bottom represents several video frames that capture the top-down view of a room with three blue doorways. The top is the visualization created from a series of slit-tear columns, with each column capturing the next frame in the series. A person has drawn 4 strokes in the order numbered in the left-most frame. Let's now consider how the visualization captures particular events over time by comparing several frames with their corresponding slit-tear columns.

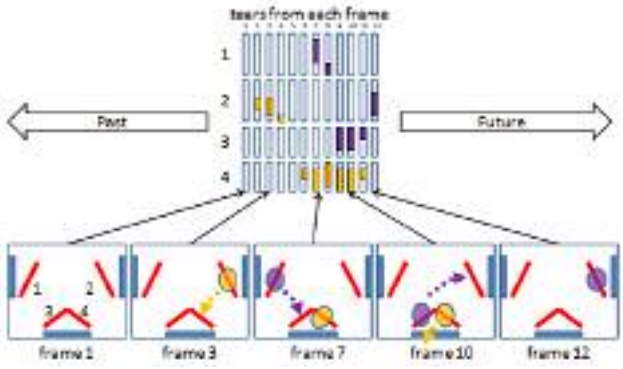


Figure 5. How slit-tears create a visualization

We see a person walking out of the right door across slit 2 (columns 2-4, frame 3), and then moving in front of the bottom door across slit 4 (columns 6-11, frame 7). While this is happening, a 2nd person walks out of the left door across slit 1 (columns 7-8, frame 7), and then also stands in front of the bottom door on slit 4 (columns 9-11, frame 10) until finally leaving out of the right door across slit 2 (column 12, frame 12). The two are also seen standing side by side (columns 9-11, frame 10). While abstract, a viewer can learn to read this visualization to interpret the flow of traffic in the room, infer casual encounters that take place as people move through that room, and optionally scrub over particular areas of interest to verify what actually happened.

The system. We have created two systems that realize the slit-tears method; their combined capabilities are described below. Figure 6 illustrates a situation identical to Figure 5. On the left is the video player, while the right shows the visualization. The video player normally shows the current frame from the running video. However, if a person scrubs over the visualization, a red scrub line appears (right side) and the video player immediately displays the frames from that scrubbed moment in time.

Users can create one or more slit-tears atop the live video frame, using the line or sketch tool (Figure 6, bottom left). In the figure, the analyzer began by sketching tear 1 atop the frame to capture traffic going through the left doorway. Similarly, tear 2 captures the right doorway, and tears 3 and 4 capture movement and activity around the opening to a seating area. All lines were drawn from the top down. This generates the visualization, where we see four horizontal regions representing the frame pixels underneath these lines stacked atop each other (in this image, each region is separated by black, and numbered to show the slit-tear line it represents). As in Figure 5, moving left to right in the timeline visualization, we see one person wearing a white shirt and blue pants has quickly walked through doorway 2 (2nd row),

Figure 6. Screen snapshot of our slit-tear video slicing system.

Figure 7. Monitoring pedestrians entering or passing by a store in an outdoor mall. Annotations show door and people events.

and then pauses by the opening (4th row). Shortly afterwards, the 2nd person wearing a blue shirt and black pants comes out of the left door (1st row), lingers at the opening along with the 1st person (3rd and 4th rows), and then leaves through the right door (2nd row). The first person disappears by continuing through the opening (the blur as he leaves suggests his direction of movement.

The system works with both live and previously-captured video (AVI files). For live video, the visualization updates itself with incoming video frames, generating new columns corresponding to the slit-tears. For previously-captured video, there are two options. One can treat it like live video, where tears and updates for subsequent parts of the visualization are done as the video plays. Alternately, one can update the entire visualization—past, present and future frames—to reflect the slit-tears. For previously captured video, the usercan also select different playback speeds (the speed slider in Figure 6) and the level of granularity of playback (the skip frames slider). When frames are skipped, details may be lost. However, the visualization then gives the viewer a broader picture of when and where events happened over longer periods of time.

We have now defined the slit-tear method and illustrated our system. Next, we will show how slit-tears can be used for user-driven analyses of video data, where strategic placement of slit-tears can create engaging and useful visualizations. We will show how an analyst can use tear-based video-slicing to explore video data to study incidents that occur at a point in time (events), and incidents that repeat over time or over space (patterns).

4. EVENT-LEVEL ANALYSIS

Slit-tears can visualize and emphasize events of interest. Consider Figure 7. The top left corner shows a cropped region from a low-quality 320 x 240 video. The scene is an outdoor mall, with a sliding door into a particular store at its center. The video analyzer is interested in the traffic patterns around the door in this small, blurry region within this video. She draws a single horizontal line across the typical path (labeled 1-path), and then scribbles a line over the doorway (2-door). The resulting visualization (top right) shows an interval in time, and we will use this visualization to answer questions about events in this segment.

How often do people walk by the entrance? The visualization contains solid uninterrupted horizontal lines when the scene is static (e.g., far left and far right regions). Perturbations occur when people walk by. To answer this question, she simply compares the ratio of static *vs*. perturbed regions in the video.

How many people walked by? Individual people are seen as streaks in the top of the visualization. In this case, we see that three streaks—three people—were captured in this video segment.

Which direction were people coming from? In Figure 7, the streaks all diagonal downwards over time. This means that all three people are moving from left to right. The reason is that these people have entered line 1 at its left side, which displays those pixels at the top of the slit column. As they move through the line, their captured pixels appear lower in the column. If a person was walking left to right, the streak would diagonal upwards.

How many people entered the store? The opening and closing of the sliding door is clearly seen as a blackish perturbation on the mid-left side of the visualization's lower half. We also see that this was caused by P1: his streak at the top slows down and then disappears abruptly as he enters the door, and his colors 'dissolve' into the open doorway at the bottom. We also see that P2 has not entered the door: his streak continues beyond the doorway at the top, and he has clearly not walked into the door at the bottom. P3 also walks past the door: his streak is continuous at the top, and the sliding door had not opened during this interval. Scrubbing over the scene verifies this: the bottom left frame shows P1 entering the door and P2 just behind. The middle and third frame shows P2 and P3 after they have just passed the doorway.

The video in Figure 7 is of poor quality, and the analyzer is interested in only a small somewhat blurry and poor contrast region of that video. Yet the resulting visualization reveals how slit-tear video-slicing can make even obscure events highly salient to an analyst, as described below.

More generally, events of interest can be problematic to see in conventional replay of video when the image quality may be poor, events may be very brief or spatially small, and patterns over time may be hard to detect. Slit-tears are a technique that helps to overcome these difficulties.

Events are readily seen even in low-fidelity video. Slit-tears reveal changes as they occur in a region on a static background. Change highlighting works even if images are blurry, pixelated and/or low-contrast. It will also work over noisy video, as motion tends to produce regular vs. random patterns.

Spatially "small" and/or poor fidelity events are exaggerated. Some events of interest may be spatially 'small', affecting only a modest number of pixels in the scene. These can be easy to miss or to decipher when replaying a long video. Even if we are expecting an event over just a few pixels, we can make that event highly salient simply by creating several slit-tears over that area; this enlarges the event's appearance in each column. To illustrate, Figure 7 (top left) is cropped (about 1/6 of the area) from a poor quality source video. The events of interest are even smaller—each person is only 20×8 pixels in size with indistinct edges, and the doorway is not much larger. Yet activity around the doorway is made salient by scribbling a slit-tear over it—the slit-tear 'expands' the doorway area to cover more than half of the visualization's height (along the bottom).

Figure 8. Several minutes of the same scene shown in Figure 8

Brief events are made extremely salient by virtue of how the timeline is constructed. In a long video scene, the timeline is a series of fairly unbroken horizontal lines; however, when objects *do* pass through the tears, they appear as an *intrusion* in the timeline. For example, a quick visual scan of Figures 6 and 7 allows us to rapidly spot these intrusions. As another example, Figure 8 shows the same the same timeline as Figure 7, except visually compressed to show over 7200 frames (several minutes of video). Even so, it clearly shows regions of interest as people move through the path and through the doorway.

5. PATTERN-LEVEL ANALYSIS

One of the strengths of slit-tear visualization is that it allows us to easily see not only individual events, but *patterns* in the scene. By patterns, we refer to how events relate to one another over time. Because the tears traverse space, and the timeline traverses time, temporal patterns of movement and behavior are visualized spatially. As well, events can be more easily correlated when the analyzer strategically places slit-tears to juxtapose them in the timeline. Examples are presented below.

What is the interplay of cars and pedestrians at an intersection? Figure 9 illustrates a scene captured from a traffic intersection. Frames t_1 to t_5 capture this scene at particular moments in time; these times are similarly marked in the visualization. The analyzer has drawn three tears. Tear 1 follows the path of cars traveling from the top to the bottom of the main road; the painted white lines marking the crosswalk and the center of the crossing road are also visible as faint horizontal white lines in the visualization. Tears 2 and 3 follow the path of the two pedestrian crosswalks. The visualization reveals interaction patterns between pedestrians and cars across this intersection. Moving from left to right in the visualization, we first see a red car moving into the scene (the angled red streak), then stopping for a while (the red horizontal streak around t_1 to t_2). We can understand this stop by other events occurring around that time. At time t_1 in tear 1, we see a bicyclist captured crossing the road in the center of the intersection. Looking at Tear 2 between t_1 and t_3, we see that the car is waiting for a pedestrian walking across the crosswalk (the pedestrian's path is the diagonal black streak). As the pedestrian approaches the other side of the crosswalk, the red car continues onward (we can see how far apart they are by the distance between the red car and the pedestrian in tear 2). After t_3 and until t_4, we see no cars, and several pedestrians are crossing the other crosswalk (tears 1 and 3). We can also tell their direction: those appearing as a downward diagonal streak are walking left to right, while those

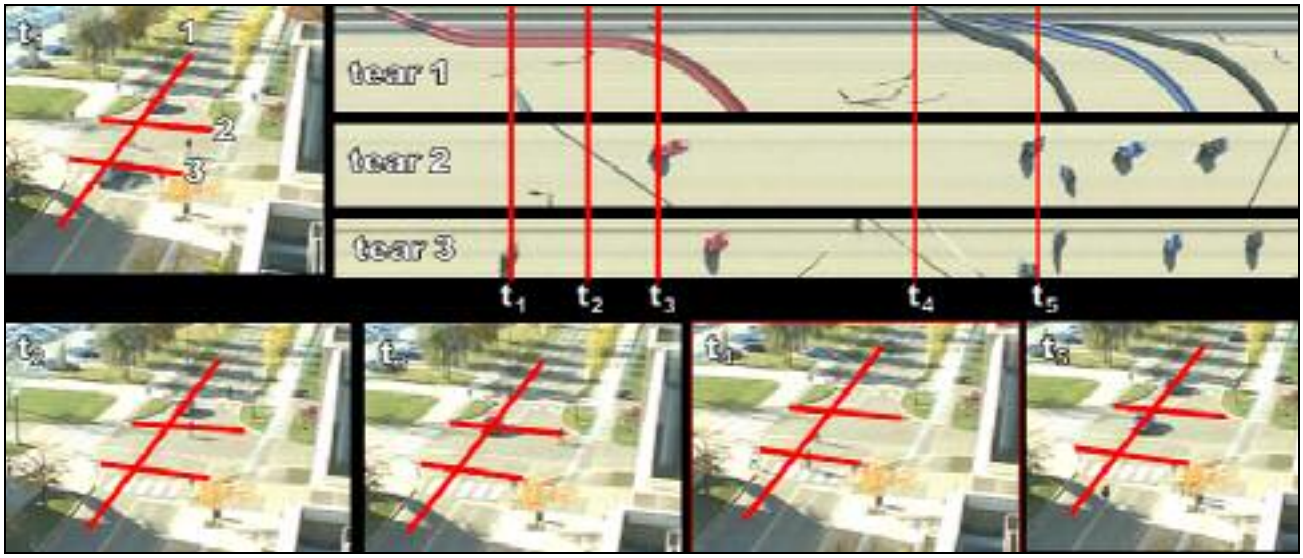


Figure 9. A traffic intersection showing the interaction patterns of cars and pedestrians

appearing as an upward streak are walking right to left. While this segment of the visualization does not show them, it could easily reveal other patterns, such as:

- cars that did not stop for pedestrians (e.g., a person enters the crosswalk but the car keeps on going),
- when pedestrians ran across the crosswalk in spite of approaching cars,
- near-misses.

When are people online over time and how does this lead to interaction? Figure 10 illustrates how an analyzer examines events and patterns over time in Google Talk, an Instant Messaging client. In this example, screen capture software was used to record Google Talk usage at 2 frames a second. Google Talk's interface is illustrated at a moment in time at the left. It alphabetically lists a person's contacts. Each name includes an 'availability status' icon on its left, and a personal image on its right. Conversations are shown as a white bubble, while green, orange and red icons indicate whether the contact is available, idle, or busy. The Google Talk user also gives its user the option to change a text message seen by others by typing over the field under their name (shown under the name at the top). Finally, if a person moves their cursor over a contact, the background will be shaded grey. The analyzer is currently interested in three things.

1. What is the availability status of contacts over time? She draws a slit-tear through each icon.
2. When do people change their personal image (a fairly rare event)? She draws a slit-tear through each image to capture this.
3. When does the person change their broadcast message? She draws a horizontal slit-tear through the current message so that message changes will be visible.

The timeline visualization shows several minutes of active GoogleTalk use. The grey strip preceding the conversation with the first person at t_1 suggests that the local user initiated the conversation (i.e., moving the mouse over the name, then double-clicking to initiate talk). The duration shows that it was a lengthy conversation. A parallel conversation happened at t_2 with person 2. Later conversations include person 1 at t_6, person 2 at t_3 and again at t_8, and person 3 at t_4 and t_5. A few very brief conversations (e.g., t_5) are suggestive of a quick message with the person not waiting for a response. We also see how availability status has changed: Person 2 has set their status from available to busy (the red strip), and person 4 has gone idle around t_6. Note that the local user contacted Person 2 as soon as their status changed from busy to available around t_8. We also see that one person has changed their personal icon at t_7. The constant image for the broadcast message indicates that it remained unchanged during this interval.

These examples show how tear-based video-slicing can reveal particular patterns to an analyst. These are generalized below.

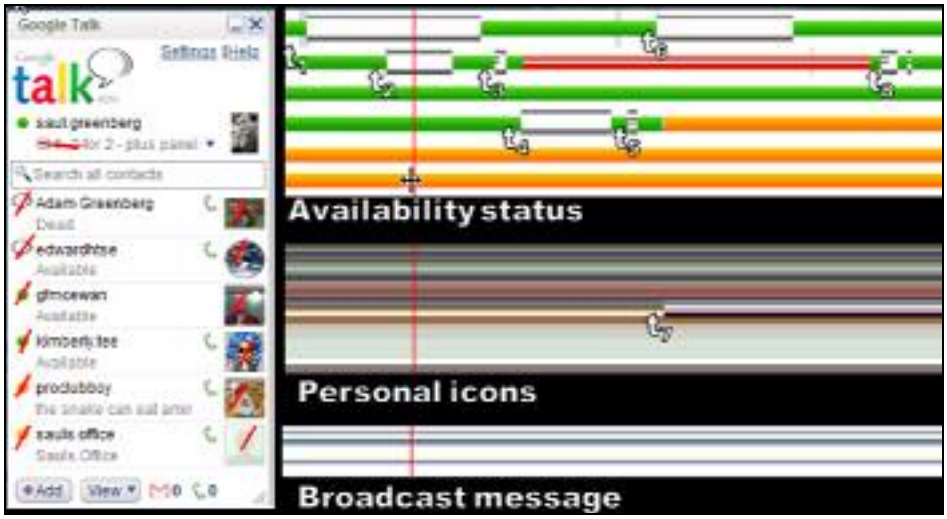


Figure 10. On-line status and talk in GoogleTalk

Rhythms and periodicity over a tear are easy to see. Movement or events that reoccur through a tear are strikingly easy to see since they appear in the timeline as a repeated intrusion on the scene. For example, if we ran the visualization in Figure 9 for a longer time and compressed the timeline, we would likely see traffic flow over the day, e.g., peaks in the morning, lunch, and work-end, quiet times at night. Similarly, we would easily see Google Talk user rhythms wax and wane during periods of the day and even across the weekend [1].

Similar individual events can be compared as a category. When multiple similar events are occurring, the analyzer can juxtapose them in the visualization by the ordering of the tears. For example, Figure 10 (top) juxtaposes the status of 6 people, leading to easy comparison of how conversations overlapped.

Different individual events can be correlated. Slit-tears of different events can be juxtaposed to see whether correlational relationships exist between objects, movement, or patterns in the video scene. Figure 9 correlates vehicle movement and speed through the intersection with pedestrian traffic on the crosswalk.

Directionality and velocity can be easily ascertained. With strategic placement of slit-tears, the directionality and comparative velocity of objects moving about the scene can be easily ascertained. Figures 7 and 9 show direction by the angle of the streaks, be they people or cars. Figure 9 also shows velocity: steep angles are high speed vehicles, shallow angles are low speed, and horizontal streaks means that the vehicle is stationary. Similarly, drawing a diagonal slit-tear through a side-view of an object, as done in Figure 11 (left), exaggerates direction and velocity. Cars passing through this tear are seen at different widths, and these widths suggest their comparative speed. "Longer" cars are moving more slowly than "smaller" cars; they are longer because they have stayed under the tear for more frames. For example, the extended black "limousine" in Figure 11 is actually a car waiting to make a left turn. Cars in Figure 11 are also slanted diagonally (somewhat cartoon-like), because different parts of cars going in different directions actually pass through different parts of the tear at different times. Thus, cars slanted forwards, are actually going right, while cars slanted backwards are actually going left.

6. DISCUSSION

We have introduced the concept of slit-tears and its interactive timeline visualization. We have also shown how it can be used across various video scenes in a way that reveals both events and patterns across time. We now briefly discuss some general advantages and limitations of this approach, as well as some extensions to our current system.

6.1 Advantages of Slit-tears

Slit-tears allows for exploratory video analysis. The strategic placement of slit-tears on the scene can visualize key events and patterns in the timeline. If one knows ahead of time what one is interested in, then one just places slit-tears over those areas of interest. Yet the power of the slit-tear technique is that it also allows for data exploration. The interactivity of slit-tear placement at any time (including clearing old slit-tears) means that people can use it as a tool for ongoing generation and provisional testing of hypotheses about the video data. For example, an analyzer of the traffic scene in Figure 9 could decide, after the fact, to look for jaywalking events (people on the road outside of the crosswalk), whether bicycles stop at the stop signs, the effect of traffic and pedestrians on right-hand vs. left hand turns, and so on. These and other explorations could be triggered by seeing unexpected events in the visualization.

Slit-tears is a generalizable video-analysis technique. For particular situations mentioned above, we could easily conceive of other analysis tools that might be better than slit-tears. For example, maybe other visualization methods could reveal patterns and events with greater clarity. Automated methods could analyze the scene (e.g. [2][7][10]), which could ease the analyst's burden. Alternately, we could deploy sensors at key locations to track specific types of events, thereby producing data more amenable for automated analysis, e.g., descriptive statistics.

The problem is that these alternate approaches are not generally applicable to the many every-day situations we may want to analyze. Automated analytic tools, for example, are typically only good for detecting particular kinds of events. They are also highly error-prone (e.g., as in the low-fidelity video in Figure 6 where the area of interest is only a few pixels in size). They easily miss events of interest: most operate through signal processing, which is divorced from the semantics of the objects and actors in the video itself. Sensors require heavy investment in time and materials in terms of their placement, to the point where their use is infeasible and/or illegal. Specialized visualizations need to be programmed to reveal particular kinds of information of interest. Aside from their inaccuracy and expense, these methods typically require that the analyzer knows *a priori* the events of interest, where they can then choose their tool accordingly.

In contrast, slit-tear visualization is a general approach for video collection and analysis that can augment other approaches. It requires no specialized equipment aside from an off-the shelf digital video camera, a tripod, and a computer for generating the visualization. It can be used in a broad variety of settings. This includes video captured for other purposes, e.g., surveillance and traffic cameras. It relies on no special analysis algorithms. It is an interactive visualization that lets the analyst decide upon what events should be captured by where slit-tears are positioned in the scene, and how they are interpreted.

Slit-tears is grounded in the actual data. Many statistical or analytic techniques abstract events and patterns in a scene, and present it in summary form: numbers, summary and correlation statistics, graphs, and so on. While useful, the analyzer may consequently overlook nuances of the captured data, simply because the raw data is stripped away or not linked to the summary view. In contrast, our slit-tear tool conceptualizes the analysis space in a readily understood "camera" and "pixel" space. The timeline simply takes selected areas of the video and translates the time dimension to a spatial dimension. The raw data is still visible in this representation. Further, rapidly scrubbing over areas of interest in the timeline reveals the source video, so that details that generated the visualization can be readily examined. This not only adds to a person's understanding of the pixel renderings in the timeline view, but shows other events in the scene that may reveal why events or patterns occurred.

6.2 Limitations

It only works with stationary video. Our examples are all based on stationary video. Tears are most easily interpreted as "portals" into a fixed location, where changes at those locations are easily spotted in the visualization. If the camera itself moves around the scene, the visualization is much harder to decipher. Slit-scanning has been used effectively with moving cameras before [9]; future research could explore when such situations are appropriate.

Appropriate camera placement is crucial. Aside from the normal concerns of video (lighting, field of view, etc.), the videographer has to ensure that events of potential interest are captured in the field of view, and that other events will not intrude to produce spurious patterns. For example, reconsider Figure 7, where the events of interest are the pedestrian traffic around several stores. If the camera was placed (say) across the street, then cars passing along the street in front of the camera would block events of interest and generate spurious patterns in the timeline. Birds-eye view and highly oblique views tend to produce the most useful video sources.

The analyzer must be immersed in the data. The analyzer needs to take an active role in placing slit-tears, and how to interpret the visualization. In our experience, this often requires a trial and error strategy: the analyzer places slit-tears, views the visualization and explores it by scrubbing to ensure that the correct data is not only captured, but presented in a way that leads to easy analysis.

6.3 System Extensions

The current system implements the core notion of slit-tears

Figure 11. Car speed through an intersection

(Figure 6), but omits several features of a proper video analysis tool. Several of these have already been implemented in our earlier TimeLine system [11], but we reiterate them here.

Overview to detailed views. For the system to be truly useful, the analyzer would need to examine the visualization across different time spans. The most detailed timeline uses a pixel column for every frame in the video (e.g., a single timeline row on a 1200 pixel-wide screen would display only 80 seconds of a 15 fps video). Yet, if a 24 hour video were to viewed in that same horizontal space, then each slice would have to represent 7.2 seconds. Clearly, events could be easily missed.

Nunes et. al. [11] addressed this problem with several strategies (Figure 4): first, the visualization shows the last minute, hour, day and week simultaneously; second, when a slice represents more than one actual frame, the frame that differs most from the previous slice is chosen (thereby ensuring each change is visible in the timeline), and finally, overviews are linked—clicking on a slice generates the appropriate views in the other rows. Alternately, a fisheye view strategy could be used to visualize details in an overview [12].

Resources. Our current system keeps all video frames in memory. This is clearly impractical for very large videos. We previously suggested several resource-reduction methods [11], all applicable to our extension into tear-based video slice generation.

Editing operations. Split-tears should be individually editable, where the visualization is dynamically updated as editing is occurring (our current system only allows tears to be drawn and removed). The system should include conventional line-editing methods as found simple drawing applications. As well, the system should let a person change a slit-tear's drawing direction (to flip its appearance in the visualization), its stacking order (to reorder where it appears in the visualization, and even duplicate it (so one can strategically copy and place the same tear next to several others in the timeline).

Annotation. The analyzer should be able to annotate the video and the timeline directly. While we have simple annotation in one of our prototypes, a full annotation tool would be very handy. Ramos et. al. gives an excellent example of how annotation could be incorporated in videos and timelines [12].

Assisted Analysis. While the placement of a slit tear is dependent on a user, there is potential to exploit automated techniques, such as Markov models (e.g. [10]), statistical methods (e.g. [2]), or topographical tools (e.g. [7]) to augment this analysis. The inclusion of such methods would help make our system a proper video analysis tool.

7. Conclusion

Tear-based video slicing is a general and powerful user-driven video analysis and exploration technique. It is cheap to use, works with off-the-shelf equipment, and no site preparation. It can be fruitfully applied in many domains where exploration and analysis of stationary video data is required. It works in real time over live and stored video. We have illustrated several examples of its use with several application genres. We demonstrated that it is possible to study event-based occurrences in the video data, and more importantly, pattern-level occurrences. In the future, we intend to evaluate the effectiveness of this tool for novice users in realistic tasks.

8. Video

Static images are a poor means for illustrating a highly dynamic and interactive system. Consequently, a video illustrating the system is available at http://grouplab.cpsc.ucalgary.ca/: select Videos and then iLab Video Reports.

9. ACKNOWLEDGMENTS

Thanks to Michael Nunes, Carl Gutwin and Sheelagh Carpendale. Research is funded by Nectar NSERC Research Network, and the NSERC/iCORE/SMART Chairs in Interactive Technologies.

10. REFERENCES

[1] Begole, J., Tang, J. C., Smith, R. B., and Yankelovich, N. 2002. Work rhythms: analyzing visualizations of awareness histories of distributed groups. *Proc ACM CSCW,* 334-343.

[2] Chung, M. G., Lee, J., Kim, H., Song, S. M.-H., and Kim, W. M. 1999. Automatic segmentation based on spatio-temporal features. *Journal of Korea Telecom*, 4 (1), 4-14.

[3] Davidhazy, A. Slit-scan photography. School of Photographic Arts and Sciences, Rochester Institute of Technology. Accessed Mar, 2007. URL=http://www.rit.edu/~andpph/text-slit-scan.html.

[4] Del Mar Thoroughbred Club. Accessed Nov, 2007. URL=http://www.dmtc.com/racinginfo/photofinishes/.

[5] Elliot, E. 1993. Watch, grab, arrange, see: Thinking with motion images via streams and collages. *MS Thesis in Visual Studies*, January, MIT.

[6] Fels S., Lee, E. and Mase, K. 2000. Techniques for interactive video cubism. *Proc ACM Multimedia,* 368-370.

[7] Guimarães, S. J. F., Couprie, M., Araújo, A., and Leit, N. J. 2003. Video segmentation based on 2D image analysis. *Pattern Recognition Letters*. 24, 947-957.

[8] Komlodi, A. and Marchionini, G. 1998. Key frame preview techniques for video browsing. *Proc ACM Conference on Digital Libraries (DL '98)*, 118-125

[9] Levin, G. An informal catalogue of slit-scan video artworks. Accessed Nov, 2007. URL=http://www.flong.com/writings/lists/list_slit_scan.htm.

[10] Ngo, C. W., Pong, T. C., and Chin, R. T. 1999. Detection of gradual transitions through temporal slice analysis. *Proc IEEE CVPR*, 36-41.

[11] Nunes, M., Greenberg, S., Carpendale, S. and Gutwin, C. 2007.What did I miss? Visualizing the past through video traces. *Proc ECSCW'07 European Conf on Computer Supported Cooperative Work*, Springer-Verlag.

[12] Ramos, G. and Balakrishnan, R. 2003. Fluid interaction techniques for the control and annotation of digital video. *Proc ACM UIST '03*, 105-114.

[13] Silk, G. Hammer Thrower. Image reproduced from National Gallery of Australia gallery. Accessed March, 2007. URL=http://www.nga.gov.au/Silk/Gallery.htm.

[14] Yeung, M. M., and Yeo, B-L. 1997. Video visualization for compact presentation and fast browsing of pictorial content. *IEEE Transactions on Circuits and Systems for Video Technology*. 7 (5), 771-785, IEEE.

Exploring the Role of Individual Differences in Information Visualization

Cristina Conati, Heather Maclaren
University of British Columbia
2366 Main Mall, Vancouver, BC
conati@cs.ubc.ca

ABSTRACT
In this paper, we describe a user study aimed at evaluating the effectiveness of two different data visualization techniques developed for describing complex environmental changes in an interactive system designed to foster awareness in sustainable development. While several studies have compared alternative visualizations, the distinguishing feature of our research is that we try to understand whether *individual user differences* may be used as predictors of visualization effectiveness in choosing among alternative visualizations for a given task. We show that the cognitive ability known as *perceptual speed* can predict which one of our target visualizations is most effective for a given user. This result suggests that tailored visualization selection can be an effective way to improve user performance.

Author Keywords
Evaluation of visualization techniques; individual differences.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

1. INTRODUCTION
In recent years, information visualization has been gaining importance as a means of managing the often overwhelming amount of digital information available to users. From generic search engines to specialized software in areas as diverse as bioinformatics, economics and the social sciences, many applications need to be able to help users understand and manipulate bodies of data with various degrees of complexity. Research in information visualization strives to provide graphical representations that can help deal with this complexity. However, despite the many attempts to identify mappings between user needs and effective visualizations (e.g. [6], [4]) results are still partial and often conflicting (see [7]). Most of these attempts have been based on the assumption that it is possible to identify optimal visualizations given type and amount of data to be visualized as well as nature of the perceptual task involved. We argue that this assumption is one of the reasons for the lack of consistency in findings, and that there are other factors that determine a visualization's effectiveness. In particular, in this paper we explore the hypothesis that the mapping between user needs and effective visualizations is influenced by individual user differences. Given a task and corresponding data, different visualizations may work best for different users, given user traits such as cognitive skills, knowledge and personal preferences.

There is already anecdotal evidence in the literature that different users may have different visualization preferences [2], and several studies have linked individual differences to visualization effectiveness for search and navigation tasks in complex information spaces [e.g., 13,14,15]. Velez et al. [9] have shown that cognitive measures related to spatial ability correlated with performance accuracy in performing 3D mental rotations supported by 2D visualizations. Our research extends these results by analyzing how spatial abilities and other user-specific traits affect performance on two different visualizations for interpreting geographical data.

One of the ways in which visualization methods are used within Geographical Information Systems (GISs) is to show how an area of interest will change over time. Georgia Basin Quest (GB-Quest) and QuestVis, both developed at the University of British Columbia, use different visualization methods to illustrate how a particular geographical region (the Georgia Basin in British Columbia, Canada) would change in 40 years time depending on the behaviours of its inhabitants. Anecdotal evidence (i.e., feedback from participants in environmental workshops that used GB-Quest and observations from pilot studies on QuestVis), suggests that the effectiveness of the visualization methods used by these two systems may depend on the user viewing them. If these observations were to be empirically confirmed, they could have

AVI'08, 28-30 May , 2008, Napoli, Italy

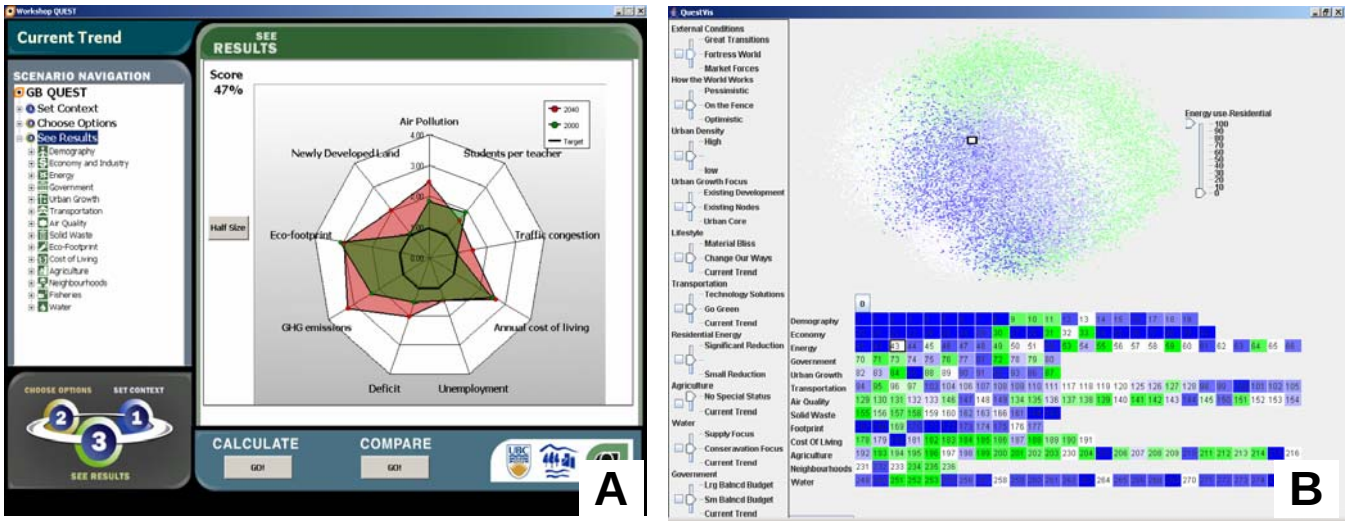


Figure 1: Screenshots from GB-Quest (A) and QuestVis (B)

important implications for research in information visualization, as they open the door to the idea of *user-adaptive visualizations*. These are visualizations that are tailored in real-time to the needs of each individual user, where these needs derive from specific user traits as well as user tasks. Thus, we decided to run a formal study to test the role that individual differences may play in the effectiveness of the visualizations used by GB-Quest and QuestVis.

In the rest of this paper, we first discuss related work. Next, we introduce GB-Quest and QuestVis and the visualization methods that they use. We then describe the study and its results. We conclude with a discussion of the implications of these results and plans for future work.

2. RELATED WORK

While most of the research on the factors that define visualization effectiveness has focused on properties of the data to be visualized or the tasks to be performed, some studies have started considering user individual differences as a promising avenue of investigation.

Most existing research in this area has focused on exploring the link between individual differences (mostly related to spatial abilities) and visualization effectiveness for information retrieval and navigation in complex information spaces [e.g. 13, 14, 15]. Baldonado et al. [2] cite differences in user profiles as one reason to have multiple individual views available in information visualization. Velez et al. [9] explored the link between five spatial abilities (selected from the Kit of Factor-referenced Cognitive Tests [5]) and proficiency in a visualization task involving the identification of a 3D object from its orthogonal projections. The data analysis in this paper is mostly based on correlations and thus does not provide insights into the actual predictive power of the target spatial abilities. It does, however, provide initial evidence that cognitive abilities may affect visualization effectiveness in a data interpretation task.

In [3], Brusilowsky et al. explore the idea of *adaptive visualizations* that are automatically tailored to the user's knowledge in the context of an educational system. This system visualizes available practice problems based on their similarity, and adaptively adds icons to each problem to indicate how suitable they are for the knowledge level of the current user. While Brusilowsky et al. [3] adapt the content of the visualization but maintain a fixed visualization technique, we investigate whether individual differences exist that may require selecting alternative visualizations for different users.

3. VISUALIZING ENVIROMENTAL CHANGES WITH GB-QUEST AND QUESTVIS

3.1 GB-Quest

Georgia Basin Quest (GB-Quest) [8] was designed to bridge the gap between scientific research, policy making and public engagement. It has been used in workshops during which residents and policymakers are asked to identify environmental strategies to achieve their ideal future for the Georgia Basin region. These strategies are inputs to GB-Quest, which computes their effects and produces a scenario describing the environmental changes that they will cause in the region in 40 years.

In GB-Quest, each scenario is characterized by a set of 294 indicators, grouped into 9 high-level variables that represent the most salient indicators of the economic, environmental, and social health of the region (e.g. government deficit, traffic congestion, student per teacher, air pollution). GB-Quest's main tool to visualize changes to the region is the *radar graph* (see right panel in Figure 1A), which illustrates

changes to the high-level variables between the present and a 40-year horizon.

In the radar graph, the values of each variable in both the present time and in 40 years are displayed simultaneously along the corresponding radial line in the graph, using different colors (red for the present and green for the future in the actual application; dark grey and light grey respectively, if this paper is printed in black and white). Variable values increase with the distance from the center of the graph. Because each variable represents an indicator that is inversely correlated with quality of life, values closer to the center are more desirable.

The user is required to make a visual comparison between the two values for each variable in order to determine how it has changed. For instance, in Figure 1A the amount of newly developed land decreased in 40 years, while the annual cost of living remained unchanged. The user must also make a visual comparison of the areas outlined by all the variable values for each year in order to determine any overall trend in the current scenario. Large areas in the lighter color represent "good" scenarios in which most variables have decreased, while large areas in the darker color represent "bad" scenarios in which most of the values have increased.

Users can input alternative strategies to try and improve the scenario (i.e., the environment's evolution in the next 40 years). GB-Quest, however, does not have a dedicated mechanism to support direct scenario comparison; that is, there is no easy way to view alternative radar graphs together in order to compare them.

User surveys conducted after the workshops indicated that users generally appreciated GB-Quest as a tool to increase their environmental awareness. However, anecdotal evidence indicates that the radar graph visualization is intuitive for some users, but rather incomprehensible for others.

3.2 QuestVis

QuestVis [10] (see Figure 1B) is a redesign of GB-Quest aimed at facilitating the exploration and comparison of different scenarios. Here we describe the main differences with GB-Quest. Two were introduced in order to facilitate scenario generation. The first difference is that QuestVis reduces input choices to a set of 11 sliders with 2 or 3 possible choices each, replacing the more cumbersome input mechanism available in GB-Quest. The second difference is that, while GB-Quest waits until the user has finalized all choices before calculating the results, QuestVis uses a pre-computed database to show the effect of each choice as it is made. This highly reactive behavior was introduced to improve the user's sense of the connections between input choices and their effect on the region.

Another difference (the most relevant to this paper) is that QuestVis uses a new visualization to show changes to the region. This technique, known as the Multiscale Dimension Visualizer (MDV) [10], is shown in the right panel in Figure 1B, and was introduced primarily to facilitate a scenario's analysis and comparison. The only way to observe scenario changes at the level of individual indicators in GB-Quest is to abandon the radar graph view and access visualizations based on bar graphs that show the changes at the level of small subsets of related indicators. In contrast, QuestVis uses the MDV to represent all 294 of the individual indicators simultaneously. The normalized value of each indicator is color encoded to enable a compact representation of the results. The color scale uses blue and green to represent, respectively, an increase and a decrease in value relative to the present-day value. The saturation of the color represents the extent of the change, normalized relative to the minimum and maximum possible values for that particular indicator across all scenarios. The more saturated the color, the larger the change from the current value. Thus, unlike with GB Quest, the user is not required to perform a mental comparison of values in order to determine how much each variable has changed over time[1].

Figure 2: QuestVis's MDV aggregated view (left) and two aggregated views side-to-side (right)

QuestVis can also produce a summary view of a scenario by (i) aggregating the 294 indicators in the high-level variables used by GB-Quest; and (ii) using the same color conventions to represent direction and magnitude of changes over these variables, as shown on the left of Figure 2. Using this aggregated view (called *colored boxes* from now on), the user can compare multiple scenarios side by side (see Figure 2, right), something that would be hard to do with the radar graph, especially when comparing more than 2 scenarios.

[1] The numbers in the boxes in Figure 1 and 2 relate to variables, they do not represent additional information about the scenario.

Table 1. The cognitive abilities tested in our study

Name		Description
Visual Memory (VM)		The ability to remember the configuration, location, and orientation of figural material
Spatial Visualization (SV)		The ability to manipulate or transform the image of spatial patterns into other arrangements
Perceptual Speed (PS)		Speed in comparing figures or symbols, scanning to find figures or symbols, or carrying out other very simple tasks involving visual perception
Disembodiment (D)		The ability to hold a given visual percept or configuration in mind so as to disembed it from other well defined perceptual material
Need for Cognition (N4C)		The tendency to engage in and enjoy tasks that require thinking
Learning Style (LS)		Preferences for the manner in which information is received and learned, i.e., preference for:
	Active/Reflective (A/R)	• proactive participation to the learning process vs. passive reception of instruction
	Sensing/Intuitive (S/I)	• concrete, well defined learning objects/strategies vs. discovering possibilities and relationships
	Visual/Verbal (V/V)	• receiving information visually vs. verbally
	Sequential/Global(S/G)	• learning in linear, logical steps vs. learning in large non-methodical jumps, absorbing material almost randomly and then suddenly "getting it."

4. COMPARING THE RADAR GRAPH AND THE *COLORED BOXES* VIEW

The MDV technique in QuestVis has two obvious advantages compared with GB-Quest: it supports the explicit comparison of alternative scenarios and it includes a flexible, integrated mechanism to observe scenarios at different levels of detail.

However, it is not obvious that the MDV colored boxes visualization is better at providing an overview of the changes in terms of high-level variables. As with the radar graph, observations from informal pilot studies indicate that not all users find the colored boxes intuitive. These observations suggest the hypothesis that the colored boxes and the radar graph could be used more effectively as alternative visualizations within the same system, and that the choice between the two may need to be based on knowledge of individual user differences.

The study described in the rest of the paper was designed to shed light on this issue. In particular, we wanted to investigate the following questions: (1) Is one visualization more effective than the other for all users? (2) If not, can the most effective visualization for a given participant be predicted from specific individual traits? In the rest of this section, we first describe the individual traits we chose to investigate in our study. We then describe the tasks and design details of the study, and finally discuss the data analysis and related results.

4.1 Individual traits explored in the study

A variety of individual traits could influence a user's perception of different visualizations, including cognitive abilities, expertise with visualization techniques and affective elements such as personality. For this study, we chose to focus on cognitive abilities. We selected four that have been previously linked to visualization capabilities [9], as well as five additional abilities that, to our knowledge, have never been considered in the context of information visualization research. The four previously explored abilities come from Velez et al.[9] (see Section 2) and are listed in the first four rows of Table 1. The rest of the rows show the new cognitive abilities that we introduced, i.e., *need for cognition* along with four indicators that define a person's *learning style*.

4.2 Experimental tasks

Because we want to study the effectiveness of the radar graph and colored boxes as alternative means to visualize the same information, an evaluation involving interaction with the complete system would be confounded by the different interaction styles and functionalities of QuestVis and GB-Quest. Therefore, for this study we used a series of basic tasks that would allow us to compare user performance with each visualization in isolation from the system in which it is embedded. The tasks were based on a set of low-level analysis tasks that Amar et al. [1] identified as largely capturing people's activities while employing information visualization tools for understanding data. In consultation with one of the researchers involved in the design of QuestVis (who was also highly familiar with GB-Quest and its radar graph visualization), we chose a subset of Amar et al.'s tasks that are most relevant for interacting with visualizations of scenarios for environmental changes. Tasks were left out when they required knowledge of absolute values instead of unmarked scales (e.g., "Retrieve value" of a variable in a single scenario, "Determine Range" of a variable's values), because neither the radar graph nor the colored boxes represent absolute values. The

Table 2: The ten task types in the study and related sample questions

One Scenario

Task	Sample Question
Filter	Find the variables that increased in the scenario.
Compute derived value	Taken as a whole, how much did the scenario increase or decrease?
Find extremum	Name the variable that decreased the most.
Sort	Rank the following variables, putting the greatest increase first.
Characterize distribution	Describe the distribution of values within the scenario (choose all options that you think apply).

Two Scenarios

Task	Sample Question
Retrieve value	For each of the following variables, do you think it is larger in the scenario on the left or on the right.
Filter	Find the variables whose values decreased in the scenario on the right compared to the scenario on the left.
Compute derived value	As a whole, how much did the scenario on the right increase/decrease compared to the scenario on the left.
Find extremum	Find the variable whose value in the scenario on the right decreased the most compared to the one on the left.
Sort	Rank the following variables in terms of greatest increase in the scenario on the right compared to the scenario on the left.

tasks were framed as a series of questions that participants had to answer while viewing a single scenario or pair of scenarios, and they are listed in Table 2. The scenarios and corresponding questions were presented via automated software developed specifically for this study. Note that we changed the radar graph's original red-green color scheme to avoid complications due to color-blindness.

Participants repeated each of the tasks in Table 2 on four different scenario "types" that varied in terms of the skewness of the distribution of variable values. Two of the four types are shown in Table 3 (with only the Radar graph for brevity). Distribution skew was varied to make participants perform each task type at different levels of difficulty. For instance, performing the sorting task "Rank the following variables, putting the greatest increase first" is easier with the spiky distribution shown at the top of Table 3 than with the uniform distribution shown at the bottom of the table.

4.3 Design and procedure

The study was a within-subject factorial design with visualization type (Radar Graph or Colored Boxes) as the primary factor and visualization order as a between-subjects control variable. There were 45 participants, 18 male, 27 female, all students at a local university. Students were paid $30 for their time and came from a variety of departments, including commerce, engineering, and dentistry. The experiment was designed and pilot-tested to fit in a single session lasting at most 2 hours (average session length was 1h 45'). Participants took part in the study in small groups of 1 to 4 people. The study took place in a room set up with 4 separate stations, each equipped with a laptop computer with the testing software pre-loaded for immediate use.

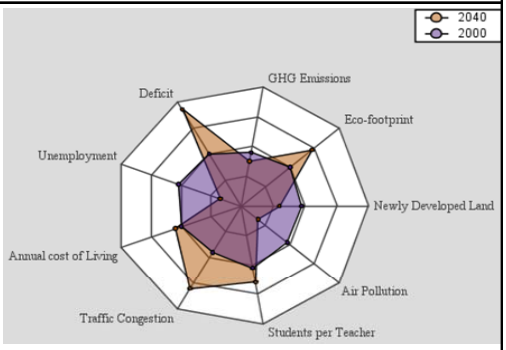
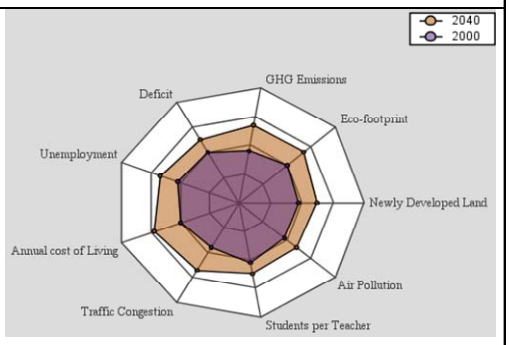
Each session started with the participant taking pencil-and-paper tests for the six cognitive traits in Table 1, with a 5-minute break after the first three tests. For spatial abilities we used the Kit of Factor-referenced Cognitive Tests [5]; for need for cognition we used the test described in [12]; and for learning style we used the the ILS inventory [11]). After completing the tests, participants took a second break, received a brief training on the two visualizations and the testing software, and finally started interacting with the software. Each participant performed a block of basic visualization tasks twice, once with each visualization method. A task block was structured as follows: First, subjects were presented with a single scenario and had to answer the five questions listed in the "One Scenario" portion of Table 2. Then subjects were presented with two scenarios and had to answer the five questions listed in the "Two Scenarios" portion of Table 2. Participants repeated the above cycle four times, once for each of the four distribution types described earlier. Visualization order was fully counterbalanced to account for learning effects, making visualization order a between-subject control variable in our design

4.4 Measures

We measure visualization effectiveness in terms of accuracy on the visualization tasks, computed as the number of correct answers generated for the questions in the testing software. More specifically, accuracy on each of the 10 task types in Table 2 is computed by summing all

correct answers to the related questions across the four distribution types.

Table 3: Two of the four distributions in the study

Spiky	
Uniform	

5. DATA ANALYSIS AND RESULTS

5.1 Cognitive abilities as predictors of the most effective visualization

Recall that the main goal of this study is to verify whether one of our two visualizations dominates the other over any of the target tasks, or whether best visualization depends on one or more of the cognitive abilities in Table 1.

To answer these questions, we use an analysis based on General Linear Models (GLM). We started the analysis by running a repeated-measures 2 (visualization type) by 10 (task) by 2 (visualization order) GLM with the 9 cognitive test measures as covariates and task accuracy as the dependent variable. In this and all subsequent GLM, we applied the Greenhouse-Geisser adjustment for non-spherical data. We report statistical significance at the 0.05 level (unless otherwise specified), as well as partial eta-squared (η^2), a measure of effect size. To interpret this value, .01 is a small effect size, .06 is medium, and .14 is large [7].

The salient findings from this first GLM include:

- No significant main effect of visualization type on accuracy
- No main or interaction effects of visualization order, indicating that the counterbalancing of visualization presentation successfully avoided ordering effects
- A significant interaction of visualization type with the cognitive ability related to *perceptual speed* (PS) ($F(1,34)=4.8$, $p = 0.035$, $\eta^2= .124$). This interaction means that perceptual speed has a significant effect in determining which visualization generates better accuracy for each individual user. In other words, PS is a significant predictor of the *difference* in accuracy between the two visualizations.
- No other cognitive ability had a significant interaction with visualization type
- A significant main effect of task ($F(9,26)= 5.951$, $p < 0.01$, $\eta^2= .149$), indicating that accuracy outcome significantly varies with task type.

Because only perceptual speed (PS from now on) generated a significant interaction with visualization type, we will focus on this ability in the rest of the analysis. To better understand the relationships among visualization type, perceptual speed and task type, we ran a series of GLMs with task accuracy as the dependent variable, visualization type as the main factor and perceptual speed as covariate, one for each of the ten tasks in Table 2 (we applied a Bonferroni adjustment for 10 post-hoc comparisons, bringing the alpha level for significance down to 0.005). Note that we no longer include visualization order in the analysis, because it did not show any significant effect in the overall model.

For nine of the tasks, there was no significant difference in accuracy between the two visualizations, and no significant effect of perceptual speed on accuracy. In contrast, the GLM for accuracy on the "Compute Derived Value" task with two scenarios generated a significant interaction between visualization type and perceptual speed, with a high effect size ($F(1,43) = 14.442$, $p < 0.005$, $\eta^2= .251$).

Recall, from Table 2, that "compute derived value" with two scenarios requires users to compare the scenarios in terms of how much they changed as a whole. The eta-square value reported above indicates that variation in PS can explain 25.1% of the variance in accuracy difference between the two visualizations for this task. An analysis of the relationship between PS and accuracy with each visualization on this task shows that PS is a significant negative predictor of accuracy with radar graph (β correlation coefficient = -.475, $t = -3.6$, $p = .001$). It is also positively correlated with (although not a significant predictor of) accuracy with the colored boxes for this task. Figure 3 shows the interaction between PS and visualization type if PS is converted to a categorical variable with values HIGH and LOW determined by the median of the original covariate

These results indicate that users with high PS will be more accurate when comparing scenarios in terms of how much they changed as a whole if they use the colored boxes rather than the radar graph, and that PS can be used as a factor to decide which of our two target visualizations will be more effective for accomplishing this particular value-derivation

task. This predictive ability is especially important in the context of systems focused on sustainable development like GB-Quest and QuestVis, because evaluation of overall environmental changes is a focal concept for these systems

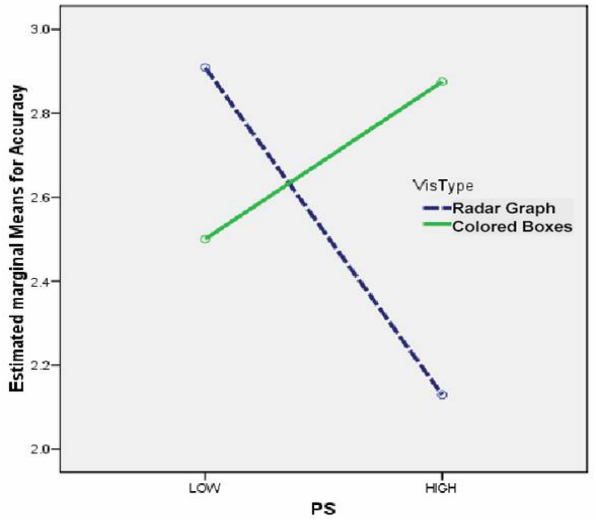


Figure 3: Interaction between PS and Visualization Type

and was one of the main targets in the GB-Quest workshops described earlier. Thus, adaptive visualization selection based on this predictor could have direct applications in the activities that GB Quest or QuestVis will be used for.

Table 4: Linear Regression results for Individual Accuracies. "Pred." stands for "Predictors"; "$R^{2"}$ stands for adjusted R^2

One Scenario				
Task	**Measure**	**Pred.**	**β**	**Linear Model**
Sort	AccCB	N4C, V/V;	.445 -.282	F=6.91, p=.003, R^2= .150
Characterize Distribution	AccCB	SV	.434	F=9.97, p=.003, R^2= .169
Two Scenarios				
Filter	AccCB	VM	.443	F=9.91, p=.003, R^2= .168
Compute Derived Value	AccRadar	PS A/R	-.47 -.36	F=8.33,p=.001, R^2= .25

Although we don't have a conclusive explanation for the direction of the relationships that we found among perceptual speed, accuracy with the radar graph and accuracy with the colored boxes, our results for the radar graph visualization are consistent with the negative correlation found by Velez et al. between perceptual speed and accuracy in deriving 3D shapes from 2D projections. In the radar graph, scenario change is derived by performing a visual comparison of the areas outlined by the individual variables. Thus, both this task and the derivation of 3D shapes from 2D projections studied by Velez et al. require comparing 2D shapes. This commonality is a plausible explanation of why we found the same negative correlation as Velez et al. between perceptual speed and task accuracy.

5.2 Cognitive abilities as predictors of accuracy with individual visualizations

While the main goal of this study was to understand whether user cognitive abilities can predict which visualization is most effective for each user, there is also value in exploring whether these abilities can predict task accuracy with each visualization. Towards this end, we ran a series of 20 linear regression analyses for each of the two available visualizations, with accuracy on each task as the dependent variable and the cognitive test scores as predictors. Table 4 summarizes the results of this analysis as follows. The first column lists all the tasks for which we found a significant ($p < 0.0025$ with adjustment for multiple tests) or marginally significant ($p < 0.005$)[2] linear model for predicting accuracy. The second column reports, for each task, which accuracy measure we can predict (AccCB = Accuracy with Colored boxes; ACCRadar = Accuracy with Radar). The third column reports the significant or marginally significant predictors in the model. The fourth column lists their correlation coefficients. The fifth column summarizes the model statistics.A relevant result from Table 4 is that the new cognitive abilities we added in this study compared to the study by Velez et al. (2005) (i.e. need for cognition and the four linear scales for learning styles) do play a role as predictors of visualization accuracy. In fact, they are the only predictors for the visualization accuracy of sorting with one scenario with the colored boxes (see first row in Table 4). Furthermore, they are comparable to the spatial abilities from the Velez et al. study in terms of the amount of accuracy variance they can explain (see R^2 values in the 5th column of Table 4).

It should be noted, however, that the variance accounted for by all our linear models is rather low, ranging from 12.9% to 25%. This result suggests that other user traits should be explored to understand how individual differences affect visualization effectiveness. One promising candidate is expertise with visualizations, which we could not include in our study due to the difficulty of finding a reasonable number of visualization experts in the user population available to us.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a study aimed at exploring whether user's individual differences can be used as predictors to select the most effective visualization for different users. The study involved a variety of tasks that can be accomplished using two alternative visualizations

[2] We include results with close marginal significance because they are most likely due to not having sufficient subjects to run a linear regression with nine predictors

for representing value changes in a set of variables: one is the Radar Graph, which relies on both spatial elements (linear distance and area) and color to represent change; the other is the Multiscale Dimension Visualizer (called *Colored Boxes* in the paper), which uses primarily color.

Our data analysis shows that, while there is no significant difference in task accuracy with the two visualization for 9 of the task types performed during the study, for the 10th task type (comparing how the values of two sets of variables change as a whole) the best visualization depends on the user's Perceptual Speed (PS). That is, users with high perceptual speed perform this task better if they can see the relevant data via the colored boxes, while users with low perceptual speed perform better with the radar graph. This finding suggests the idea of having a system that can display both visualizations but that adaptively selects which one to recommend based on the user perceptual speed, if one of the tasks to be performed involves comparing overall changes in variables of interest. Information of the user's perceptual speed could be obtained at the onset of the interaction by administering to the user a test for this cognitive ability [5]. If the test is administered on-line, then its results can be automatically computed by the software and used in real time to suggest the optimal visualization.

Our data analysis also provided initial evidence that some cognitive abilities can be used as predictors for the effectiveness of each individual visualization. The predictive power of cognitive abilities related to spatial reasoning had already been identified by previous research. Our contribution lies in the identification of additional predictors not related to spatial processing, specifically *need for cognition* and measures related to learning style. These findings could be used to provide users with further automatic support to ensure effective visualization processing. While using a specific visualization, the user may receive help or clarifications from the system if the system detects that this user scores low on the cognitive abilities that predict success in using the current visualization. For instance, one of our findings was that Need for Cognition is a positive predictor of user accuracy in sorting variables with the colored boxes. If a system detects that the user has a low need for cognition, it can offer help in interpreting the visualizations during this task.

However, in order to provide effective help, the system also needs to know what type of problems a user with low scores on the relevant cognitive measures may have when using a specific visualization. We plan to investigate this issue in the context of the visualization systems discussed in this paper by running further studies specifically designed to uncover these problems. We also plan to investigate additional individual traits that may function as predictors of visualization effectiveness, including abilities related to color perception and user expertise.

ACKNOWLEDGEMENTS

This research was funded by the GEOIDE network grant P31HS20. We would like to thank John Robinson, the project leader, for his continuous support and Tamara Munzner for her help with the study design.

References

1. R.A. Amar, J. Eagan and J.T. Stasko, Low-Level Components of Analytic Activity in Information Visualization., in: 16th IEEE Visualization Conference (VIS 2005), (2005).

2. M. Baldonado, A. Woodruff and A. Kuchinsky, Guidelines for Using Multiple Views in Information Visualization., in: Advanced Visual Interfaces 2000.

3. P. Brusilovsky, J. Ahn, T. Dumitriu and M. Yudelson, Adaptive Knowledge-Based Visualization for Accessing Educational Examples., in: Proceedings of Information Visualization, (2006) .

4. W.S. Cleveland and R. Mcgill, Graphical Perception and Graphical Methods for Analyzing Scientific
Data. Science 229 (1995) 828-833.

5. R. Ekstrom, J.French, H. Harman and D. Dermen, Manual from Kit of Factor-References Cognitive Tests. 1976, Educational Testing Service (1976): Princeton, NJ.

6. J. Mackinlay, Automating the Design of Graphical Presentations of Relational Information. Transactions on Graphics 5 (2) (1986) 110-141.

7. L. Nowell, R. Schulman and D. Hix, Graphical Encoding for Information Visualization: An Empirical Study, in: IEEE Symposium on Information Visualization (InfoVis'02), (2002) 53-81.

8. J. Tansey, J. Carmichael, R. Van Wynsberghe and J. Robinson, The Future Is Not What It Used to Be: Participatory Integrated Assessment in the Georgia Basin. Global Environmental Change: Human and Policy Dimensions 12 (2) (2002) 97-104.

9. M. Velez, D. Silver and M. Tremaine, Understanding Visualization through Spatial Ability Differences., in: Proceedings of Visualization 2005.

10. M. Williams and T. Munzner, Steerable, Progressive, Multidimensional Scaling, in: InfoVis 2004,

11 R.M. Felder and L.K. Silverman, "Learning and Teaching Styles in Engineering Education," *Engr. Education*, *78*(7), 674-681 (1988),

12 Cacioppo, J. T., R. E. Petty, et al. The efficient Assessment of Need for Cognition. *Journal of Personality Assessment* 48(3): 306-307, 1984.

13. Chen C. Individual Differences in Spatial-Semantic Virtual Environments *Journal of the American Society of Information Science*, 51, 6, 2000

14. Dillon D. Spatial Semantics: How Users Derive Shape from Information Space. *Journal of the American Society of Information Science*, 51, 6, 2000

15 Allen B. Individual Differences and Cundrums of User-centered Design: Two Experiments. *Journal of the American Society of Information Science*, 51, 6, 2000

An Empirical Evaluation of Interactive Visualizations for Preferential Choice

Jeanette Bautista and Giuseppe Carenini
Department of Computer Science
University of British Columbia
201-2366 Main Mall
Vancouver BC V6T 1Z4
bautista, carenini@cs.ubc.ca

ABSTRACT

Many critical decisions for individuals and organizations are often framed as preferential choices: the process of selecting the best option out of a set of alternatives. This paper presents a task-based empirical evaluation of ValueCharts, a set of interactive visualization techniques to support preferential choice. The design of our study is grounded in a comprehensive task model and we measure both task performance and insights. In the experiment, we not only tested the overall usefulness and effectiveness of ValueCharts, but we also assessed the differences between two versions of ValueCharts, a horizontal and a vertical one. The outcome of our study is that ValueCharts seem very effective in supporting preferential choice and the vertical version appears to be more effective than the horizontal one.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI)*; I.3.6 [**Computer Graphics**]: Methodologies and Techniques—*Interaction Techniques*; H.4.8 [**Information Systems Applications**]: Types of Systems—*Decision Support*

General Terms

Design, Experimentation, Human Factors

Keywords

Visualization techniques, preferential choice, empirical evaluation, user studies

1. INTRODUCTION

Developing effective interactive visualization interfaces requires a possibly long process of iterative design, in which task analysis, analytical evaluation and user studies are successively applied. We have followed this methodology in the development of an interactive visualization framework to support preferential choice: the process of selecting the best option out of a set of alternatives. Preferential choice has been extensively studied in decision theory as many critical decisions for individuals and organizations are often framed as preferential choices. For instance, selecting a house to rent or buy, deciding who to hire, selecting the location of a new store or deciding where to spend your next vacation are all examples of preferential choices. When people are faced with such decisions, they look for an option that dominates all the others on all aspects they care about (objectives in decision theory). However, such an option often does not exist. For instance, when selecting a house within a specified price range, you may find one that is situated at the ideal location but does not have all the amenities you seek. In this case you will have to consider the tradeoffs. People are generally not very effective at considering tradeoffs among objectives, and require support to make this process easier [9].

According to prescriptive decision theory, effective preferential choice should include the following three distinct interwoven phases. First, in the model construction phase, the decision maker (DM) builds her decision model based on her objectives: what objectives are important to her, the degree of importance of each objective, and her preferences for each objective outcome. Secondly, in the inspection phase, the DM analyzes her preference model as applied to a set of alternatives. Finally, in sensitivity analysis, the DM has the ability to answer "what if" questions, such as "if we make a slight change in one or more aspects of the model, does it effect the optimal decision?" [9].

In the development of interactive tools for preferential choice, we argue that full support for - and fluid interaction between - all three phases are essential in making good decisions.

In [8] we presented ValueCharts (VC), a set of interactive visualization techniques to support preferential choice. VC in its original form was designed by mainly focusing on supporting the model inspection phase. Furthermore, the design of the interface relied on a rather simple task analysis exclusively based on decision theory.

In [5], we described the second major iteration in the development of VC. We presented the Preferential choice Visualization Integrated Task model (PVIT): a much more sophisticated compilation of domain-independent tasks that considers all aspects of preferential choice, some new ideas from decision theory, and more importantly, an integration

AVI '08, 28-30 May , 2008, Napoli, Italy.

Figure 1: Horizontal ValueCharts - VC+H

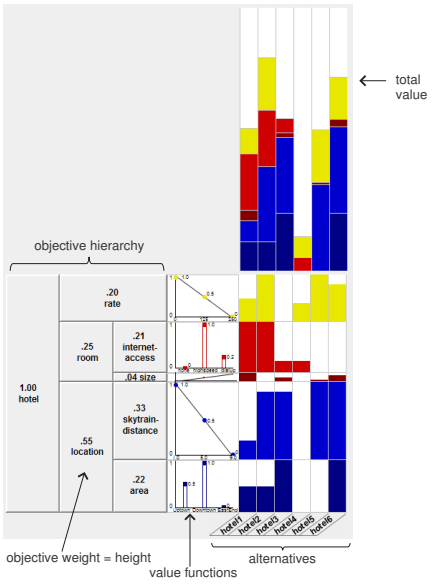


Figure 2: Vertical ValueCharts - VC+V

of task frameworks from the area of Information Visualization (InfoVis). For a detailed account of how tasks from literature in InfoVis and Decision Theory were integrated into our model, please see [5].

A new version of VC, called VC+, was developed to effectively support all the tasks included in PVIT (See [5] for the rationale behind the redesign). We also used the PVIT model to compare analytically VC+ with other existing tools for preferential choice [3, 2, 6]. VC+ appears to clearly dominate all its competitors (30% more effective). We developed two alternative versions of VC+. In VC+H the information is displayed horizontally, while in VC+V the information is displayed vertically. The PVIT model suggests that VC+V should be more effective than VC+H.

In this paper, we present our extensive empirical testing of VC+. In our evaluation approach we follow [16]. In particular, as advocated by Plaisant, we give subjects real problems, we ground the evaluation in a comprehensive task model and we measure both task performance and insights.

Since the analytical evaluation indicated that the other tools fare considerably worse than VC+ in supporting the tasks of the PVIT model, we gave low priority to a comparison study. Instead, we performed a comprehensive user study, grounded in the PVIT, to verify the usefulness and effectiveness of VC+, as well as focused on the assessment of the differences between the horizontal (VC+H) and the vertical (VC+V) versions.

The key contribution of this paper is that we applied several distinct approaches in order to achieve a more comprehensive assessment. First, we performed a quantitative, controlled usability study to see how effectively users performed the low-level tasks of the PVIT. Second, we qualitatively observed subjects using the tool in a real decision-making context of their choice (out of three possible ones). The subjects then answered a number of questions regarding their experience with VC+ in the decision-making process. In addition, we attempted to measure the users' insights in the decision problem. And finally, we used interaction logging throughout the experiment for further study. By triangulation of methods, we aimed to more fully understand the DM's experience in using VC+ (and VC+H versus VC+V) to perform preferential choice.

As a preview of the paper, we first briefly summarize VC+ and the PVIT model. We then describe our evaluation methodology and experimental design. Next, we present the controlled experiment on the PVIT low-level tasks. Finally, we describe the exploratory study to assess the effectiveness of VC+ in terms of user experience and insights.

2. VALUECHARTS+ AND THE PVIT MODEL

2.1 ValueCharts+ (VC+)

We developed two variations of VC+. In VC+H the information is displayed horizontally (Figure 1), while in VC+V the information is displayed vertically (Figure 2). We will describe the general features of VC+ by referring to the vertical version and then we will examine the differences between the two versions.

VC+ is a set of interactive visualization techniques for preferential choice. It supports the DM in the construction, inspection and sensitivity analysis of a DM's preference model as an Additive Multiattribute Value Function (AMVF) [1]. In an AMVF the DM's objectives are hierarchically organized. In VC+ this hierarchy is displayed as an exploded stacked-bar, see Figure 2 left-bottom quadrant. The vertical height of each row indicates the relative weight assigned to each objective (e.g., *size* is much less important than *internet-access*). Each column represents an alternative, thus each cell portrays an objective corresponding to an alternative (bottom-right quadrant). The amount of filled color relative to cell size depicts the alternative's preference

[1]For a detailed description of AMVF and VC+, see also [8, 5]

value of the particular objective (e.g., the *rate* for *hotel4* is bad, *hotel3* is worst). The values are then accumulated and presented in a separate display in the form of vertical stacked bars, displaying the resulting score of each alternative (top-right quadrant).

Several interactive techniques are available in VC+ to further enable the inspection and sensitivity analysis of the preference model. For instance, center-clicking on an alternative label displays the corresponding domain values. Double-clicking on the row heading ranks the alternatives according to how valuable they are with respect to the corresponding objective. As an example of sensitivity analysis, an objective weight can be changed by sliding the row headings to the desired weight.

The two versions of VC+ we have developed are informationally equivalent. They differ only in how the same information is displayed and in how it can be accessed. As shown in Figure 2, besides the different orientation the main difference between VC+V and VC+H is that VC+V allows for persistent display of the AMVF's component value functions (Figure 2 center bottom-half). In an AMVF, there is one component value function for each objective, and it specifies how valuable different levels of the corresponding objective are to the DM. For instance, the lower the *skytrain distance* the better. As discussed in [5] including these functions persistently in VC+H would be visually misleading and possibly confusing, so in VC+H component value functions are only accessible on demand.

At first glance, it appears that offering a persistent view on the component value function presents only advantages. The DM should be able to more effectively inspect tradeoffs among objectives as the range of their levels is readily visible and objective names are also more readable, because the label width is now mainly affected by the depth of the tree instead of by the number of objectives. Yet, since the functions do take up some screen real estate, they can become less readable and useful if the number of objectives increases, and more importantly making them permanently visible requires a vertical orientation that may negatively affect some PVIT tasks.

An important goal of the user study presented in this paper is to clarify whether the different orientation and persistent component value functions affect the DM performance in using VC+V versus VC+H.

2.2 The Preferential choice Visualization Integrated Task model (PVIT)

The Preferential choice Visualization Integrated Task model [5] is a framework for the design and evaluation of information visualizations for preferential choice (See Figure 3). The PVIT model starts top-down from the task that defines preferential choice: *to select the best alternative*. The first decomposition is into the three main phases of preferential choice: *construction*, *inspection*, and *sensitivity analysis* of the preference model applied to the set of alternatives.

The next level below incorporates two task taxonomies from the area of Information Visualization. The first is a classic: Ben Shneiderman's task by data type taxonomy (TTT) [18], which includes the tasks from his information-seeking mantra of "*overview first*, *zoom and filter*, then *details on demand*", as well as *relate*, *history*, and *extract*. In more recent literature [1], Amar and Stasko recognize the need for information visualizations to not only support representation of data, as the TTT does well, but also facilitate higher-level analytical tasks such as decision-making and learning. To bridge what they call "analytical gaps" (the gaps between representation and analysis), we incorporated their high-level *Knowledge Tasks* (e.g., *concretize relationship*) into our PVIT model by expanding the *relate* task from the TTT (see center of Figure 3).

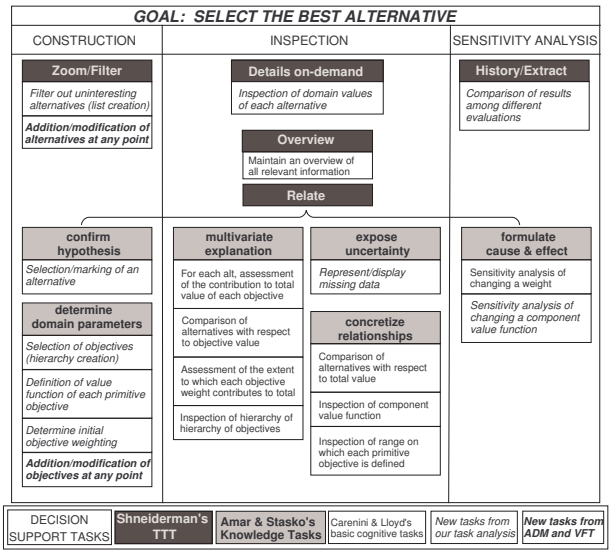


Figure 3: The PVIT model

In further refining PVIT into primitive tasks (leaf nodes), we go back to decision theory by first considering the original set of basic decision tasks for preferential choice proposed by Carenini and Lloyd [8]. Then, we augment this set so that all the generic knowledge tasks are instantiated and also by taking into account relatively recent ideas from decision theory (i.e., Value Focused Thinking [9]).

3. EVALUATION METHODOLOGY

Our evaluation methodology relies on [16], in which several guidelines for effective evaluation of interactive information visualizations are proposed. First, it is crucial that the empirical study matches tools with users, tasks, and real problems. In our study, we give subjects real preferential choices and we ensure the decision situation is of interest to the subject by letting her choose among three possible scenarios. Second, the evaluation should be grounded in a comprehensive task model. We follow this guideline by relying on the PVIT. Third, evaluation should not be limited to task performance but should also try to measure discovery and insights. Our evaluation comprises two parts. In Part A, we took a quantitative approach by performing a controlled usability study to see how users performed the primitive tasks of the PVIT. In Part B, we followed a more qualitative approach by observing subjects using the tool in a real decision-making context. In this second part of the study, we attempted to measure the users' insight in the decision problem.

Once the subjects had completed Part B, they filled out a questionnaire regarding their experience with VC+ in the decision-making process. Interaction logging were also col-

lected throughout the experiment for further study.

Since we were comparing two versions of the same tool, we decided on a between-subjects experimental design to avoid the obvious learning effect that would come with a within-subjects design. Each subject was assigned to either VC+V or VC+H interface.

Subjects were recruited through the Reservax [2] online experiment reservation system. Prior to beginning the experiment, each subject read, signed and dated a consent form and filled out a pre-study questionnaire. 20 subjects, all students at UBC, of age ranging from late teens to 50+, agreed to spend 60 minutes with our experiment and receive $10 in compensation. Of the sample, 8 were male. All subjects were fairly computer proficient, ranging from 10 - 50+ hours per week. Each subject worked with only one VC+ interface: 10 subjects worked with VC+V and the other 10 worked with VC+H.

We found a good match in the grouping of the subjects in each treatment. Both groups had the same breakdown in sex and English proficiency, and the average computer use was very close. There was a slight difference in average age group: VC+V subjects were a younger group overall, half of them being less than 20 years old, and in the VC+H group, most subjects were in the 20-29 age group. All subjects had no previous exposure to formal decision analysis methods.

4. PART A: CONTROLLED STUDY

In this first part of the evaluation, we tested the hypothesis that the difference between the two orientations influences subject performance on a set of tasks form the PVIT model. In addition to the total-time-to-complete and correctness of tasks, we looked at each task individually.

4.1 Tasks

For this controlled study, we used data sets accompanied by scenarios: for training, we used the scenario of shopping for a used television set, and for testing we put the user in the situation of deciding on a hotel to stay at in Vancouver. It was assumed that the construction phase had already been completed (it is the same in both versions of VC+), and participants performed inspection tasks interspersed with sensitivity analysis tasks.

We considered the following four basic types of sensitivity analysis tasks (instances of the *formulate cause and effect* in the PVIT model): (i) What if [objx]'s weight is increased of k, and consequently [obj y]'s weight decreased of the same amount? (ii) What if [objx]'s weight is increased of k, and [all other n objectives] decreased of k/n? (iii) What if a component value function is changed in a numerical domain (e.g. money)? (vi) What if it is changed in categorical domain (e.g., neighborhood)?

We considered all nine primitive inspection tasks from the PVIT model. However, since four of the inspection tasks are implied by sensitivity analysis tasks (e.g. *Inspection of component value function* is implied when the user is asked to perform value function sensitivity analysis), we explicitly tested only the remaining five (see Table 1 for an example of these five tasks mapped to the house domain).

[2]http://www.reservax.com/hciatubc/index.php, *HCI@UBC Subject Sign-up System*

What are the top 3 alternatives according to total value?	List the 3 highest valued houses
For a specified alternative, which objective contributes to its total value the most?	For HouseX, which is its strongest factor according to your preferences?
What is the domain value of objective x for alternative y?	How many bathrooms are there in House1?
What is the best alternative when considering only objective x?	Which is the least expensive house?
What is the best outcome for a objective x?	What is the best bus-distance?

Table 1: Sample inspection tasks mapped to the House domain

4.2 Tutorial and Training

The PVIT construction tasks are assumed to help the user bridge the *Rationale* gap [1], i.e., they help the DM learn about the decision problem at hand, the model, and the decision analysis technique in general. So, although construction tasks were excluded from our evaluation, it was important to include the construction interface in the training.

The training session was performed on the TV domain. With the construction interface, the experimenter explained the objective hierarchy, the given alternative data, specifying value function, and objective weighting. After constructing the chart, the experimenter described the inspection interface in detail, covering all the types of tasks that the subject was to complete. The subject was then given a set of tasks to perform on the given model, which were task examples of the testing phase.

4.3 Procedure

After the training phase, each participant had an opportunity to ask questions for clarification before the testing phase began. Subjects were reminded that time and correctness were being measured, and that this time they did not have the opportunity to ask questions.

Once again, the experimenter walked the subject through the construction of the model (this time using the Hotel data set), but the testing did not start until after the VC+ view was in place. The subject was then given a set of tasks much like what they saw in the tutorial. The set was organized so that after each of the sensitivity analysis tasks, subjects completed a round of inspection tasks. In total, there were five rounds of inspection tasks (including one when the interface is first presented). Each subject performed each task, writing down the answer to applicable tasks that asked a question about the data.

4.4 Results

All subjects completed the procedure successfully. In terms of correctness of tasks performed, there was no significant difference between the two interfaces. In fact, there were very few mistakes made during testing and the average overall mean score for VC was 18.5, or 97.4% correct (participants scored 18.6 with VC+V and 18.4 with VC+H).

The high percentage of correctness does give us a good indication that subjects did well. However, we could not determine if the subjects performed well overall for time to complete tasks, since there is no benchmark to compare

against in this measure. Instead, we looked closely at these results to find an indication of whether one version of VC+ was better than the other for performing the tasks.

Our analysis follows a two-step approach that has been already successfully applied in Computational Linguistics [11], as well as in Human-Computer Interaction [14, 4], when two systems are compared on a relatively large number of tasks. First, you verify whether the performance of the two systems differ in a statistical significant way both across tasks and when performance for all tasks is aggregated (using the t-test). Then, you verify whether the two systems differ in a statistical significant way in the number of tasks in which one system is better than the other (using the Sign test [19]).

The mean time to complete all tasks was only slightly better for VC+V than VC+H for the training phase (19%). Although still non-significant, there was a more prominent difference seen in the testing phase (30%), in which subjects performed better with VC+V. When we broke down the evaluation by task, there were similarly no significant differences found.

Finally, in the second step of our analysis we determined, for each task, what interface the subjects performed better. VC+V performed better on all five inspection tasks and also performed better on three out of the four sensitivity analysis tasks. Then we applied a two-tailed Sign Test [19] to the obtained data (VC+V better 8 out of 9). This test measures the likelihood that the subjects performed better on one version over the other on m or more out of n independent measures under the null hypothesis that the two versions are equal. This test is insensitive to the magnitude of differences in each measure, noticing only which condition represents a better result. The outcome of this test is that overall subjects performed significantly better with VC+V in the testing phase ($p = 0.039$).

According to our analysis we can conclude that even though there are no significant differences in training time between the two interfaces, subjects work more efficiently with the vertical interface after the initial training.

In addition to these overall results, we looked closely at individual task results and interaction logs and found some interesting observations. For example, we were able to understand why there were no significant differences in time to complete value function sensitivity analysis tasks. In the VC+H subjects took extra time because they had to recall what the value function was and how to access the hidden display, whereas subjects did not experience this problem in VC+V. They did, however, take longer to interact with the smaller display, and some subjects ended up opening the on-demand view. Although the persistent display did not affect time to perform the task, we will see that it played a bigger part in overall decision-making (See Part B below). We also found problems in our design regardless of orientation. For example, we found that subjects had trouble with the pump function for sensitivity analysis of weights (see [5]) in both VC+H and VC+V. These and several observations by task will be taken into consideration with future design iterations of ValueCharts.

5. PART B: EXPLORATORY STUDY

In Part A of our evaluation, we looked closely at how subjects performed tasks that are important for effective analysis in decision-making. Because these tasks still need to be appropriately combined to lead to effective preferential choice, in Part B we attempt to more fully understand the DM's experience in using VC+ to perform preferential choice. To achieve this goal we observed subjects using the tool in a real decision-making context of their choice (out of three possible ones). After interacting with the system, subjects filled out a questionnaire regarding their experience with VC+ in the decision-making process.

5.1 Insight Characteristics

A primary purpose of visualization is to generate insight [7]. It has been argued that the generation of insights leads to a better understanding of the domain and problem situation, thus favoring better decisions. An effective visualization will aid the DM to see things that would otherwise go unnoticed, as well as enable her to view information about her preferences in a new light.

In our exploratory study we measure the amount of insight each subject gains from using VC+ for a particular decision-making scenario. We use the definition of insight provided in [17]:*an individual observation about the data by the participant, a unit of discovery.* In terms of our model, we consider the DM's preferences and weighting as part of the data observed.

Saraiya and North in [17] propose an evaluation protocol for insights based on a set of "characteristics of an insight". Although this set is assumed to work for other domains, it is accepted that it may require some adaptation.

Notice that Saraiya and North's study for evaluating insight is in a very specific and technical domain (i.e., microbiological and microarray data). And the subjects had extensive domain knowledge. In contrast, our study is less specific (subjects worked in different domains they could choose from), much less technical (e.g., house rental) and the subjects were not experts.

Based on these observations, in our study we applied some slight modifications and generalizations to Saraiya and North's original set "characteristics of an insight".

The following is our characterization of insight as applied to preferential choice:

- **Fact:** The actual finding about the data (e.g. "Samsung [cell phones] are the smallest")
- **Value:** How to measure each insight? We determined and coded the value of each insight from 1 - 3, whereas simple observations of domain value and top ranking (e.g. "cheapest place is in East Van") are fairly trivial, and more global observations regarding relationships and comparison (e.g. "more expensive phones have all the features") are more valuable.
- **Category:** Insights were grouped into several categories:
  - Simple fact: an alternative rank or identification of domain value e.g. "This phone is fairly light", "This phone is only [ranked] fourth for battery"
  - Sensitivity: how a change affects the results e.g. "This house again!", "Now this phone is third"
  - Realization of personal preferences: users often stated that they made a realization about their preferences e.g. "it makes sense, because I really like hiking and nature", "brand should be more important [to me]"

These categories were defined after the experiment, and the grouping closely lends to the value coding.

5.2 Domain Data Sets

In order to ensure that the users had the capability to determine insightful facts about the information presented to them, it was important that they had a genuine interest in the domain that was studied. The subjects were asked to choose one out of three different decision problems. Each of the decisions included a scenario in the following domains:

House Rental: data was taken loosely from current postings on AMS Rentsline, where any missing information was fabricated. General information, such as rent, location, type, etc, were consistently available, but other more detailed information (bedroom size by sq-ft) was often fabricated. The scenario is that the DM goes to school at UBC and would like to move off campus. It is assumed that the DM is only considering Point Grey, Kitsilano, Downtown, and EastEnd. The House Rental decision problem contained 13 objectives and 10 alternatives.

Cell Phone: data was taken from Rogers Video website, and there were only a few cases of missing information. The information was narrowed down to 17 primitive objectives, and anything the participant was looking for (i.e. text-messaging) was assumed to be a feature included in all phones. The scenario is that the DM is looking for a phone from Rogers Wireless based on a 3-year plan (as prices were quoted). The Cell Phone decision problem contained 17 objectives and 12 alternatives.

Tourism: in this situation, the data was taken from Tourism Vancouver Official Visitor's Guide. The alternatives were narrowed down to those listed as being Downtown, East End, West End, and North Van. The scenario is that the DM is looking to take a visiting friend to a local tourist attraction. Alternatives were further categorized as type (scenic, historic, etc.), and indoor/outdoor. Cost was assumed as average/adult. The Tourism decision problem contained 17 objectives and 12 alternatives.

We realize that one possible problem in this methodology is that the number of alternatives and objectives differ (which may affect the number of insights). We do, however, believe that the advantages dominate this possible disadvantage.

Each scenario was explained to the participants, and they were asked to choose which one they would like to work with.

5.3 Procedure

At the onset of Part B of the study, subjects have already undergone considerable training and practice from Part A. However, since the construction interface was not tested in the controlled study, experimenters worked with the subjects to build the initial decision model. Objectives were presented to them in a pre-existing hierarchy with all available factors, and were told to remove and rearrange as they pleased (additions were not allowed since data set was fixed and could not be extended).

To set their initial preference model, they were instructed to go through the list of objectives and set the value function of each one to reflect their true preferences. Default functions were provided, where typically linear continuous functions were given (i.e. positive for battery talk time, negative for price), and each discrete objective was set with a best, worst, and 0.5 for others. Finally, the subjects ranked the objectives with the SMARTER weighting technique [12]. Their resulting decision model was then presented with VC+. The subject was asked to use the interface to analyze the decision model, perform any sensitivity analysis changes as they see fit, and view any information that they required. They were instructed to work with the interface to make a decision about the data, where the decision could be to select one or more preferred alternatives.

Subjects were asked to "think aloud" as they analyzed the preference model, being sure to let the experimenter know anything interesting that they saw. Notes were taken by the experimenter, and interaction logging was turned on once VC+ was created.

The subject was asked to take as little or as much time as she needed in order to reach his decision. If she was finished quickly, the experimenter would probe, but end the session if she was satisfied with the decision. The time for the experiment (total of both Part A and B) was 60 minutes, and if subjects were approaching the 60 minute mark, they were warned by the experimenter but welcomed to stay until as long as the 75 minute mark.

At the end of the exercise the subject was asked what their decision was, and to keep that in mind when answering the post-experiment questionnaire.

5.4 Results

It appeared that every subject had a genuine interest in the domain that they chose (10 cell phone, 6 tourism, and 4 house). Overall, subjects were able to use the tool and conclude on a best decision.

Subjects went through the construction phase carefully. The time spent inspecting the interface (minus construction) ranged from 3-16 minutes. The number of insights ranged from 0 to 10.

5.4.1 Comparison between the two interfaces in terms of insights

Table 2 summarizes two measures of insight gained and usage time, illustrating the two different interfaces. It shows a) mean number of insights acquired, b) the mean sum of value for all insight occurrences, and c) the average total time each subject spent using the tool until they felt that they reached a decision.

VC+V	mean	sd	min	max
Count of insights	4.7	5.0	0	10
Total insight value	8.8	2.9	0	15
Total time	9.93	2.8	3.14	16.45
VC+H	mean	sd	min	max
Count of insights	3.3	6.2	0	10
Total insight value	5.9	3.2	0	15
Total time	8.69	5.3	3.03	12.65

Table 2: Insight Results

Statistical analysis indicates that there are no significant differences, despite the fact that there appeared to be a great difference in the mean insights and value (49% and 34% more, respectively). Because of these noteworthy differences we also measured effect sizes (the magnitude of the differences) to determine the practical significance of the differences. Cohen's d [10] provides a standardized measure of

the mean difference between two treatments. In this measure $d > 0.8$ is considered to be a large effect, $0.8 > d > 0.5$ to be a medium effect, and $d < 0.5$ to be a small effect. We found the effect sizes of the insight count and insight value to be 0.40 and 0.51 respectively. So, although our results are not statistically significant, according to Cohen's criteria, using VC+V has a medium effect on the value of insights reported by our participants.

Since the evaluation method is more qualitative and subjective than quantitative, general comparison of the tendencies in the results is also appropriate. There were more insights counted for the vertical interface, which also fared better when value factor was considered. Looking more closely at the interaction logs reveal that subjects tended to perform more sensitivity analysis on VC+V, which in turn led to more insights on sensitivity. There were 89% more sensitivity analysis of value function performed on the vertical interface than the horizontal. We conclude that the reason for this is that the persistent view a) acts as a reminder of what the value function is and that it can be changed and b) is more inviting for users to directly manipulate value function. We hypothesize that there is a benefit from the persistent view of the component value functions, but may revisit the persistent sensitivity analysis technique in future iterations.

More time was spent on the vertical interface. In contrast to time measurement in Part A that we used to gauge performance of lower-level tasks, more time spent performing the overall task of making a decision can not be viewed as negative. In fact, the general trend was that the more time spent by the subject on the decision problem, more insights were reported.

It should be noted that, regardless of the interface, the results were very mixed. Some subjects did not have any insights, and some had many. The standard deviation was high overall (see Table 2). Individual differences were more apparent in this part (versus Part A) because subjects' personalities could affect the amount of insights reported (a challenge of the think-aloud technique [13, 15]). In addition, the possible varying level of interest in each subject's selected domains may contribute to this variance. Nonetheless, we believe that providing the subject with a selection of domains helped with degree of interest. A more extensive study might specify a single domain and recruit participants with a specific requirement (e.g. recruit participants who are in the market for a new cell phone, and plan to purchase or upgrade in the next month).

5.5 Post-study questionnaire

Following the exploratory study, we completed the session by asking the subject a number of open questions and having them fill out a post-experiment questionnaire. They were asked to answer each question by selecting the degree of agreement of the statement from 1 to 5 where 1 is strongly disagree and 5 is strongly agree. Some questions were specific to the exercise they performed in Part B, while others were about the overall experience of using VC+. This questionnaire provides information not only on differences between VC+V and VC+H, but also on the subjects' experience and satisfaction with VC+ in general.

All subjects were generally satisfied with the decision that they made ($\mu = 4.25, \sigma = 0.55$), although their level of confidence was slightly lower overall ($\mu = 3.95, \sigma = 0.76$). A closer look shows that 4 of 6 subjects who gave this a 3 or "neutral" rating had 3 or less insights. Subjects felt that VC+ was a good tool for learning about their preferences in the selected domain ($\mu = 3.95, \sigma = 0.69$). This was tied closely to insights as well, as we found a significant positive correlation between the rating of this question and insight. This analysis further supports the assumptions made in Part B that more (insights, time, interaction) is better.

Our subjects, who did not have any previous exposure to decision analysis methods, felt that they learned much about how to analyze their decision model ($\mu = 4.20, \sigma = 0.62$). We attribute this much to the construction interface that they were exposed to in training for Part A and working with building their decision model in Part B, since it represents tasks that support the higher-level analytical task of learning.

Overall VC+ was very well-received. All subjects thought that VC+ is useful, intuitive, easy to use and quick to learn. In particular, subjects rated the usefulness very high ($\mu = 4.40, \sigma = 0.50$), and strongly agreed that visualizing their preferences helps in their understanding of the decision ($\mu = 4.45, \sigma = 0.51$).

Details on the answers to the most informative questions in the post-study questionnaire are shown in Table 3.

	strongly disagree (1)	disagree (2)	neutral (3)	agree (4)	strongly agree (5)
I am satisfied with the decision I made					
VC+V	0	0	1	5	4
VC+H	0	0	0	8	2
I am confident about the decision I made					
VC+V	0	0	4	2	4
VC+H	0	0	2	7	1
I learned a great deal about my preferences in [selected domain]					
VC+V	0	1	1	6	2
VC+H	0	0	1	8	1
This is a useful tool for making decisions					
VC+V	0	0	0	5	5
VC+H	0	0	0	7	3
Visualizing my decision model helps me understand it more clearly					
VC+V	0	0	0	7	3
VC+H	0	0	0	4	6
I learned a great deal about how to analyze my decision model					
VC+V	0	0	0	6	4
VC+H	0	0	2	6	2

Table 3: Results of post-study questionnaire

6. CONCLUSIONS AND FUTURE WORK

We addressed some challenges of information visualization evaluation [16] in several manners. First and foremost, we developed and applied a taxonomy of tasks that represents a benchmark framework for design and evaluation of visualization techniques for preferential choice. In addition to a controlled experiment, we used a triangulation of methods that includes an exploratory study in which we matched users with real data in realistic scenarios and included a measure of insight.

We looked at ValueCharts in several angles with this evaluation focusing on comparing two versions with different orientations. First, we assessed how the subjects performed

on the low-level tasks. On average the subjects performed well in correctness, varying to some degree in length of time spent to complete the tasks. In turn, when asked to perform the high-level task of making a decision with our tool, the subjects reported that they were quite satisfied with their decision. These results corroborate our claim that if an interface supports the lower level tasks of PVIT well, then the interface also will enable the higher level tasks of the model. We pruned our task model to focus more on the basic tasks of inspection and sensitivity analysis, which more directly support the higher level task of decision-making. Since subjects were generally satisfied with the construction phase as well, it added to the success of ValueCharts as tools for preferential choice.

Some of the evidence that we have collected suggest that the vertical and horizontal ValueCharts designs are not equivalent interfaces since a) the Sign test indicates that subjects perform better on the VC+V than VC+H on low level tasks, and b) VC+V has a medium effect on insight value as we explored how subjects performed the higher level task of decision making. However, the lack of statistical significance for the difference in insights (count and value) indicates the need for a larger experiment.

Nonetheless, our overall evaluation of ValueCharts Plus is very promising. Subjects rated our tool very high in usefulness, learning, and understanding.

In future iterations of the ValueCharts design, we would like to address some of the issues and observations that we discovered in these studies. We also plan to conduct a more extensive experiment using a larger pool of subjects and focusing on a single domain with participants screened for specific requirements (i.e. who are in the market of that particular domain). We will also consider some changes in our experimental procedure such as using other HCI experts to conduct the analytical evaluation. Additionally, we intend to conduct further studies of the construction interface.

7. REFERENCES

[1] R. Amar and J. Stasko. A Knowledge Task-based Framework for Design and Evaluation of Information Visualizations. In *Proceedings of InfoVis '04*, pages 143–150, Austin, TX, USA, 2004. IEEE Computer Society. Best Paper.

[2] N. V. Andrienko and G. L. Andrienko. Informed Spatial Decisions through Coordinated Views. *Information Visualization*, 2:270–285, 2003.

[3] T. Asahi, D. Turo, and B. Shneiderman. Visual Decision-Making: Using Treemaps for the Analytic Hierarchy Process. In *Proceedings of CHI '95*, pages 405–406, New York, NY, USA, 1995. ACM Press.

[4] R. Bade, F. Ritter, and B. Preim. Usability comparison of mouse-based interaction techniques for predictable 3d rotation. In *Smart Graphics*, pages 138–150, 2005.

[5] J. Bautista and G. Carenini. An integrated task-based framework for the design and evaluation of visualizations to support preferential choice. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 217–224, New York, NY, USA, 2006. ACM.

[6] V. Belton. VISA: Visual Interactive Sensitivity Analysis. SIMUL8 Corporation, Boston, MA, 2008.

[7] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., 1999.

[8] G. Carenini and J. Lloyd. ValueCharts: Analyzing Linear Models Expressing Preferences and Evaluations. In *Proceedings of AVI '04*, pages 150–157, Gallipoli, Italy, 2004. ACM Press.

[9] R. T. Clemen. *Making Hard Decisions*. Duxbury Press, Belmont, CA, USA, 2nd edition, 1996.

[10] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences (2 ed.)*. Lawrence Earlbaum associates.

[11] B. DiEugenio, M. Glass, and M. J. Trolio. The DIAG experiments: Natural language generation for intelligent tutoring systems. In *The Second International Natural Language Generation Conference*.

[12] W. Edwards and F. H. Barron. SMARTS and SMARTER: Improved Simple Methods for Multiattribute Utility Measurement. *Organizational Behavior and Human Decision Processes*, 60(4):306–325, 1996.

[13] K. A. Ericsson and H. A. Simon. *Protocol Analysis: Verbal Reports as Data*. MIT Press, Cambridge, MA, 1984.

[14] K. Hinckley, R. Pausch, D. Proffitt, J. Patten, and N. Kassell. Cooperative bimanual action. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 27–34, New York, NY, USA, 1997. ACM Press.

[15] J. Nielsen, T. Clemmensen, and C. Yssing. Getting access to what goes on in people's heads?: Reflections on the think-aloud technique. In *NordiCHI '02: Proceedings of the second Nordic conference on Human-computer interaction*, pages 101–110, New York, NY, USA, 2002. ACM Press.

[16] C. Plaisant. The challenge of information visualization evaluation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 109–116, New York, NY„ 2004. ACM Press.

[17] P. Saraiya, C. North, and K. Duca. An evaluation of microarray visualization tools for biological insight. In *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 1–8, Washington, DC, USA, 2004. IEEE Computer Society.

[18] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of VL '96*, page 336, Washington, DC, USA, 1996. IEEE Computer Society.

[19] S. Siegel and N. J. J. Castellan. *Nonparametric statistics for the behavioral sciences*. McGraw Hill, 1988.

Interaction Environment Design

Model-based Layout Generation

Sebastian Feuerstack, Marco Blumendorf, Veit Schwartze, Sahin Albayrak
DAI-Labor, TU-Berlin
Ernst-Reuter-Platz 7, D-10587 Berlin
{Sebastian.Feuerstack, Marco.Blumendorf, Veit.Schwartze, Sahin.Albayrak}@DAI-Labor.de

ABSTRACT
Offering user interfaces for interactive applications that are flexible enough to be adapted to various context-of-use scenarios such as supporting different display sizes or addressing various input styles requires an adaptive layout. We describe an approach for layout derivation that is embedded in a model-based user interface generation process. By an interactive and tool-supported process we can efficiently create a layout model that is composed of interpretations of the other design models and is consistent to the application design. By shifting the decision about which interpretations are relevant to support a specific context-of-use scenario from design-time to run-time, we can flexibly adapt the layout to consider new device capabilities, user demands and user interface distributions. We present our run-time environment that is able to evaluate the relevant model layout information to constraints as they are required and to reassemble the user interface parts regarding the updated containment, order, orientation and sizes information of the layout-model. Finally we present results of an evaluation we performed to test the design and run-time efficiency of our model-based layouting approach.

Categories and Subject Descriptors
H.5 [**Information Interfaces and Presentation**]: User interfaces; D.2.2 [**Software Engineering**]: Design Tools and Techniques- *User Interfaces*; H.1.2 [**Models and Principles**]: User/Machine Systems-*Human factors;* H.5.2 [**Information Interfaces and Presentation**]: User Interfaces-*graphical user interfaces, interaction styles, input devices and strategies, voice I/O*.

General Terms
Design, Human Factors

Keywords
Layouting, model-based user interfaces, constraint generation, context-of-use, human-computer interaction.

1. INTRODUCTION
Interactive applications that are deployed to smart environments must be able to support different context-of-use scenarios. Such scenarios include e.g. adapting the user interface seamlessly to various interaction devices or distributing the user interface to a set of devices that the user feels comfortable with in a specific situation. Such adaptations require flexible and robust (re-) layouting mechanisms of the user interface and need to consider the underlying tasks and concepts of the application to generate a consistent layout presentation for all states and distributions of the user interface. The broad range of possible user interface distributions and the diversity of available interaction devices make a complete specification of each potential context-of-use scenario during the application design impossible.

Specifying the interdependencies between the user interface components using constraints is a common approach to address these issues and nowadays constraint solvers can calculate hundreds of constraints in a reasonable amount of time. To our knowledge there is still an approach missing that supports designers of a user interface in generating these constraints based on the design specifications. A manual constraint setup has two disadvantages: first the pure amount of constraints that is required even to address small interactive systems is hard to handle, and second, the fault tolerance of the constraint setup is complex to attain. Even one single constraint that is not properly specified can destroy the complete layout in a specific situation that has not been considered by the designer during the development process.

This paper introduces an approach for a model-based user interface layouting that differs from previous approaches in two general aspects:

1. We interpret the information from already existing user interface design models, such as the task tree, the dialog model, the abstract user interface model (AUI), the concrete user interface model (CUI), the domain model and the context model for deriving the user interface layout. Therefore we propose an interactive, tool-supported process that reduces the amount of information that needs to be specified for the layout. The tool enables designers to comfortably define design model interpretations by specifying statements and subsequently applying them to all screens of the user interface.
2. We shift the decision about which of the statements are applied from design-time to run-time to enable flexible context-of-use adaptations of the user interface layout. This allows us to describe new context-of-use adaptations of the layout without the need to change the application itself just by describing the layout characteristics of a new platform or a new user profile.

The next section discusses the related work that has been considered to support our approach. Section 3 presents the layout model and its relation to the other user interface design models. Section 4 describes our approach to generate constraints based on the interpretation of design models by an interactive, tool-supported process. Section 5 presents our implementation, integrating a layout model agent into our run-time-environment (MASP) [2,5]. Section 6 discusses results of an evaluation we did to test the efficiency of the model-based layout process at design-time by measuring the performance of the constraint generation and constraint solving at run-time.

AVI'08, 28-30 May , 2008, Napoli, Italy

Finally Section 7 summarizes the paper and outlines future work.

2. RELATED WORK

Nichols et al. lists in PUC [10] a set of requirements that need to be addressed in order to generate high-quality user interfaces. As for layout information they propose to not include specific layout information into the models as this first tempts the designers to include too many details into the specification for each considered platform, second delimits the user interface consistency and third might lower the chance of compatibility to future platforms. Different to PUC we are not focusing on control user interfaces, but end up in a domain independent layout model that specifies the containment, the size, the orientation and the order relationships of all individual user interface elements. Therefore we do not want to specify the layout manually for each targeted platform and do not rely on a set of standard elements (like a set of widgets for instance) that has been predefined for each platform.

The SUPPLE system [7] treats interface adaptation as an optimization problem. Therefore SUPPLE focuses on minimizing the user's effort when controlling the interface by relying on user traces to estimate the effort and to position widgets on the interface. Although in SUPPLE an efficient algorithm to adapt the user interface is presented, it remains questionable if reliable user traces can be generated or estimated. While SUPPLE also uses constraints to describe device and interactor capabilities they present no details about the expressiveness of the constraints and the designers effort in specifying these constraints.

The layout of user interfaces can be described as a linear problem, which can be solved using a constraint solver. Recent research has been done by Vermeulen [16] implementing the Cassowary algorithm [1], a weak constraint satisfaction algorithm to support user interface adaptation at run-time to different devices. While he demonstrates that constraint satisfaction can be done at run-time, to our knowledge he did not focus on automatic constraint generation. Other approaches describe the user interface layout as a space usage optimization problem [8], and use geometric constraint solvers, which try to minimize the unused space. Compared to linear constraint solving, geometric constraint solvers require plenty of iterations to solve such a space optimization problem. Beneath performance issues an efficient area usage optimization requires a flexible orientation of the user interface elements, which critically affects the user interface consistency.

Richter [13] has proposed several criteria that need to be maintained when re-layouting a user interface. Machine learning mechanisms can be used to further optimize the layout by eliciting the user's preferences [9]. The Interface Designer and Evaluator (AIDE) [14] and Gadget [6] are incorporating metrics in the user interface design process to evaluate a user interface design.

Both projects focus on criticizing already existing user interface layouts by advising and interactively supporting the designer during the layout optimization process. They follow a descriptive approach by re-evaluating already existing systems with the help of metrics. This is different to our approach that can be directly embedded into a model-based design process (forward engineering).

In the next chapter we present our approach for a layout model, that is designed to be part of a model-based user interface design approach [15] like proposed by the Cameleon Reference Framework [3]. Following a model-based user interface development involves a developer specifying several models using a model editor (such as a task model, a domain model, and a dialog model). Each abstract model is reificated to more concrete models until the final user interface has been derived.

3. THE LAYOUTING MODEL

Like illustrated by figure 1 our layouting model is part of such a model-based user interface design process. To derive a layout model the designer has to specify interpretations of the design models by defining layout statements. In general two different statements are possible: First, layout statements that are explicitly specified for one user interface and second, layout statements that are defined independent of the user interface. The latter interprets pre-defined context information to address layout adaptations for specific devices and users or specific environments. Currently we are focusing on interpreting the context, task tree, AUI and dialog models to derive layout information.

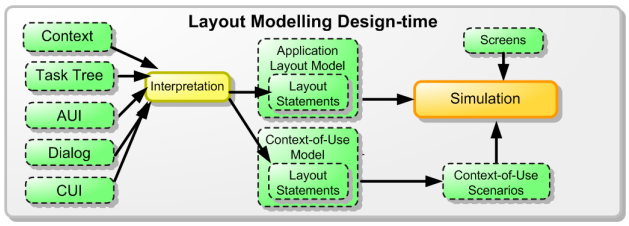


Figure 1: The layouting process is embedded into a model-based user interface design process.

For each new layout statement that is written into the layout model, the designer can initiate a simulation to preview the result. The simulation positions the individual user interface elements based on the specified layout model statements for all screens and context-of-use scenarios that are known at design-time.

Our layout model basically consists of a list of ordered statements. Like illustrated by figure 2, each statement is composed of six properties: the characteristic of the resulting layout primary addressed (containment, orientation, and size) (3.1), the design models used for the constraint generation (3.2), the context-of-use information (3.3), the addressed scope (3.4), the type of condition (3.5) and finally the priority value (3.6). In the following sections we describe these constituent parts of a layout statement in greater detail.

3.1 Layout Characteristics

We identified four of these characteristics that can be used to specify the layout of a graphical user interface: The *containment*, the *order*, the *orientation* and the *size* of the user interface elements.

Like illustrated by figure 3, the *containment* describes the relation between two basic types of entities: Containers (like *c1*) consist of a set of nested containers (*c2*+*c3*) and nested elements (*c2* contains *e1* and *e2*). Elements can present information to the user or enable the user to enter data to the application and cannot be decomposed any further. Additionally a layout describes an *order* of elements (e.g. from left to right and from top to bottom: *e1* before *e2* and *c2* before *c3)*. The *orientation* distinguishes between elements that are oriented horizontally or vertically to each other (e.g. *e1* vertical to *e2*). Finally the *size* specifies the width and height of containers and elements (e.g. the width of *e3* is ½ of the width of *e4*).

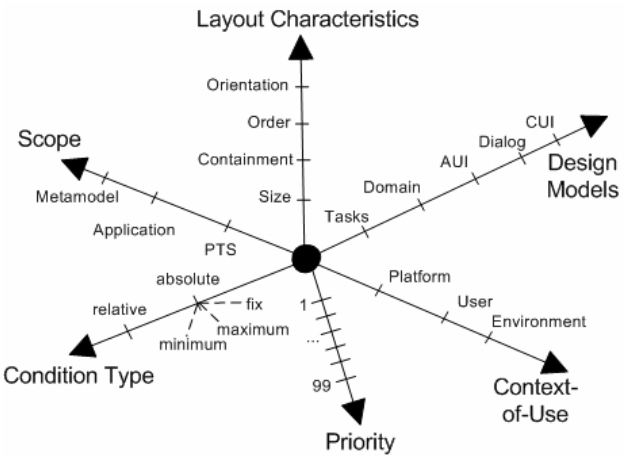


Figure 2: The six axes of the space of properties of the layout statements.

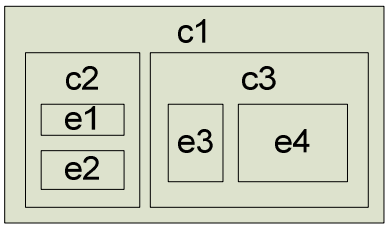


Figure 3: Exemplary sketch of a user interface layout.

3.2 Design Models Interpretation

Design Models are used to specify the interactive system on different levels of abstraction. We interpret the information of these models to derive the user interface layout. A task model, a domain model, an abstract user interface, and the dialog model are typically part of a model-based user interface design. Beneath the task model's hierarchical structure that can be used to derive a basic containment structure for the layout [5] other information can be derived: For instance, the sum of all atomic tasks related to the task tree depth or related to its width can be used to balance the presentation size of the tasks. For instance the CTT notation [12] categorizes interaction tasks into "edit", "control" and "selection" tasks. This task information can be addressed differently related to the context-of-use, for instance by prioritizing those tasks that require user input.

Looking at the abstract user interface model, the interaction object type can be used by a layouting statement to derive an orientation. E.g. navigational elements can be set vertically or horizontally depending on the menu level, whereas selection elements can be oriented vertically for a large amount of elements and horizontally for small amounts by a layouting statement.

3.3 Condition Type

Each statement describes either an absolute condition (minimum, maximum, or fixed) or a relative condition that relates two or more elements. A relative condition targeted to the orientation characteristic is for instance: "e1 over e2", regarding the size a relative condition can specify e.g. "e4 double the width of e3" and finally regarding the containment it has the form of "c3 contains e4". A maximum statement containing an absolute condition can be used to specify a column layout where elements are wrapped to the next row after a specified amount of elements is exceeded. Further on, a maximum statement can restrict the size (regarding its height or width) or the number of elements in a container. If the size limit is exceeded new containers are generated.

3.4 Application Scope

Each statement has a fixed scope to address every application (application independent statements), the whole application, a set of reoccurring elements or a specific screen to handle very fine grained design requirements. Application independent statements are used to characterize context-of-use adaptations that are required to be considered when layouting for a specific device (such as specifying the screen size limitation, or the minimum size of control buttons for a touch screen). Application wide statements help the designer to generalize design decisions and maintain consistency as layouting decisions can be modeled just once and are automatically applied for each reoccurring situation. The more global than local statements have been defined the better is the robustness for context-of-uses changes and the better layout consistency can be expected. Finally a statement can be limited to address a single screen to fine tune the layout for aesthetical reasons or to refine an application wide layout statement.

3.5 Context–of-Use Scope

The context-of-use describes the user, who has preferences and demands for the actual situation, a set of devices that she likes to use in a certain environment. A layout statement can be specified to be relevant for a specific context-of-use situation only. For instance in an environment that supports location tracking, the distance of the user to a device can be used to scale the control elements of the user interface. In the former case the control elements are sized small if the user has no way to control because of his distance to the display, whereas in the latter case the control tasks are sized to meet a pen or a finger print respectively.

3.6 Strict Order by Priority

Specifying the priority of a statement is required on the one hand to support a general-to-specific layouting approach and on the other hand to prevent the generation of conflicting layout constraints. Thus, general layouting principles, as described in style-guidelines or given by a corporate design can be generally defined and overwritten to address more specific situations later on. We address these aspects by specifying a strict order in that the layout statements are evaluated to generate the constraints that we indicate by the priority property.

3.7 Conclusions

Deriving an interface layout based on the design models of a model-based interface development approach results in a consistent layout. Further on, such a layout model derivation reduces the information that has to be specified for the interface layout as a lot of information is already available in the design models. The more global application layout statements can be derived from the design models the better robustness of the interface to unknown context-of-use changes can be expected.

To realize such a model-based layout generation that is based on model interpretation we require (1) an efficient way for the designer to select suitable model interpretations for generating a layout (2) a process that eases the identification of global interpretations to enforce the layout's consistency and robustness against context-of-use changes and finally (3) a model-based run-time system that can evaluate these interpretations in an efficient manner so that layouting adaptation of a user interface is possible at run-time.

We introduce our model-based layout editor in the next section that we implemented to address requirements (1) and (2), and describe how we realized the layouting in our run-time environment, the Multi-Access Service Platform (MASP), to adapt to context changes (3).

4. LAYOUT MODEL GENERATOR

Using the layout model generator the designer has to initially load all design models of an interactive application as well as already known contexts-of-use scenarios that contain device capability descriptions and the preferences of the user.

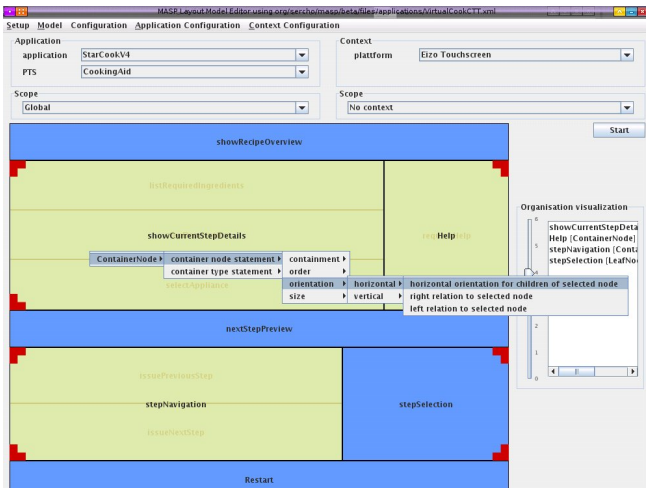


Figure 4: The MASP Layout Model Generator

Figure 4 shows a screenshot of the editor: Using the pull down menu in the upper left corner ("PTS"), the designer can browse through all screens of the application. Each screen consists of a set of elements that should be presented simultaneously to the user on a single device. Predefined interface distributions that allow to one screen to several devices, each containing complementary parts of the interface can be defined as a set of separate screens with the same context-of-use

The result of the layouting process, the layout model is visualized by a box-based layout that represents each individual user interface element that is part of a screen as a box. By the box-based layout the designer gets an impression of the layouting results concerning the individual elements size, containment, order and orientation relationships. Different to the layout result that is calculated during run-time and ends up with absolute coordinates for each box, the simulator linearly scales the preview s but considers the aspect ratio of the targeted device in order to comfortably support layout modeling for large display.

Using the layout editor, the designer specifies all layout statements by using a context menu that is related to the box-based simulation area. The application scope (global, application or screen specific), and the context-of-use of a statement can be set by two separate pull down menus above the simulation area.

The process of deriving layout statement is supported by the tool following several subsequent steps:

1. The designer decides about the layout characterization that the statement should address: the containment structure, the element order, the orientation or the size.
2. The designer defined a new layout statement that interprets one or more
    a. design model information (such as the AUI type: input, output, control, or selection task or the CUI type)
    b. context model information that require a layout adaptation.
3. The designer can visually weight a relational statement. E.g. relate the size-ratio between input and output elements in general or specify size relations between two specific boxes.
4. The Model Generator automatically applies the new statement consistent to the design models to all screens of the applications (limited by the scope of the statement).
5. The Model Generator updates the boxed simulation area to reflect the new layout for all screens and all actually supported context-of-use scenarios of the user interface layout.
6. The designer checks the result and manipulates the order of the statements.

In order to ease the identification of global layout statements to force the layout consistency and robustness against context-of-use changes (requirement 2), we implemented an abstract-to-detail slider, which is depicted to the right in figure 4. The slider allows the designer to browse through the nested boxes by moving the slider up and down starting from the box that contains the whole application, to the atomic elements that describe individual user interface widgets. Following such an abstract-to-detail layout modeling, the designer is supported to start specifying statements on the highest abstraction level possible. The editor visualizes atomic elements in blue and boxes that contain nested elements through a yellow overlay like depicted in figure 4.

To prevent specifying conflicting statements the designer is allowed only to define relational statements between elements that have been specified on the same nesting level (which corresponds to the abstraction level of the task tree if the task model has been used to derive the containment). In the editor, we use the red corners to indicate elements that are located on the same nesting level and thus can be target of a relational statement. For instance in figure 4 the red corners indicate two separate boxes of an exemplary application that are not directly related: The upper one highlights the two boxes "showCurrentStepDetails" and "Help" whereas the lower one consists of one box "stepNavigation" and one individual element "stepSelection". In this case the designer has the option to define an interpretation for the relation between "showCurrentStepDetails" and "Help" but not the option to specify a direct relation containing elements of the upper and the lower box (since such a relation has to be set on a higher level of abstraction which contains both boxes).

Each statement that has been defined is written into the layout model and gets instantly evaluated to a set of constraints that is solved to update the box-based preview. This process happens without any remarkable delay so that we can recalculate the constraints on the fly to give an instant visual feedback.

Figure 5 presents a screenshot of the editor's view of the layout model. The layout statements are grouped by the layout characteristic they are primarily addressing. In case conflicting constraint sets have been generated the last statement that the designer has entered and the one that caused the conflict is highlighted red.

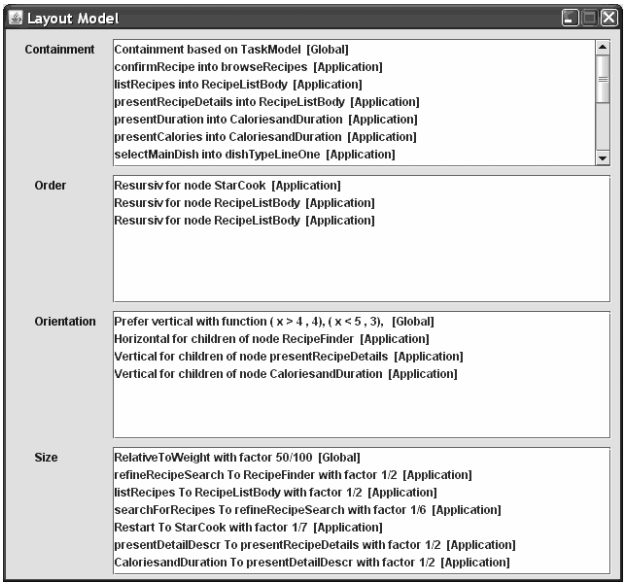


Figure 5: The actual layout model consisting of a set of statements that are grouped by the layout characteristic they are targeting to.

Not all of the four layout characteristics can be handled independently from each other. First, the containment constraints the order, orientation and size characteristics and second, the element order constraints the orientation and the element size. To manage these interdependencies we define a general order in which the statements are processed based on the layout characteristic they are mainly addressing: Like depicted by the screenshot in figure 5 we process the containment-related statements before the order-related ones Thereafter the orientation-related statements and finally the size -related statements are processed.

After a suitable set of constraint generating functions has been identified, the designer can check the resulting layout for its adaptivity to manage certain context-of-use scenarios by browsing through a set of predefined contexts-of-use. Predefined contexts-of-use contain further context-specific layout statements that have been specified independent from a certain application and are reflecting the capabilities of a device or the preferences of a user that are already known at design-time. Like illustrated in figure 1 the layout statements of predefined contexts-of-use are merged to the layout statements of the application to simulate the user interface layout. In the following section we describe how the layout statements are evaluated to constraints in our run-time environment.

5. CONSTRAINT GENERATION AT RUN-TIME

Following the idea of using software agents to coordinate the user interface management system [3] we are using an agent-based run-time environment, the Multi-Access Service Platform (MASP) to generate and adapt user interfaces. But instead of requiring a hierarchical organization to several agents like proposed by PAC-Amodeus [11], the communication flow between the agents in our environment can be flexibly configured based on the requirements of the interactive application. As illustrated by figure 6 the environment is driven by several agents where each interprets one user interface model. In contrast to other approaches [3,15] that refine a user interface model at design-time to end up with a compiled version of the user interface, we keep all of the models alive at run-time. This allows us to more flexibly react to context-of-use changes that have not been desired at design-time by specifying the required adaptation on an abstract model-level.

Each agent is comprised of two parts: a tuple space to store the instantiated model information and a manager containing the semantics and functionality to manipulate the model information. Whereas the manager has complete access to its own tuple space it is not aware of the other agents connected to the system. We connect the agents by using tuple space operations (atomic read/manipulate/write) and the eventing system of a tuple space. The eventing system allows a manager to register for changes of another tuple space. Each agent, handling one user interface model is instantiated once to run a single application, but is able to handle several sessions for different users that are accessing the same application.

The communication processes between all agents are not hard wired but instead configured for each application based on the user interface models that are relevant for the applications domain. Therefore we can easily add the layouting model agent as an additional component to the MASP..

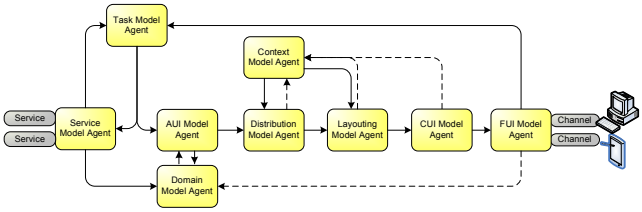


Figure 6: The layouting model is embedded as an agent into our run-time environment.

As illustrated by figure 6, the layouting agent registers itself for events from the distribution agent, which calculates the distribution of a presentation task set to all platforms that are connected to the MASP. For each new or updated user interface distribution the layouting agent receives an event containing all the elements of the user interface that should be simultaneously presented on a specific platform. While the distribution agent is required to calculate a reasonable user interface distribution based on the actual context of use, the layout agent has to layout a presentation for all the individual elements it receives from the distribution agent for a single device.

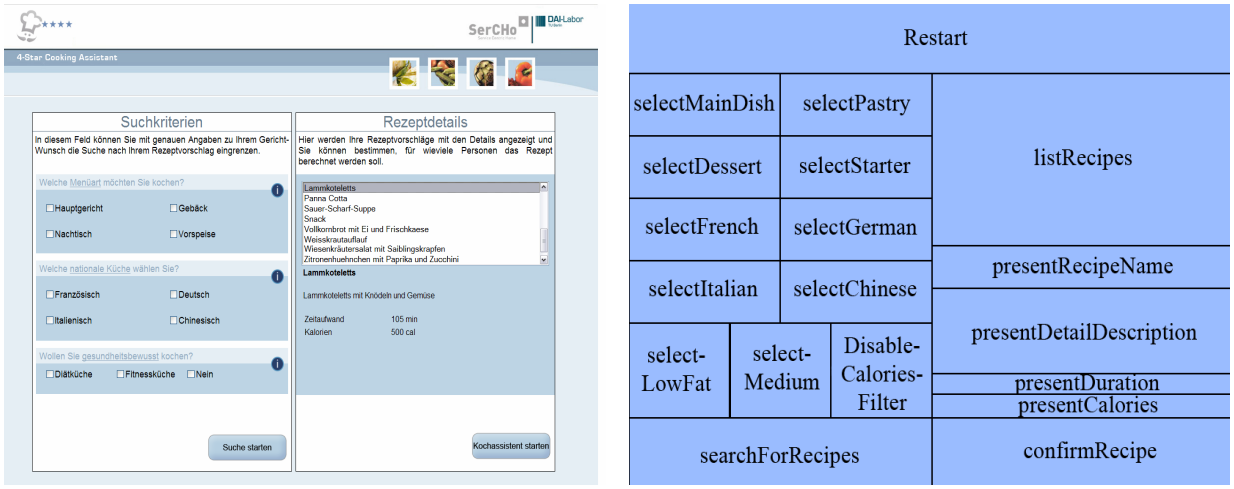


Figure 8: The screen for the recipe search and the final box-based layout result of the layout-model

As soon as such an event from the Distribution Model Agent has been received the Layout Model Agent reads the actual context-of-use and evaluates all of the layouting statements that are relevant for the actual user interface screen.

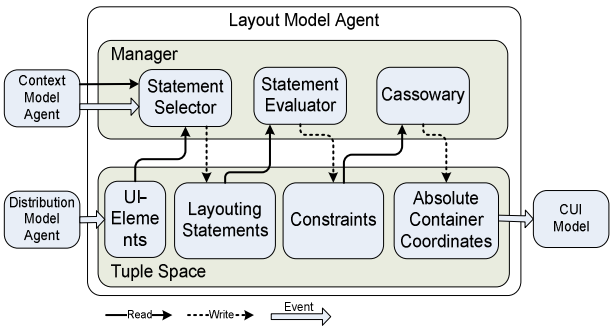


Figure 7: Each Model Agent is comprised of a manager that encapsulates the agent's functionality and a tuple space to store its data.

Figure 7 depicts the internal setup of a Layout Model Agent and its internal as well as its external communication. The agent senses for two external events to happen: First, for a new distribution of the user interface and second, for a change of the context-of-use. Both stimulate the agent to select and assemble the layouting statements. The selection of suitable statements is done by the following way:

1. Retrieve layouting statements for the actual context of use that have been specified independently from the application and that specify layout requirements to address a certain user or a specific device.
2. From the ordered statement list select the statements for a screen *s* that:
    a. address application wide layout interpretation
    b. address reoccurring elements that are used by *s*
    c. directly address the screen *s*
    d. are defined for this application and the relevant context-of-use scenario.

The statements that have been selected and ordered by priority are then evaluated to a set of constraints by the statement evaluator. Thereafter the layout agent finally solves the new constraint setup using the cassowary constraint solver [1]. Solving the constraints results in absolute positions for each element of the user interface that are stored within the layouting agent's own tuple space. The CUI Model Agent is registered for updates to the absolute positions and therefore receives updates for each change of these coordinates that the CUI Model Agent will use to re-position the user interface elements.

Different to other approaches that use a constraint solver to calculate the user interface layout, we introduced an additional level of abstraction for defining the user interface layout by a separate layout model that includes statements that are derived using an interactive and tool-supported process and are consistent to the other user-interface models. Since we decide at run-time which statements to evaluate to generate constraints, we can flexibly address layout adaptations to new contexts-of-use scenarios that can even be independently specified from an application but have been introduced together with a new device or a new kind of user type.

In the next chapter we present first results of an evaluation we did to test the efficiency of our approach. The evaluation has been done as part of a research project where we realized a multi-modal cooking assistant that supports the user in finding recipes, creating a shopping list and guides the user step by step through the cooking process.

6. EVALUATION

We tested our approach regarding two aspects: first, the efficiency at design-time for the designer to generate the layout model by using the layout model generator. Second we tested the efficiency of the implementation to generate and solve the constraints in our run-time system.

6.1 Design Efficiency

To test the design efficiency of the approach, we asked a designer to realize a layout for an interactive cooking assistant application based on a textual description of a scenario of how the cooking assistant should support the user. The designer created three screens and one user interface distribution scenario where one screen is split to two different devices: The initial screen asks the user to search for a recipe based on several search options. The second screen is about assisting the user to generate a shopping list by asking the user which of the

required ingredients are available and which are not available. This screen could be split into two parts where one part gets distributed onto a PDA that could be taken along during shopping and the other part remains on a touch screen in the kitchen. The last screen assists the user during cooking by offering multi-medial help, controlling the kitchen appliances and by splitting each recipe into a list of steps containing the required ingredients as well as a detailed description about what to do in each step. Figure 8 presents the initial screen for the recipe search as it has been realized by the designer and the result of the model-based layouting using the box-based layout of the editor.

Independently from the designer we asked a developer to follow a model-based development approach. Initially both, the designer and the developer shared the same textual description of a scenario for the cooking assistant. Based on the results of the model-based development approach including a fine grained task model, a domain model and an AUI model, we then derived a model-based layout that should correspond to the screens of the designer as close as possible. Finally we measured the amount of statements that have been required to end up with the same layout as the designer has realized.

Each screen has a different layout complexity consisting of a number of elements that are nested based on the abstraction level of the task model.

Screen	1)	2)	3)	4)	5)	6)	7)
1.Recipe Search	19	7	9	3	3	4	19
2.Shopping List	13	8	0	6	1	2	12
3. Distribution: PDA,Touch	4,9	8	0	1,2	0	0	0
3.Cooking Aid	15	10	0	2	2	4	8

Table 1: Complexity of the screens that need to be layouted and the amount of statements required. 1) Elements to layout, 2) Abstraction levels 3) Number of containment-, 4) orientation-, 5) order- 6) site-related statements 7) total amount of statements

Table 1 lists the level of complexity (number of elements, and the maximum nesting level utilized) for the three screens that have been sequentially layouted and the amount of statements that were required to realize the layout of the designer. After the first screen has been layouted the derived statements have been reapplied to the second screen and finally to the third screen. The second column of table 1 lists the different levels of UI-complexities that we have considered by the three screens: Whereas the RecipeFinder screen has a lot of elements (19) and a less nested structure of 7 levels, the Cooking Aid screen has 15 elements on 10 nesting levels as it is composed of various parts that are not directly related (e.g. the multi-medial help and the appliance control). For the ShoppingList screen two further layouts have been designed that are reflecting a distribution scenario where parts of the screen get distributed to a PDA (4 elements) and some parts (9 elements) remain on the screen. By analyzing the amount and type of statements that were required to layout the screens in the same way like the designer did, several observations have been made and are listed in the following paragraphs:

- Containment and order related statements can be derived from a task tree efficiently.
- If the task model is used to derive the containment and atomic tasks are identical to individual widgets, the introduction of further containment-related statements is required (for our application we required 8 containment statements for grouping checkboxes for the recipe search screen).
- Size related statements can be defined very efficiently on an application wide, global level based on the information of the design models (such as weighting input to output tasks, or by giving control tasks that usually end up presented as buttons a global minimum /maximum size restriction).
- The aspect ratio has to be defined pictures that should be presented within a task (using a size relational statement) which can be automatically derived at run-time when loading the picture.
- The orientation related statements can only be very limited specified on a global level but have to be reapplied for most of the individual screens. This is because our design models have no information that can be used to derive an initial orientation. So we applied a heuristic approach that produces elements with a balanced width to height relation by switching the orientation of the elements. Therefore we toggle the orientation horizontal to vertical and vice-versa, for each nesting level that has been derived from the task model.
- The container, order and size related statements of the layout model helped to assemble layouts for user interface distributions that have not been explicitly addressed at design-time. Orientation related statements caused problems as after a distribution has been initiated the re-orientation of the remaining user interface parts were not expected by the users.

6.2 Efficiency at Run-time

In order to check the run-time performance of generating and solving the constraints, we measured the performance of both the statement evaluation and the constraint solving separately.

Screen	1)	2)	3)	4)
1.Recipe Search	25	142	<1ms	14 ms
2.Shopping List	20	107	<1ms	8 ms
3. Distribution PDA,Touch	8,10	56,81	<1ms	8,10ms
4. Cooking Aid	23	130	<1ms	13 ms

Table 2: Complexity of the screens the need to be layouted and the amount of statements required. 1) Number of statements to evaluate, 2) Number of evaluated constraints 3) Measurement for statement evaluation (ms) 4) Duration for constraint solving (ms)

Table 2 shows the results of the performance evaluation for our cooking assistant application. For each screen we have measured the amount of statements that have been selected as relevant for layouting each screen (second column) and the amount of constraints that have been generated by evaluating the selected statements. It could be observed that currently an average of 5 to 7 constraints is generated by one statement. In the last two columns the measured average calculation time (of three runs) for selecting the required statements and the duration

for solving the generated constraints are listed: We could observe that the time to choose between the statements that are relevant for a specific situation was always under 1ms and the amount of constraints (and the amount of selected relevant statements) is related to the solving time. Thus, the bigger the difference between the overall number of layouting statements and the number of selected statements the shorter constraint solving times can be expected.

7. CONCLUSION

The information of the design models of a a model-based interface design approach can be interpreted to derive a layout model. We describe these interpretations by statements that create a layout model that we evaluate at run-time. This approach offers two advantages: First, since the statements are interpreting the design models, they ensure a consistent user interface layout and second, as the statements are evaluated at run-time, they enable flexible context-of-use adaptations even to situations that have not been directly considered during application design.

We are currently investigating further evaluations as the initially evaluation data is based on a relatively small application. Although we initially hoped to identify a set of predefined layout derivations based on preexisting design models that can be generally applied for all applications, we are now trying to classify application types and try to figure out if we can support the layout designer by proposing different statement sets based on the application type.

8. ACKNOWLEDGMENTS

We thank the German Federal Ministry of Economics and Technology for supporting our work as part of the Service Centric Home project in the Next Generation Media program.

9. REFERENCES

1. G. J. Badros and A. Borning; The Cassowary linear arithmetic constraint solving algorithm; In ACM Transactions on Computer-Human Interaction, 2001
2. M. Blumendorf, S. Feuerstack, S. Albayrak; Multimodal User Interfaces for Smart Environments: The Multi-Access Service Platform; Accepted as demo paper for ACM Advanced Visual Interfaces Conference 2008; Napoli, Italy
3. G. Calvary. et all; A unifying reference framework for multi-target user interfaces. In: Interacting with Computers, Vol. 15, No. 3. pp. 289-308, 2003.
4. J. Coutaz; PAC: An object oriented model for implementing user interfaces; In:SIGCHI Bull., vol. 19, no. 2, pp. 37--41, 1987
5. S. Feuerstack, M. Blumendorf, S. Albayrak; Prototyping of Multimodal Interactions for Smart Environments based on Task Model; Workshop on Model Driven Software Engineering for Ambient Intelligence Applications, European Conference an Ambient Intelligence 2007, Darmstadt, Germany.
6. J. Fogarty and S. Hudson; GADGET: A toolkit for optimization-based approaches to interface and display generation, 2003.
7. K. Gajos and D.Weld; SUPPLE: Automatically Generating User Interfaces; In: Proceedings of Conference on Intelligent User Interfaces 2004, Maderia, Funchal, Portugal; pp. 93-100, 2004
8. H. Hosobe (2001), A modular geometric constraint solver for user interface applications, *in* 'UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology', ACM Press, New York, NY, USA, pp. 91—100
9. K. Gajos and D. S. Weld, Preference elicitation for interface optimization, UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology, 2005 New York, NY, USA
10. J. Nichols, Brad A. Myers, Thomas K. Harris, Roni Rosenfeld, Stefanie Shriver, Michael Higgins and Joseph Hughes. "Requirements for Automatically Generating Multi-Modal Interfaces for Complex Appliances," IEEE Fourth International Conference on Multimodal Interfaces, Pittsburgh, PA, Oct 14-16, 2002a. pp. 377-382
11. L. Nigay and J. Coutaz, Formal Methods in Human Computer Interaction, Ch. Software architecture modelling: bridging two worlds using ergonomics and software properties, Springer Verlag, pp. pages 49-73, 1997
12. F. Paterno: Model-based Design and Evaluation of Interactive Applications. Springer Verlag. Berlin 1999.
13. K. Richter (2006), Transformational Consistency, *in* 'CADUI'2006 Computer-AIDED Design of User Interface V'.
14. A. Sears. Aide: a step toward metric-based interface development tools, pages 101–110, 1995
15. J. Vanderdonckt; P. Berquin, "Towards a very large model-based approach for user interface development," User Interfaces to Data Intensive Systems, 1999. Proceedings , vol., no., pp.76-85, 1999
16. J. Vermeulen, Widget set independent layout management for uiml, Master's thesis, School voor Informatie Technologie Transnationale Universiteit Limburg, 20

A Mixed-Fidelity Prototyping Tool for Mobile Devices

Marco de Sá, Luís Carriço, Luís Duarte, Tiago Reis
LaSIGE & Department of Informatics
Faculty of Sciences, University of Lisboa
Campo Grande, 1749-016, Lisboa, Portugal
{marcosa,lmc}@di.fc.ul.pt, {lduarte,treis}@lasige.di.fc.ul.pt

ABSTRACT
In this paper we present a software framework which supports the construction of mixed-fidelity (from sketch-based to software) prototypes for mobile devices. The framework is available for desktop computers and mobile devices (e.g., PDAs, Smartphones). It operates with low-fidelity sketch based prototypes or mid to high-fidelity prototypes with some range of functionality, providing several dimensions of customization (e.g., visual components, audio/video files, navigation, behavior) and targeting specific usability concerns. Furthermore, it allows designers and users to test the prototypes on actual devices, gathering usage information, both passively (e.g., logging) and actively (e.g., questionnaires/Experience Sampling). Overall, it conveys common prototyping procedures with effective data gathering methods that can be used on ubiquitous scenarios supporting in-situ prototyping and participatory design on-the-go. We address the framework's features and its contributions to the design and evaluation of applications for mobile devices and the field of mobile interaction design, presenting real-life case studies and results.

Categories and Subject Descriptors
H5.2. [Information interfaces and presentation] (e.g., HCI): User Interfaces–*Evaluation, Prototyping, User-centered Design.*

General Terms
Design, Experimentation, Human Factors.

Keywords
Mobile Interaction Design, Prototyping, Usability, Evaluation.

1. INTRODUCTION
Designing for mobile devices is an increasingly demanding challenge. Besides the hardware constraints that are imposed by their size, interaction modalities, diversity and portability, their pervasiveness and multi-purpose functionality imply an entire new set of usage paradigms.

As a consequence, new design approaches are required, particularly for the evaluation and prototyping phases. The absence of specific methods and techniques is patent [14], which leads to none or to incomplete, and definitely inadequate, evaluations. In fact, these stages are usually supported by common methods which are impracticable or not suited to mobile scenarios, generally neglecting their ubiquitous nature.

During the design of a few applications directed to PDAs we were faced with several recurrent problems: (1) the prototyping techniques found in the literature and commonly used for desktop applications were inadequate to the ubiquitous nature of our applications; (2) prototypes started to mislead users due to the used material hindering, at times, their participation on the process and were limited regarding some of our goals; (3) lab experiences proved to be insufficient while determining usability issues with the developed prototypes; (4) techniques such as the Wizard of Oz or direct observation posed restraints to our evaluation since they were extremely difficult to apply on real world settings and (6) methods such as Experience Sampling Method (ESM) [5] or diary studies required extra effort and, although providing qualitative data, did not cover the interaction details that we wanted to evaluate.

These difficulties propelled the adoption of alternative techniques and experiences that brought out very positive results [23]. One of the main contributions that came about from this process was the integration of several functionalities and tools into a specific framework that entangles various techniques for mobile prototyping and evaluation purposes. The framework takes into account previous work within this area, the lessons that we learned and introduces new contributions that foster participatory and continuous in-situ design supporting designers, and the design's evolution, through the initial stages of user-centered design of mobile applications.

We start by addressing the existing work in this area. Afterwards we describe our tool's concept, its goals and novel contributions also detailing its architecture and features. We then present some already achieved results and delineate future work directions.

2. MOTIVATION/RELATED WORK
Design methods and techniques for mobile devices, albeit being recent and somewhat immature fields of research are increasingly being addressed by researchers, leading to the appearance of different approaches for a wide range of problems [13]. Unsurprisingly, given their differences from desktop systems, most efforts have been directed towards prototyping and evaluation, with some references also pointing to the generation of UI design guidelines specific for small screens [2].

Regarding prototyping, new techniques and orientations, particularly for low-fidelity prototypes, have been introduced [23]. These suggest the need for more detailed and carefully built

AVI'08, May 28-30, 2008, Napoli, Italy

prototypes that offer a more resembling picture of final solutions and their characteristics [10]. In fact, the adopted prototyping technique can be determinant during the consequent evaluation stages, allowing users to freely interact with them, improve them and use them on realistic settings without misleading users [23]. Furthermore, to assert on various details that might be relevant at different stages of prototyping, the concept of mixed prototyping has emphasized the need to create different prototypes to evaluate different dimensions of usability [20]. On these aspects, prototyping tools can play a paramount role, allowing designers to maintain their sketching and writing practices while creating prototypes that can actually run giving users a more tangible and realistic feel of the future application.

DENIM [19] and SILK [17] are two prototyping tools that give designers the ability to quickly create sketch-based prototypes and interact with them on the computer, also including the possibility of replacing drawn components with actual programmatic components. More recently, systems such as SketchWizard [6] or SUEDE [16] have also emerged, supporting new modalities and interaction modes such as pen-based input on the former and speech user-interfaces on the latter. Ex-A-Sketch [9] also allows designers to quickly animate sketches drawn on a whiteboard. On a different level, the BrickRoad project [23] also supports the design of location-enhanced applications, especially during early design stages.

However, although these tools have useful functionalities and features, and provide sketching and quick prototyping mechanisms, the integration with the evaluation stages is rarely addressed. Moreover, the evolution from early based sketches to more advanced prototypes is only present on SILK and DENIM which lack crucial components (e.g., sound) and active behavior or are deeply focused on specific domains. Furthermore, none addresses the specific needs of mobile devices or provides usability guidelines and aids to designers while creating their prototypes. Nevertheless, the automatic support for Wizard-of-Oz prototypes and the ability to animate hand drawn sketches has shown very positive results.

As aforementioned, problems are felt again when evaluating the developed prototypes. Although some recent studies reflect an increasing amount of attention towards contextual evaluation, out of the lab, its relative inexistence contrasts with the importance and benefits it presents to mobile devices [7],[21]. Existing examples usually point guidelines on how to emulate real world settings within labs [1],[15] or provide solutions [5] that are useful as a complement but, even if obtaining positive results, do not address specific usability problems, do not provide quantitative data and focus mainly on user satisfaction. Furthermore, they show little regarding user interaction towards the applications.

Some recent approaches have also addressed this stage of design, focusing methods to gather usage data remotely through active – requiring user intervention - (e.g., ESM, Diary Studies) and passive modes – without user intervention - (e.g., Logging). For instance, with close goals to our framework regarding evaluation, the Momento [4], and the MyExperience [8] systems provide support for remote data gathering. The first relies on text messaging and media messaging to distribute data. It gathers usage information and prompts questionnaires as required, sending them to a server where an experimenter manages the received data through a desktop GUI. On the second, user activities on Mobile Phones are logged and stored on the device. These are then synchronized depending on connection availability. The logging mechanism detects several events and active evaluation techniques can be triggered according to contextual settings.

However, and although some goals or used techniques are similar, our approach intends to provide qualitative and quantitative information that can be easily understood by non-expert users, focusing on interactions that directly relate to the developed prototypes on very early stages. Our goal here is to integrate the prototyping and evaluation stages seamlessly, facilitating user involvement and the design process. Moreover, none of these approaches integrates the prototyping and evaluation on real devices, also including means adjust the prototypes while evaluating them or to analyze them (e.g., various alternatives to one user interface), individually or simultaneously, on an easy-to-read video-like mode. Furthermore, most depend on server-client architectures, requiring a constant connection or frequent synchronizations. Still, these systems and recently conducted experiments [12],[18] validate the need to undertake evaluation on real-life settings using both passive and active data gathering techniques, even at a very early design stage.

3. CONCEPT, GOALS AND FEATURES

To cope with early design stage difficulties, which pertain both to prototyping and consequent evaluation, the developed prototyping framework's features cover both these stages, supporting an iterative and participatory design that facilitates the transition between them.

Its umbrella goal is to support the early design stages of applications for mobile devices. Like some of the aforementioned frameworks [16],[19] it provides designers with tools to quickly create prototypes and evaluate them, focusing specifically mobile and handheld devices. It supports in-situ and participatory design and enables designers to use both passive and active evaluation methods. The framework allows the construction of low, mid and high-fidelity prototypes and extends its automatic Wizard of Oz usage through their evaluation, also providing means to analyze the gathered data.

More concisely, on the prototyping stages we aim at: (a) supporting a visual, quick and easy design of realistic mobile prototypes, with flexibility regarding their fidelity (b) offering expert users or users without any programming knowledge the possibility of building or adjusting their prototypes; (c) allowing and promoting participatory design and prototyping during outdoor evaluation sessions within realistic settings.

For the evaluation stage our goals are: (a) retrieving reliable usage information without intrusive equipment, without the designer or usability engineer's presence and using seamless/passive techniques; (b) supporting the analysis of usage patterns and usability concerns through the visualization of the user's activities and (c) the integration of methods such as probing [11], ESM [5] and diary studies extending the scope of the evaluation process.

Our main contributions over previous work are the convergence of prototyping and evaluation techniques into one end-user tool, supporting several degrees of fidelity, allowing the comparison of design alternatives, and suggesting new ones if available,

facilitating the detection of usability problems or design flaws on early design stages. This coverage is extended to their evaluation on various stages of design within the context in which they are most likely to be used (e.g., device, location, environment), always centering its procedures on the user. Within these, the framework also supports in-situ participatory design, directly on the targeted devices. Globally, this can be achieved through the following features:

1. Prototyping with mixed-fidelities (thus analyzing different usability dimensions). Hand drawn sketches or interactive visual pre-programmed components can compose different prototypes with varied levels of visual refinement, depth of functionality and richness of interactivity [20], comparing different design alternatives, evaluating button sizes, screen arrangements, element placement, interaction types, navigation schemes, audio icons, and interaction modalities, among others.

2. Direct prototyping on the mobile devices. Users are able to update prototypes on mobile devices, re-arranging simple details and improving the prototypes during evaluation sessions on real settings, out of the lab. The overall building mechanism is simple, visual or wizard based allowing experienced designers or inexperienced final users to adjust their own prototypes. This enables its usage for probing purposes [11], promoting experimentation and on-the-fly design of new solutions for and on the context in which the user is interacting with the tool.

3. Integrated usability guidelines for mobile devices on mid-fidelity prototypes. When prototypes are created using the visual primitives and components, usability guidelines can be automatically enforced, if chosen by the designer/user. For instance, the location of each component, the actual size of the component or even the amount of information per screen can be automatically arranged. These guidelines are configurable and can be domain oriented (e.g., e-health – special icons, education - limited content, media players). The framework is also able to provide alternative versions of the created prototypes (e.g., displaying a similar prototype that uses radio buttons instead of a combo-box).

4. Avoid cargo cult syndrome [10]. By using actual devices, problems regarding the device's characteristics (e.g., size, weight, screen resolution, shape) emulation are solved, allowing their utilization on realistic settings. This provides users a much more tangible and realistic usage experience.

5. Automatically support the Wizard-of-Oz technique. By adding behavior to the digitalized sketches or by using visual components, users can navigate through the prototype without having to explicitly replace the screens by hand or without the presence of a designer to do so.

6. Gather data through passive and active techniques. On the former, every action that the user takes is automatically logged with customized granularities. On the latter, the use of ESM and diary studies, integrated within the tool, provides another source of data and usability information. Integrated questionnaires can be popped during or immediately after using the prototype, or even automatically during the day according to specific settings (e.g., if the user is unable to achieve a specific goal or is continuously failing to press a small button).

7. The framework also includes a log player which re-enacts (through a video-like mode) all the users' activities with accurate timing and interaction details, attenuating the need for direct observation.

4. FRAMEWORK/ IMPLEMENTATION

In order to support the aforementioned functionalities, the prototyping framework is divided into several tools.

4.1 Prototype Building Tools

The first tool, the prototype builder is divided into two modes. The first is a wizard-based user interface that guides users to create a prototype screen by screen. It supports the definition of the prototypes' fidelity, degree of functionality and behavior (Figure 1). It allows users to create each screen individually, organizing them sequentially and customizing them according to their needs. The second mode is the advanced mode. Here, designers can easily drag and drop the selected components, use hand drawn sketches, pictures or images for multiple screens, also arranging the "prototype's wireframe" or storyboard (Figure 2).

Figure 1. Prototype building tool – Wizard Mode.

On both modes, interactive output components (e.g., combo-boxes, labels, images and audio files) can be used. Input components (e.g., text data entries, sound recorder, and video recorder) are also available. These components are used to create mid to high-fidelity prototypes. For low-fi prototypes, sketches (hand-drawn and scanned or digital drawings) can be easily imported and their behavior adjusted. For each component, different configurations are also available (e.g., multiple-choices through radio buttons or combo-boxes).

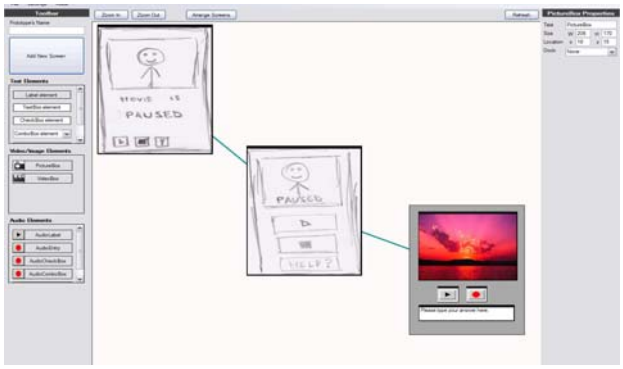


Figure 2. Prototype building tool – Advanced Mode.

The prototype's behavior can be defined within three levels: a component/element, a screen behavior and a global behavior. On the first users can define the behavior when using an individual component (e.g., a button press displays a warning). The second

defines the behavior for the entire screen (e.g., the user missed two of the screen's components and these are highlighted) and the third for the entire prototype (e.g., a questionnaire is popped once the user reached the fifth screen).

Each prototype is specified in XML and stored within a file that contains its specification which can be transferred and updated even without using any specific tool. The tool's modularity allows different components to run on different devices and systems.

4.2 Runtime Environment and Logging

The counterpart of the previous tool is the runtime environment. This tool is responsible for materializing the prototypes on the targeted device. Currently we have a runtime environment for Windows Mobile, Palm OS and SymbianOS. It is composed by a straightforward user interface that displays a list of the available prototypes for users to select. Once a prototype is chosen, it will be displayed and users can interact with.

The runtime environment also offers options to edit the prototype, save usage information at any given point or to define the granularity of the saved data. On the editing mode, every component's location, size and some content can be updated or changed. Screens and components can be deleted or their sequence arranged (e.g., card/screen-sorting and in-situ design).

Integrated within this runtime environment there is also the logging engine which stores every event. Events range from each tap on the screen, each button press or even each character that was typed by the user. Events are saved with a timestamp, allowing its reproduction for the re-enactment of the usage behavior. Other details such as the type of interaction, location of the screen tap, etc., are also stored for filtering purposes.

4.3 Analysis

The final tool pertains to the analysis of the logs generated by the logging engine. The log player resembles a "movie player" which re-enacts every action that took place while the user was interacting with the prototype. Several analysis granularities are provided ranging from each character that was typed to every visited screen. Pausing, stopping or adjusting the speed in which events are (re)played is also possible (e.g., fast-forward; double speed) through the options shown at the bottom of Figure 8.

The tools are available for desktop computers and, on a simpler version, for the abovementioned mobile platforms. The runtime environment is also available for desktop devices so that, if needed, designers can quickly review their prototypes before sending them to the mobile devices.

5. CREATING A PROTOTYPE

Following the traditional approach of low-fidelity prototyping, each prototype is composed by a set of screens (e.g., traditionally composed by paper cards). Each screen can be composed by a sketch, hand drawn and scanned to the computer or drawn using specific software. These are the lowest-fidelity prototypes where the screen is based solely on a digital version of a hand drawing made by the designer. Alternatively, as already mentioned, the framework includes visual components (e.g., drop-boxes, buttons, text-fields, track-bars, images, videos, sounds) that can be used to create a screen, alike the commonly used post-its. Screens are added as necessary and arranged on a storyboard to define their sequence (Figure 2).

At this stage, the degree and depth of functionality of the prototype can also be configured. If using a hand-drawn sketch for a low-fidelity prototype, "clickable" areas can be configured, generally over a drawn button or list. To do so, the designer visually drags a resizable rectangular area to the element he/she wants to make "clickable". Afterwards, these areas can be added with behavior (e.g., once they are clicked something happens).

On a higher fidelity prototype, the elements of the screen can be activated (e.g., drop-box contains a number of items, text-field receives an amount of characters) or de-activated (e.g., used solely for screen arrangement purposes). This allows us to test several dimensions. For instance, we can compare drop-boxes against lists or to text-fields or evaluate the location of each of these elements. Thus, it is possible to add or remove functionality to some degree or simply use the prototype for screen navigation, color, or button size tests.

Common components assume their traditional functionalities. Text-boxes allow text input; track-bars the selection of a numeric value, sound and video recorders record sound and video, etc.

5.1 Replacing the Wizard of Oz

Globally, the prototype's behavior is defined by selecting what buttons trigger the appearance of which screens, providing an automatic Wizard of Oz approach. Alternatively, these are arranged sequentially according to their location on the advanced mode, and their sequence on the wizard mode. Users can create warnings that can be popped up and shown according to specific triggers (e.g., selection from a drop-box or typing of a password). This mechanism is supported by three different types of rules.

The first ones are content-based rules, triggered when certain content, within a component, is chosen (e.g., if the user chooses yes or no from a list or a high or low value on a track-bar). Figure 3 on the left shows an initial PalmOS version of the mobile prototype builder. It depicts the adjustment of a rule that triggers a warning when the same answer is repeated 4 times.

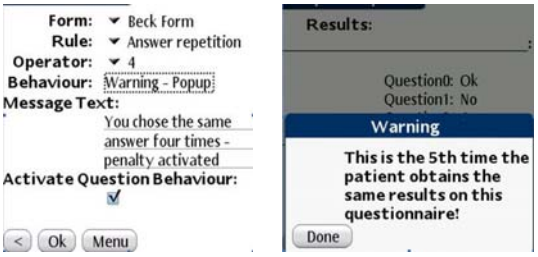


Figure 3. Rule definition and warning on a PalmOS PDA.

Time-based rules, on the other hand, are activated according to time limits (e.g., the user takes more than one minute to press a button or too long to answer a question within a questionnaire). Finally the interaction-based rules can be triggered according to the amount of taps on the screen, the location of those taps or the number of times a button is pressed.

To complement these rules and to function in concert with them, there are three types of behaviors. The first one is the "jump to" action. As the name indicates, once activated, it will automatically force a jump to a designated screen. For instance, using a "click area" that triggers a "jump to" behavior allows a user to configure an active hand drawn button, on a sketch-based prototype, to jump to the following, previous or any other screen/sketch once it

is selected. This mechanism allows designers or users to define navigational constraints without writing code or programming, replacing the designer on his sketch and component removing/inserting activities (e.g., Wizard of Oz technique).

The second type of behavior is composed by warnings (Figure 3 – on the right). Popping a warning alerting the user that he/she selected the top value, or did not select any value from a track-bar is a simple example. The third type of rule hides or shows components (e.g., if the user selects an option from a combo-box, the correspondent data entry field is shown).

These rules and correspondent behaviors, when used together, allow designers to compose fairly elaborated prototypes. However, they still maintain the necessary simplicity to be easily specified by end-users as well, through a simple to use, selection-based wizard interface.

Prototype files can be dragged directly into the device or can be transferred automatically through the building tool, if a connected device is detected. Once on the device they can be directly used on the runtime environment.

5.2 Reviewing Logs and User Behavior

Since one of the main goals of mobile evaluation is to evaluate the users' behavior on real scenarios, we intended to replace, as far as possible, direct observation with a similar mechanism. Therefore, several visualization options for the usage logs are available (e.g., event lists, selection tables). However, the most interesting one presents an exact replica of the users' behavior, emulating the mobile device and re-enacting every tap on the screen, every typed character and so on (Figure 8). Although these logs are limited to the direct interaction that the user has with the device, they still present enough detail to compare different design choices, evaluating navigation options, component placement and size, audio icons, audio volume, synthesized-text, the prototypes' feasibility and other questions that designers face on the early design stages.

6. CASE STUDIES

We have used the prototyping framework to generate and evaluate a set of prototypes on two different domains. On the first, psychotherapy, we developed low and high-fidelity prototypes for several therapeutic tools [3]. On the second, education, teachers used the framework to create different elaborated prototypes [22].

6.1 Psychotherapy

The first case study involved a team of mobile interaction designers and a team composed by a group of cognitive behavioral researchers and practicing psychotherapists. The main goal was to continue an on-going project which aimed at the support of cognitive behavioral therapy through the use of mobile e-artifacts [3]. Given the highly ubiquitous tasks that are encompassed within such type of therapies and the critical domain of healthcare in which we were working, the introduction of the framework and its functionalities aimed at facilitating the expert team to participate on the process and the quick construction of prototypes that could be easily evaluated and tested by therapists and patients. Moreover, initial tests with paper prototypes were misleading therapists regarding usage possibilities resulting on a constant rejection of most of the design team's ideas. Therapists had difficulties imagining and materializing the end result based on sketches and paper-based prototypes.

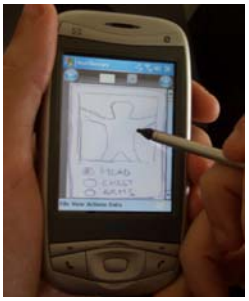


Figure 4. User interacting with a low-fi prototype for a pain therapy application on a SmartPhone.

Accordingly, several iterations of low-fidelity prototypes, that had been previously drawn were digitalized and used by the designers on the framework. "Click Areas" were defined and their behavior configured. These sketch-based prototypes were tested by therapists and some psychotherapy students, on smartphones (Figure 4) and later evolved to higher-fidelity ones, that allowed user input and reacted to usage behavior (Figure 5).

At this stage, therapists and researchers from the psychotherapy team started to use the framework as well, mainly to adjust the already developed prototypes (e.g., changing some interaction types). As the prototypes started to refine, the therapists handed the new versions to some students and used them on experimental therapy sessions within the research laboratory. Throughout this process, tools for anxiety, depression, pain therapy and associated disorders, were created and thoroughly evaluated.

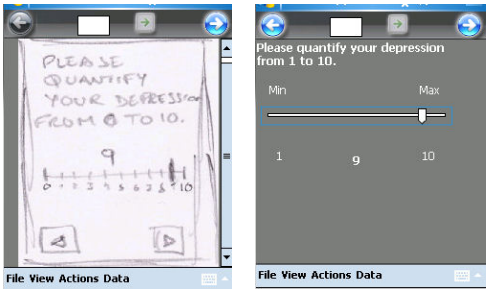


Figure 5. Left: Sketch-based low-fi prototype for a psychotherapy tool and its evolved high-fi software version.

Once most of the created prototypes had been experimented and adjusted in-situ by both therapists and researchers and on some experimental sessions, all the logs were carefully reviewed by both teams. Whereas the design team was focused on interaction details and on usability assessment, the therapists started to detect hesitations and symptomatic behaviors while users interacted with the prototypes. For instance, using the log player, therapists were able to detect questions where users spent more time or thoughts that were constantly written and deleted. Some of these behaviors led to the identification of critical subjects and to the detection of underlying problems that patients faced but did not mention during their face-to-face therapeutic sessions.

Overall, the prototypes were very well accepted and the tested versions, with some adjustments, were even used as final applications.

6.2 Mobile Learning

The second domain in which the framework was used was education [22]. In this case, the main goal was to design and evaluate a possible application for students to use, while at school or at home, to complete tests, homework, to review content provided by them. A team of 3 designers and one composed by 4 teachers were involved on the entire process as well as students for the final evaluation sessions.

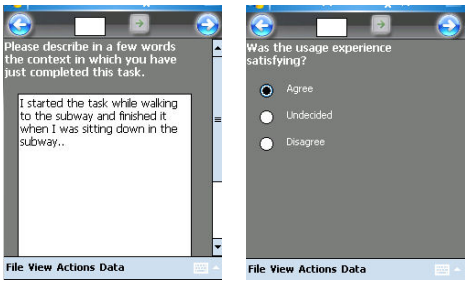


Figure 6. Questionnaire shown while using the prototype.

Teachers aimed at creating an easy to use tool that could convey the possibilities of assessment and task completion by students of various ages, as well as the access to relevant content that would be provided to students as necessary. The design process started with a set of meetings where requirements were established and ideas started to emerge, especially from the teachers' side. Given the successful experience with the previous case study, teachers were provided with the prototyping framework since the beginning, and a short tutorial (1 hour – wizard mode) was given to all the involved teachers. The functionalities were explained and the results gathered from the previous experience were described. Accordingly, teachers were given the framework and created a set of prototypes for applications that would allow students to achieve various different activities (e.g., watch a short movie, read a short book, complete tests or homework).

Given the functionalities that were explained, teachers started by creating low-fidelity prototypes for all the tools and were concerned mainly with the aesthetics, content organization, vocabulary and features of each tool. Once the prototypes were created, the design team conducted a series of evaluation sessions, with the teachers, in order to assess each of the low-fi prototypes for the targeted tasks.

The evaluation sessions were carefully planned, including a detailed description of the goals, the tasks that had to be performed, the student profiles that would be used and the locations and settings in which all the sessions would take place. In this last aspect, particular care was taken to select scenarios and settings with different conditions, regarding light, noise, user posture (e.g., walking, seating, etc) and the introduction of casual distractions (e.g., interrupting the user to ask a question, requesting the user to walk on a busy corridor, and so on). Overall, we tried to conduct the evaluation sessions on the most realistic settings possible. On some of the evaluation sessions, the students that tested these low-fi prototypes even took the devices and prototypes home with some pre-determined tasks to complete. In these situations, in order to have a glimpse of the context of use, specific questionnaires were included on the prototypes and automatically shown while using the prototype (Figure 6).

Once all the initial evaluation sessions, with the low-fi prototypes were complete, both teachers and designers started to look at the logs. Results from this process were naturally taken into account on the higher-fidelity versions of the prototypes.

On the following design cycle, teachers and designers continued to collaborate and began to create high-fidelity prototypes for the same tools. Based on the low-fi prototypes and using the available components, teachers replaced sketches with pictorial based tools for younger children and more textual (e.g., track-bars, textboxes) prototypes for teenagers or adults. Again, after a set of prototypes for each tool was created, evaluation sessions were conducted.

Half of the total of 6 tests, involving 36 students, took place at the university campus while the rest was done at various locations, including students' homes. The campus tests were filmed using a low-cost mobile kit developed specifically for this purpose (Figure 7). The initial kit used a shoulder camera. However, this approach, although capturing the user interaction with the device, provided little information regarding the context and the user's interest points. Various mobile devices (e.g., with and without keyboards were handed to the students). On one of the selected tasks, students were required to complete a test at school and another at home. Students used the prototypes to respond to tests and were free to use the devices to whatever they wished. After the tests were completed and usability questionnaires responded, students returned the devices to teachers and logs started to be analyzed, together with all the footage that was captured.

Figure 7. Mobile Video Capturing Kit.

The design team quickly detected some problems with the prototypes, particularly referring to the selected interaction modalities and the locations in which the prototypes were used. For instance, track-bars and text-boxes were difficult to use while walking or on the bus/subway whereas lists and multiple-choices (e.g., radio buttons) were easily handled. While for text-boxes this was already expected since keyboards, either physical or virtual, have to be used, for track-bars this came as a surprise.

From an educational point of view, and based on the suggestions made by the designers, teachers also tried to detect student difficulties while using the prototypes. To achieve so, and to isolate difficulties that could pertain to the components, or to the user interface itself, teachers connected three different aspects to detect learning issues.

Accordingly, to identify possible problems, it was necessary to search for questions that were often revisited, that took a long time to respond and where values were frequently changed/updated. These three aspects together excluded situations where the student could have left the device unattended or questions where students had to write instead of selecting an option. It also excluded questions that were only revised instead of edited and so on. This process allowed teachers to identify difficult subjects, preferred content and component adequacy to each age level or provided material.

Overall, both teachers and students were very pleased with the prototypes suggesting new case studies and features. Teachers appreciated the possibility of monitoring students' activities while away from classes, on a deferred mode, with the ability to define their own tools and with the inclusion of behavior and hints on content that was previously passive.

6.3 Mobile Interaction Design Implications

Regarding the design process and the usability questions that were found during the two processes, the framework allowed designers to work closely with the expert teams, easily sharing concepts and their visions through the prototypes. Furthermore, the short prototyping cycles allowed expert-users to quickly assess the feasibility of such systems even on real-case scenarios. These experiences were even more successful since the expert teams and final users were able to use the actual devices, resulting in a much more confident evaluation process, where users were actually involved and on their working environment.

Overall, the framework provided a large step forward during the design process and led to much more efficient results and collaborations. From the usability and design team standpoint, the usage of low-fidelity sketch-based prototypes and high-fidelity prototypes provided interesting results, allowing users to actively prototype their own applications and providing a softer and sounder transition between design fidelities.

Log revision also led to interesting findings. For instance, track-bars, although not requiring text input, raised some difficulties mainly given the small size of the interactive counter. Moreover, when completing a task, if users were seated, they usually used the device's QWERTY keyboard. However, once walking they preferred to use the virtual keyboard, using one hand to hold the device and the other to tap on the virtual keyboard, alternating with any other activity that required their hand. Curiously, once seated again, they would not return to the physical keyboard. Also, while walking, accuracy towards buttons was much lower.

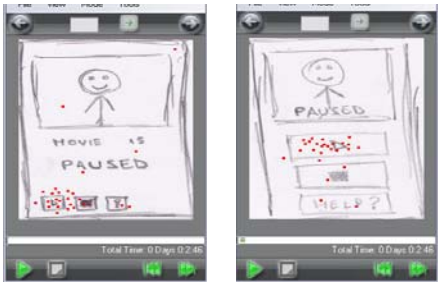


Figure 8. Two different iterations of sketch-based prototypes analyzed on the log player.

Figure 8 shows two screenshots of a low-fi prototype for the movie player being analyzed on the log player. Since all the logs have time-stamps and are cataloged by date, it was simple to correlate the logs and the locations/settings from which they resulted. Moreover, even specific portions of each evaluation session could be identified (e.g., at the beginning of the test, the user was seated; at the end of the evaluation test, the user was walking to another class). These situations were mapped to parts of the log where we noticed different accuracies regarding button selection and interaction, which allowed us to see that most of the missed taps on the screen referred to the situations where users were walking. As expected, while they were seated, accuracy was much higher. However, the log analysis provided a fairly precise idea of the necessary size and location for each button.

On the left side, a first prototype shows that users had some difficulties while using the video controls. This was particularly true when users were walking. On the right side, a second version of the same prototype, with larger buttons, shows that user accuracy, while selecting and using the controls was much higher. Each of the dots marked on the prototype identifies a tap on the screen. These can be viewed simultaneously, as depicted, showing heat zones, or sequentially, based on the actual user behavior.

Other results showed that components placed too close to the edges of the screen also raised some usage difficulties, especially when students used their fingers instead of the device's stylus.

7. RESULTS AND CONCLUSIONS

Throughout the development of the aforementioned case studies, several issues became clear and new goals started to emerge as the prototyping and evaluation sessions took place. Our initial assessment objectives referred to the prototyping framework and to the outcome that it's designing and evaluation features would provide. On the first facet, the tool allowed quick and easy creation of prototypes with different fidelities. End-users were much more satisfied by using actual devices, getting real feedback and actively participated on all stages. The initial probing goals were achieved as users created their own prototypes, generating new ideas and tools while using the framework.

The designers that were involved in both case studies responded to usability questionnaires and were very pleased with the easiness and amount of features available in the framework. This was further validated since other experts (e.g., therapists and teachers), with no particular knowledge in prototyping or programming techniques, were also able to materialize their own visions and needs through the prototyping framework. Here, the usability guidelines played an important role, limiting the amount of components in each screen and automatically docking their location, suiting several devices and screen resolutions. The shorter prototyping periods and intensive participation of non-designers together with the various fidelities and customization possibilities were frequently praised by all the users.

On the evaluation facet, all the involved designers considered the revision of users' behavior, without the need for direct observation, extremely useful. In fact, this allowed the detection of several issues which translated directly into UI improvements. Results were particularly interesting since they focused not only on a wide variety of contexts but also allowed the detection of problems that emerged while transiting between contexts. The logs and respective player provided insight on navigation patterns, size and location of components, amount of text, font size, among others. The different fidelities in concert with the realistic usage experience, since users roamed through different contexts with actual devices, allowed the evaluation of UI layouts, color arrangements, components, even detecting what colors were more adequate to certain lighting conditions and in which locations the user interfaces needed more contrast. This information was complemented by the questionnaires that were prompted during their utilization, capturing contextual information on-the-spot.

Moreover, since users interacted with the prototypes without direct observation and on familiar settings, their behavior was more natural and allowed us to see a set of interesting behavior

patterns (e.g., keyboard usage, application exchange). Once again, the utilization of actual devices played an important role since it allowed the usage of the prototypes on devices with different screen resolutions, weight, size and interaction characteristics.

These results were even more interesting when used in conjunction with video equipment. The video capturing kit that we used was composed by inexpensive and common material available in our lab (e.g., backpack, webcam, laptop, and hat). Still, it provided very useful footage of users interacting with the prototypes and with the contexts through which they passed. By correlating the time-stamped logs and videos it was possible to detect, hesitations, reactions and usability problems as well.

The two case studies validated the positive influence of the prototyping and evaluation framework on the design process. Some of the findings resulted in modifications that were specific to the domains of each case study while others can be translated into generic guidelines that can apply to most mobile devices when used ubiquitously. The prototypes and evaluation sessions gave designers and other researchers the opportunity to assess the feasibility and adequacy of the envisioned applications on real-life scenarios. Moreover, the analysis of the evaluation data played an important role on research fields such as psychotherapy and education. In fact, a conclusion that was drawn from these experiences points the possibility of using the framework to create fully functional applications to support paper-based activities.

Given the positive results from these tests and experiences, we have integrated the framework into a new group version. Although beyond the scope of this paper, the team prototyping framework is worth mentioning and has benefited from the developments and results achieved through the experiences that we have presented. It introduces a set of features that, using the mobile prototyping framework, allow designers to cooperate in the creation and adjustment of designs, sketches and prototypes. The tool includes a large screen display module where several prototypes can be seen simultaneously. Moreover, it contains communication tools that provide means to visualize the evaluation sessions in real-time. Used in concert with the log player, it enables teams to review several logs simultaneously, comparing a user or a prototype's performance in various settings.

Finally, this work is part of and based on a complete methodology that was developed and aims at supporting the design of mobile applications through a user-centered design approach. Following a parallel research direction and acting as a complement for the prototyping framework, it compiles a set of guidelines that suggest the generation of scenarios and selection of appropriate contexts and techniques for the evaluation of mobile applications.

8. ACKNOWLEDGEMENTS

This work was supported by LaSIGE and FCT, through project JoinTS, through the Multiannual Funding Programme and scholarship SFRH/BD/28165/2006.

9. REFERENCES

[1] Barnard, et al. Capturing the effects of context on human performance in mobile computing systems. Personal and Ubiquitous Computing, Vol. 11, No.46, pp.81-96.

[2] Brewster, S. (2002). Overcoming the lack of screen space on mobile computers. Personal and Ubiquitous Computing, 6.

[3] Carriço, L., Sá, M. Hand-held Psychotherapy Artifacts. In Procs. of HCII 2005.

[4] Carter, S., et al. Momento: Support for Situated Ubicomp Experimentation. CHI'07, pp. 125-134, ACM.

[5] Consolvo, S. and M. Walker, Using the Experience Sampling Method to Evaluate Ubicomp Applications. IEEE Pervasive Computing, 2003. 2(2): pp. 24-31, IEEE.

[6] Davis, R., et al. SketchWizard: Wizard of Oz Prototyping of Pen-Based User Interfaces. UIST'07, pp. 119-128, ACM.

[7] Duh, H.B.-L., G.C.B. Tan, and V.H.-h. Chen. Usability Evaluation for Mobile Device: A Comparison of Laboratory and Field Tests. Mobile HCI'06. ACM.

[8] Froehlich, J., et al. MyExperience: A System for In Situ Tracing and Capturing of User Feedback on Mobile Phones. MobiSys'07, pp. 57-70, ACM.

[9] Hartmann, B., et al. Wizard of Oz Sketch Animation for Experience Prototyping. Adjunct Procs. of Ubicomp 2006.

[10] Holmquist, L., Prototyping: generating ideas or cargo cult designs? Interactions, 2005. 12(2) p. 48-54, ACM.

[11] Hulkko, S., et al. Mobile Probes. NordiCHI'04. ACM.

[12] Iachello, G., et al. Prototyping and Sampling Experience to Evaluate Ubiquitous Computing Privacy in the Real World. CHI'06, pp. 1009-1018, ACM.

[13] Jones, M. and G. Marsden, Mobile Interaction Design. 2006, John Wiley & Sons, England.

[14] Kjeldskov, J. and C. Graham. A Review of Mobile HCI Research Methods. Mobile HCI'03, pp. 317-335, Springer.

[15] Kjeldskov, J. and J. Stage, New Techniques for Usability Evaluation of Mobile Systems. International Journal of Human Computer Studies, Elsevier, 2003.

[16] Klemmer, S. R., et al, SUEDE: A Wizard of Oz Prototyping Tool for Speech User Interfaces. In Proc. of UIST'00, San Diego, California, USA, pp. 1-10, ACM.

[17] Landay, J. SILK: Sketching Interfaces Like Krazy. CHI'96, pp.398-399, ACM.

[18] Li, Y., Welbourne, E., Landay, J. Design and Experimental Analysis of Continuous Location Tracking Techniques for Wizard of Oz Testing. CHI'06, pp. 1019-1022, ACM

[19] Lin, J. et al. Denim: Finding a Tighter Fit Between Tools and Practice for Web Site Design. CHI'00, pp.510-517, ACM.

[20] McCurdy, M., et al. Breaking the Fidelity Barrier: An Examination of our Current Characterization of Prototypes and an Example of a Mixed-Fidelity Success, CHI'06, ACM.

[21] Nielsen, C.M., et al. It's Worth the Hassle! The Added Value of Evaluating the Usability of Mobile Systems in the Field. NordiCHI'06. ACM.

[22] Sá, M., Carriço, L. Handheld Devices for Cooperative Educational Activities. *SAC'06*, pp. 1145-1149, ACM.

[23] Sá, M.d. and L. Carriço. Low-fi Prototyping for Mobile Devices. CHI'06 (extended abstracts), pp 694-699, ACM.

[24] Liu, L. A., Li, Y. BrickRoad: A Light-Weight Tool for Spontaneous Design of Locaton-Enhanced Applications, CHI'07, pp 295-298, ACM.

Gummy for Multi-Platform User Interface Designs: Shape me, Multiply me, Fix me, Use me

Jan Meskens Jo Vermeulen Kris Luyten Karin Coninx

Hasselt University – tUL – IBBT
Expertise Centre for Digital Media
Wetenschapspark 2, B-3590 Diepenbeek, Belgium
{jan.meskens,jo.vermeulen,kris.luyten,karin.coninx}@uhasselt.be

ABSTRACT

Designers still often create a specific user interface for every target platform they wish to support, which is time-consuming and error-prone. The need for a multi-platform user interface design approach that designers feel comfortable with increases as people expect their applications and data to go where they go. We present GUMMY, a multi-platform graphical user interface builder that can generate an initial design for a new platform by adapting and combining features of existing user interfaces created for the same application. Our approach makes it easy to target new platforms and keep all user interfaces consistent without requiring designers to considerably change their work practice.

Categories and Subject Descriptors

H.5.2 [**Information interfaces and presentation**]: User Interfaces – Graphical user interfaces, Prototyping

Keywords

design tools, multi-platform design, GUI builder, UIML

1. INTRODUCTION

There is an increasing need for applications that are available on multiple devices. Today, people tend to read their email or browse the web using their mobile phones or game consoles. This tendency will increase with the move towards ubiquitous computing where users are supposed to have seamless access to applications regardless of their whereabouts or the computing device at hand [20]. People need more means to access their information and applications than just a regular desktop computer. Applications that need to be available on any device at the user's disposal should be able to deploy a suitable user interface (UI) on each of these computing platforms.

We define a *computing platform* as the combination of a hardware device, an operating system and user interface toolkit. Designing a user interface for different computing platforms is far from simple. Each computing platform has its own characteristics such as the device's form factor, the appropriate interaction metaphors and the supported user interface toolkit. In current practice, designers often create a specific user interface for *every* target platform. Even though some technologies are shared between a number of devices (e.g. Java ME[1] or a modern web browser), usually each device still requires specific adjustment. There is a high cost incurred in adding a new target device and keeping all user interfaces consistent using manual approaches. Furthermore, there is no clear separation between the user interface and the underlying application logic.

A common solution to these issues is to specify the user interface in an abstract way by means of high-level models such as task models and dialogue models [9]. The platform-specific user interfaces are then generated automatically from this abstract description. The user interface has to be specified only once, which makes it easier to make changes or add a new target platform. In spite of the fact that these tools solve most of the problems with the manual approach, the resulting user interfaces usually still lack the aesthetic quality of a manually designed interface. Furthermore, the design process is not intuitive since designers have to master a new language to specify the high-level models and cannot accurately predict what the resulting user interface will look like [18].

GUMMY combines the benefits of both the manual approach and model-based techniques. Designers create and perform prototyping of a multi-platform graphical user interface (GUI) in the same way as they do when dealing with traditional GUI builders such as Microsoft Visual Studio[2] and Netbeans[3]. GUMMY builds a platform-independent representation of the user interface and updates it as the designer makes changes. This allows for an abstract specification of the user interface while keeping the design process intuitive and familiar. An abstract user interface specification avoids a tight interweaving of application and presentation logic.

GUMMY can generate an initial design for a new platform from existing user interfaces created for the same applica-

AVI'08, 28-30 May, 2008, Napoli, Italy.

[1] http://java.sun.com/javame/
[2] http://msdn.microsoft.com/vstudio/
[3] http://www.netbeans.org/

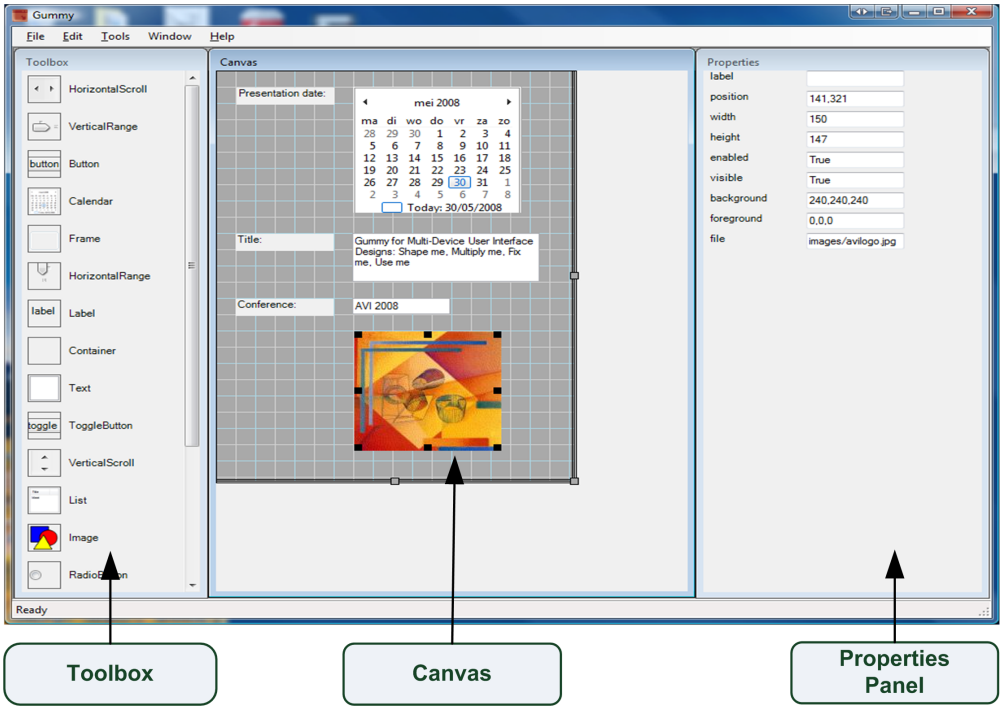


Figure 1: The three main dialogues of the Gummy tool

tion but for other platforms. This makes it it easy to target new platforms and keep all user interfaces consistent without requiring designers to considerably change their work practice. Since designers work with the concrete user interface, GUMMY allows for a true WYSIWYG[4] multi-platform user interface design process.

The main contributions we present in this paper are:

- a *multi-platform design approach* to creating user interfaces incrementally for a wide range of computing platforms while still working on a concrete level (Sect. 3).
- GUMMY, a *generic multi-platform GUI builder* to support the aforementioned design approach. This tool automatically adapts its workspace according to the considered target platform (Sect. 4). A preliminary user study with GUMMY indicated that the incremental design approach was faster than starting from scratch for each computing platform (Sect. 5).

2. BACKGROUND AND MOTIVATION

We feel that most designers prefer to have a concrete representation during their design activities, whether they are working on a single or a multi-platform user interface. This avoids their having to imagine what the final user interface would look like. We can thus conclude that working on a concrete representation reduces the mental burden on the designer [2, 18].

We structured GUMMY in a similar way to traditional GUI builders in order to allow designers to reuse their knowledge of single-platform user interface design tools. Fig. 1 shows the main components of the GUMMY interface: there is a *toolbox* showing the available user interface elements, a *canvas* to build the actual user interface and a *properties panel* to change the properties of the user interface elements on the canvas.

The resemblance to traditional GUI builders also reflects one of the main strengths of our approach: the abstractions used to support multi-platform user interface design are carefully hidden from the designer. This is contrary to existing multi-platform design tools that often expose these abstractions to the designer. In the remainder of this section we give some details about the underlying abstract language that GUMMY uses to represent multi-platform user interfaces and how they are presented to the designer.

The underlying language used in GUMMY is the User Interface Markup Language (UIML) [14], an XML language that contains a platform-independent *user interface description* on the one hand and a *mapping vocabulary* on the other hand. While the former is used to describe the structure, style, content and behaviour of a user interface using platform-independent terms, the latter contains mappings of these terms onto concrete widgets. Fig. 2 gives an example of a mapping vocabulary. The traditional rendering step is to translate the platform-independent user interface description into a concrete user interface using these mappings. For the implementation of GUMMY the opposite was done: the concrete representations were used in the tool and were internally mapped onto the associated abstractions. The designer works with the concrete graphical representations, avoiding the XML language, while the tool maintains a synchronised platform-independent UIML document for

[4]What You See Is What You Get

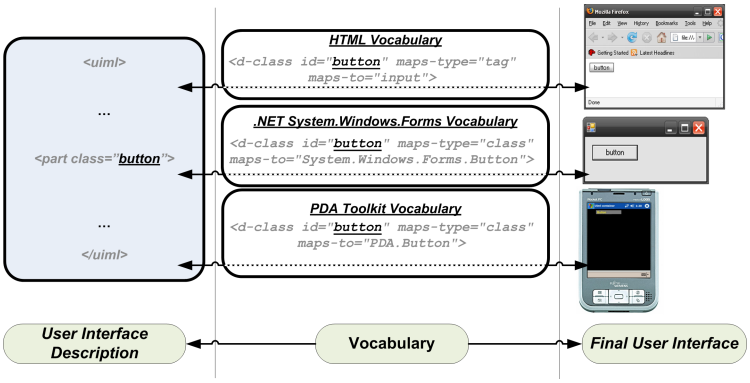


Figure 2: A UIML vocabulary relates generic terms to concrete representations

the concrete design.

3. MULTI-PLATFORM DESIGN APPROACH

Now to delve deeper into the design process supported by GUMMY, as depicted in Fig. 3. The design process is similar to that of traditional GUI builder tools, but includes an additional iteration to refine a design for a specific computing platform.

The procedure to create a user interface design for different platforms can be described in the following five steps:

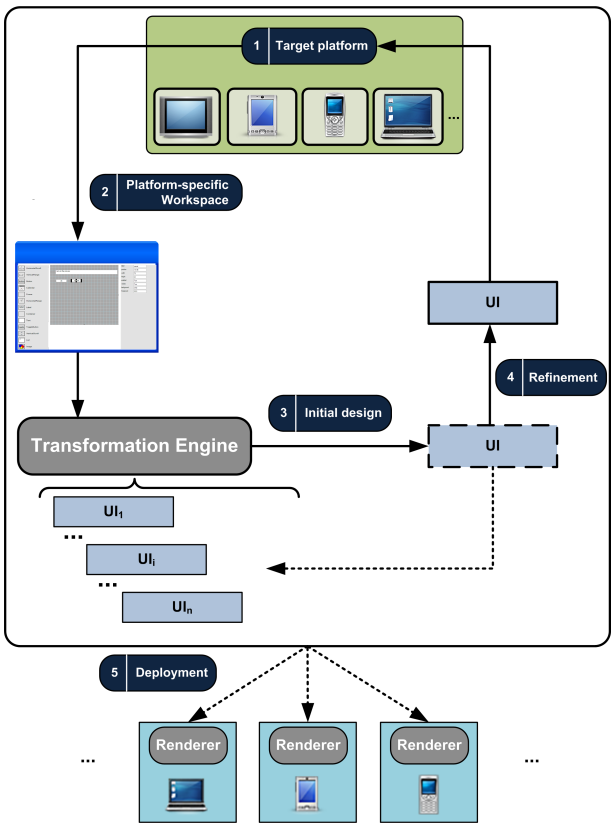


Figure 3: The approach presented in this paper to design multi-platform user interfaces

1. As a first step, the user interface designer specifies the *target platform* for which they want to design a user interface. Some possible platforms are a mobile phone with the Compact .NET framework, a digital TV with DVB-MHP, etc.

2. According to the specified platform, GUMMY automatically loads a GUI builder workspace that is fully equipped for designing user interfaces for this platform.

3. From a set of existing user interfaces that are all created for the same application, GUMMY automatically generates an initial design for the selected target platform. For this generation process, GUMMY relies on a *transformation engine* component. When there are no existing user interfaces available, this step generates an empty user interface for the selected platform. The specifics of the underlying algorithm to perform this transformation is not the main focus of this paper.

4. The designer can refine the initial user interface until it fits their vision. The resulting design is then added to the set of existing user interface designs where it may serve as an input for the transformation engine. After this step, a new iteration is started.

5. When the designer is finished, GUMMY exports all designs as one or more UIML user interface descriptions. Notice that platform-specific user interface descriptions might be needed to achieve aesthetic quality on every target platform. These UIML descriptions can then be rendered on the target platforms. In this paper a *UIML renderer* is defined as a component that can transform a UIML description into a working user interface.

4. A GENERIC MULTI-PLATFORM GUI BUILDER

Three aspects of GUMMY's architecture account for its generic nature:

- a *pluggable rendering architecture* which makes it possible to integrate any UIML renderer into GUMMY with minimal changes;
- UIML vocabularies to automatically *adapt* GUMMY*'s workspace* to a certain platform;

- a *transformation engine* that generates initial designs for new platforms based on existing designs for other platforms.

4.1 Pluggable Rendering Architecture

When a designer alters a user interface design in GUMMY, the underlying UIML description is automatically updated and visual feedback is provided immediately. GUMMY relies on an external UIML renderer to provide the visual representation of this UIML description. Different UIML renderers can be integrated into the tool as plugins.

The communication between GUMMY and a UIML renderer can be viewed as a set of inputs from GUMMY to the renderer on the one hand and a set of outputs from the renderer to GUMMY on the other hand. GUMMY will feed UIML descriptions of parts of the user interface into the renderer. In turn, the renderer will parse these UIML descriptions and render them as off-screen bitmaps. GUMMY then uses these bitmaps to visualise the underlying UIML descriptions.

In theory, every renderer that respects the communication protocol described above can be plugged into GUMMY. However, renderers might be written in other programming languages, run only on specific operating systems (e.g. embedded systems) or have limited communication possibilities. It would be inflexible to require GUMMY to know how to communicate with each of these renderers. To solve this problem, an additional layer of abstraction was added between GUMMY and the different UIML renderers. *Proxy objects* act as local placeholders for the UIML renderers and hide the communication details from GUMMY, as shown in Fig. 4. Every proxy object behaves like a regular UIML renderer but just forwards the rendering inputs to an actual renderer which can be located on any computing device. In turn, the bitmaps produced by the renderer are sent back to the proxy which finally delivers them to the design environment. Proxy objects are free to choose how they communicate with their UIML renderer, e.g. through socket communication, SOAP, etc. GUMMY's proxy objects are based on the *Remote Proxy* and *Adapter* design patterns [13].

4.2 Adapting the Gummy Workspace

In the GUMMY tool, the designer needs to specify the platform for which they want to design a user interface. GUMMY then automatically loads a UIML vocabulary designed for the selected platform. At the same time, it looks for a suitable UIML renderer for this vocabulary. In order to find a suitable renderer, all the available proxy objects need to be placed in a predefined location together with a configuration file that connects each renderer to a set of vocabularies that it can handle.

Once the vocabulary and renderer are loaded, GUMMY adapts its workspace to be fully ready for designing user interfaces for the selected platform. This platform-specific workspace will have a toolbox dialogue that contains only those user interface elements that are available for the selected platform. In order to generate this toolbox automatically, GUMMY uses the relation between generic terms and concrete user interface elements (see Fig. 2) described in the loaded UIML vocabulary. Small versions of all the concrete widgets that are described in the vocabulary are displayed as items in the

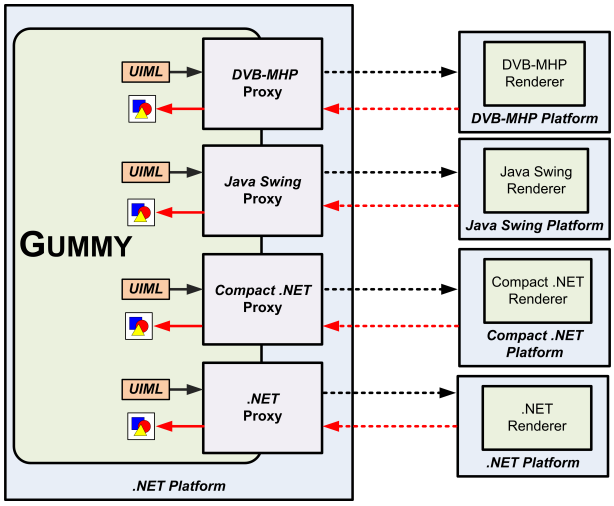


Figure 4: Existing UIML renderers can be easily integrated into Gummy using *proxy objects*

toolbox. The designer can then drag and drop these items onto the canvas. While designers manipulate concrete user interface representations on the canvas, GUMMY maintains a UIML description of the user interface in the background. Every time a designer repositions or resizes a widget through direct manipulation or modifies a property of a widget in the properties panel, the corresponding UIML description is updated and forwarded to the external renderer to update the view.

4.3 The Transformation Engine

GUMMY allows the designer to create several user interface designs for the same application. Each design corresponds to a specific target platform. During this process, a transformation engine is used to generate initial designs for new platforms based on the previously designed user interfaces. Designers can also simply copy one of the previous designs.

To prove the design approach that was introduced in Sect. 3, a basic transformation engine was integrated into GUMMY. This engine transforms existing designs into a new design based on the available screen size, as shown graphically in Fig. 5: Two existing interface designs I_1 and I_2, both representing the same application (a card game) but for two specific screen sizes, are used by the transformation engine to generate an initial design for a new screen size I_x.

The underlying algorithm used by this engine will not be discussed into detail since it is not the main focus of this paper. The transformation engine uses a set of rules to decide which properties (e.g. a widget's size or position) of the already created user interfaces should be selected and adapted according to the available screen size on the target platform. Each rule relates a property with a minimum and maximum screen size between which it is valid. For each property, the designer specifies these rules by manipulating a set of sliders that appear next to the design canvas as shown in Fig. 6. These sliders represent the vertical and horizontal screen size extrema within which the selected property is valid. Specifying these rules from scratch is a time-consuming activity.

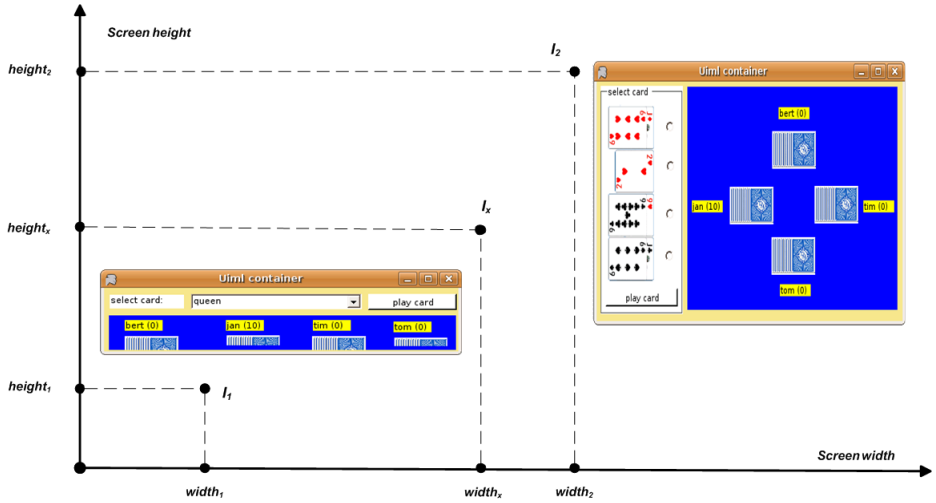


Figure 5: An initial interface for screen space I_x can be generated with the rule-based transformation engine.

Therefore, GUMMY automatically generates initial rules by using a heuristic based on the assumption that a component may only be displayed when it fits within the available screen space. The conceptual user study indicated that designers were faster when using the initial designs generated by this engine than starting from scratch for each computing platform (see Sect. 5.1).

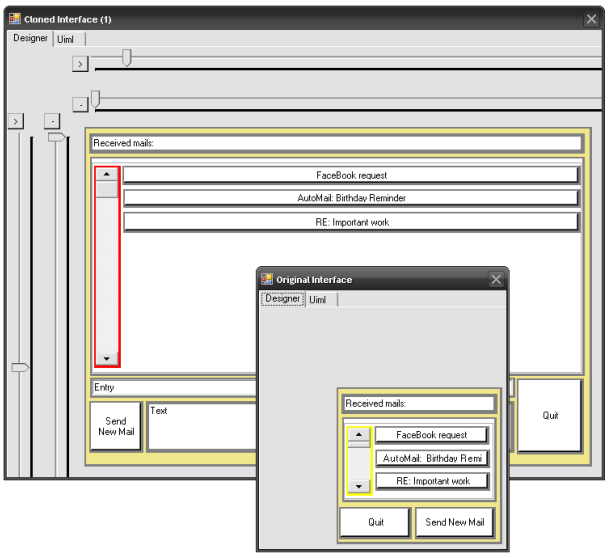


Figure 6: Two mail client user interfaces which are used as input for the rule based transformation engine

5. ANALYSIS

5.1 Conceptual User Study

To get an idea about the usability of the approach a small experiment was organized to assess the user's effectiveness at creating a user interface for multiple computing platforms with GUMMY. Ten test participants with various computer skills were recruited. Six of them were colleagues with good programming skills and a lot of experience with traditional GUI design tools. Four participants did not have a computer science background, of whom three did have experience with graphical drawing tools. The diverse test audience allowed examination of the question of whether the tool would require a certain technical way of thinking. In order to instruct all subjects in the same way before they started, they were provided with a written tutorial explaining the basic workings of our tool.

The test consisted of two parts. The first part of the test evaluated the ease of starting from an initial design for a new platform versus creating one from scratch. The test participants were divided into two groups with the same proportion of technical and non-technical people. Both groups had to arrive at a predefined user interface for a new platform. The first group had to create this user interface from scratch whereas the second one was allowed to base their user interface on the initial design generated by the transformation engine. By performing a one-way analysis of variance, it was determined that the subjects who were able to use the initial design were significantly faster in obtaining the desired user interface than the members of the other group ($F_{1,8} = 15.935, p < 0.005$).

In the second part of the experiment, the participants were asked to manipulate the transformation rules in order to obtain a predefined initial design for a new platform. Subjects rated the difficulty of this assignment on a *Likert* scale from *very easy* to *very hard*. A *Spearman rho* analysis indicated that there was a negative correlation between the programming experience of the test subjects and the perceived difficulty of the task ($p < 0.05$). This suggests that customising transformation rules (see Sect. 4.3) is not very intuitive for non-programmers. However, this does not invalidate the multi-platform design approach presented in this paper. The first part of the experiment gave an indication that changes to the transformation rules were usually not necessary. A more intuitive way of specifying transformations could resolve this issue.

5.2 Evaluation of Effectiveness

As was pointed out by Olsen [6], usability testing in its traditional form is rarely suitable for evaluating UI architectures, toolkits and design tools. Olsen argues that the three basic assumptions of usability testing, (1) minimal required training; (2) a standardised task to compare; and (3) being able to finish a test in short period of time, are rarely met by these systems. For GUMMY, at least the first two assumptions are not met. Because of this, the evaluation was extended with an analysis based on Olsen's evaluation framework.

This framework uses a number of attributes of good tools and methods to demonstrate that a particular tool supports them. The ones considered for GUMMY are:

- *reduce solution viscosity* with flexibility and expressive match;
- simplify interconnection and allow easy combinations to achieve *power in combination*;

5.2.1 Reduce solution viscosity

This implies that a good tool should foster good design by reducing the effort required to iterate on many possible solutions [6].

A tool is *flexible* if it allows the making of rapid design changes that can then be evaluated by users [6]. Since GUMMY allows designers to work on a concrete level they can easily make changes to the user interface using direct manipulation. These changes can be tested immediately by instructing GUMMY to deploy them to the appropriate renderer. Initial designs for new target platforms can be automatically generated using the transformation engine (see Sect. 4.3). These initial designs can again be easily modified (e.g. widgets can be moved, resized, deleted or remapped to another concrete widget) after which the changes can be evaluated. If necessary, designers can easily intervene and correct the tool.

The manual approach offers roughly the same benefits but only within the visual design tool for one specific platform. Designs for other platforms can neither be quickly created nor changed since they have to be recreated from scratch. While most model-based design tools support flexibility by allowing changes to the models and evaluation of the result, these changes take more effort than with GUMMY. For instance, while designers could remap widgets in these tools by altering the transformation model, selecting an alternative widget through direct manipulation is much easier. Due to its better expressive match, GUMMY requires less effort from the designer to intervene.

Expressive match is an estimate of how close the means of expressing design choices are to the problem being solved [6]. GUMMY allows designers to create a user interface for different platforms in much the same way as they do with single-platform visual design tools. A visual design tool is a better expressive match for the task of designing a (multi-platform) user interface than a tool to manipulate abstract user interface models. With model-based techniques, the connection between the abstract models and the resulting user interface is often not clear to the designer [18]. Thus, being able to visually design multi-platform user interfaces lowers designers' skill barrier.

5.2.2 Power in combination

Power in combination refers to a common infrastructure that can support new components to create new solutions [6]. This can be supported mainly by simplifying interconnections and by ease of combination. GUMMY accomplishes both.

Simplifying interconnections deals with reducing the cost of introducing a new component from N (connect to every other component) to 1 (just implement a standard interface) [6]. As we explained in Sect. 4, GUMMY can use any combination of vocabulary and UIML renderer. Every renderer just needs to supply a proxy object that implements a common programming interface in order to communicate with GUMMY. Traditional GUI builders and existing multi-platform design tools usually support only a fixed set of platforms. Adding a new platform to one of these tools often requires specific changes to its internals.

Ease of combination refers to the fact that it is usually not sufficient to be able to connect different components [6]. The connection should also be simple and straightforward. This is clearly the case here. GUMMY only requires renderers to provide a proxy object that conforms to a simple programming interface. We were able to integrate the Uiml.net renderer as well as renderers for Java ME and DVB-MHP without much effort.

6. RELATED WORK

Model-based and automatic techniques have been frequently used for multi-platform user interface design [9]. This approach requires designers to define a high-level specification of the user interface which is then used to automatically produce an appropriate user interface for each target platform. Two examples of tools that rely on this technique are MOBI-D [19] and Dygimes [4]. One of the major drawbacks of this type of tools is that the design process is not intuitive for designers. As discussed in the previous section, GUMMY does not exhibit this problem.

Other tools that try to facilitate the design of multi-platform user interfaces have traditionally focused on low-fidelity or medium-fidelity prototypes. Notable examples include Damask [15] and SketchiXML [5]. Damask lets designers sketch a user interface for one device and indicate the design patterns the interface uses. From this initial design Damask will then automatically generate the other device-specific user interfaces. Although GUMMY and Damask share many concepts (e.g. building an abstract model in the background, generating initial designs which can be refined later, etc.), it is possible to conclude that they serve different purposes. While Damask is mainly targeted towards prototyping, the designs that are created with GUMMY can be directly used as the final user interface and coupled to existing application logic [17]. Recently, Damask was extended with the concept of *layers* for managing consistency between designs for different computing platforms [16]. The motivation for this was a survey among designers that identified consistency as one of the major burdens for cross-device user interface de-

sign. It would be interesting to examine if layers could be used within Gummy to propagate changes between designs for different platforms.

SketchiXML [5] creates an abstract user interface specification from a user interface sketch that can then be deployed on multiple devices. However, the tool has a limited set of abstractions that designers can employ to design a user interface. It builds upon the UsiXML language to describe the abstract user interface which has a predefined set of abstract widgets. With Gummy, the set of abstractions is defined externally in a UIML vocabulary and thus can be changed at any time. SketchiXML has recently been extended to support the entire range of prototype fidelities [5]. After the user evaluation, most participants indicated a preference for medium- or high-fidelity prototyping. This reflects our belief: We feel that most designers prefer to have a concrete representation available during their design activities.

Collignon, Vanderdonckt and Calvary [3] describe an interesting visual tool to specify *plasticity domains* for user interfaces. A plasticity domain defines a range of contexts of use for which a user interface is valid (e.g. a mobile phone and a PDA). Their tool can embed several UIs corresponding to different platforms into one running application. If the context of use changes, the most appropriate of these UIs is automatically selected and used. Contrary to the present approach, this work does not facilitate the design of multi-platform user interfaces. Instead, they focus on defining possible transitions between user interfaces for different platforms and on exploiting these transitions at runtime. The different designs still have to be created by hand.

In the Gummy tool, a basic transformation engine is used that is able to generate an initial design for a new computing platform depending on the available screen size. It was not an aim of the present work to contribute to developments in this area but the transformation engine was implemented to prove the utility of the design approach presented in this paper. Similar but more sophisticated techniques for adapting user interfaces include Supple [11], splitting rules for graceful degradation [10] and Artistic Resizing [7]. Each of these techniques require other types of input to steer the adaptation and are solely used at runtime. It will be interesting to explore these and other transformation algorithms that take into account a more general notion of context than just the available screen size (e.g. different interaction techniques and input devices such as pinch zooming on a multi-touch display). Supple [11] looks promising as it already has basic support for adapting to interaction techniques (e.g. making user interface elements larger to ease interaction on a touch screen), and to the user's preferences and abilities [12]. Modelling input devices and interaction techniques [1, 8] might be useful to cope with the differences between computing platforms and to manage overall consistency.

7. DISCUSSION

This paper presented Gummy, a multi-platform GUI builder that allows designers to easily target new computing platforms without having to give up their current work practices. As designers work on the final user interface, Gummy builds up a corresponding UIML description. The additional abstraction provided by this UIML description allows Gummy to generate initial designs for new platforms based on existing user interfaces created for the same application. Gummy combines many of the advantages of existing multi-platform design tools with those of traditional GUI builders. This design approach lowers the skill barrier to multi-platform user interface design by allowing designers to easily intervene in the process and reuse their existing knowledge of single-platform design tools. Gummy's architecture is flexible enough to easily integrate a wide range of computing platforms and UIML renderers.

We feel that our work opens up interesting possibilities for further research. In particular, the process of empowering domain experts to design user interfaces with Gummy will be examined. Existing tools are not tailored toward non-technical domain experts, even though their input is very important during the design process and helps to shape the final user interface. To provide support for domain experts, the Gummy workspace should not only take into account the target computing platform but also the domain for which the user interface will be designed. UIML vocabularies allow us to do this [14].

At the moment, Gummy does not explicitly enforce consistency. Consistency is only ensured between an initial design and the existing designs it was generated from. Although this is sufficient in most cases, it does not scale well to widely varying computing platforms. Sometimes breaking consistency is desirable because of the specific nature of the target platform (e.g. excluding labels due to space constraints). In the future, Gummy should allow designers to control consistency between computing platforms at a high level of granularity (e.g. exclude labels for both PDAs and mobile phones, but include them for other platforms). As mentioned in the discussion on related work (Sect. 6), Damask's concept of layers [16] in a modified form might be a good way to realise this.

Gummy lacks a number of features that are crucial for its applicability to real-world problems. For one, only user interfaces with a single screen can be designed. Support for multiple dialogues would allow designers to create complex multi-platform user interfaces with the tool. The dialogue flow should be kept consistent when dialogues are split for certain platforms and merged for others. Furthermore, designers currently have no way of specifying how a user interface should behave when it is resized at runtime. The main challenge here is to integrate different platform-specific layout managers in a generic way.

We are aware of the limitations of our current rule-based transformation engine (Sect. 5). Since the focus is mainly on an intuitive multi-platform user interface design approach, our efforts will not concentrate on developing a more advanced transformation engine. Instead, an attempt will be made to extend the Gummy tool to enable the integration of multiple transformation engines. This would allow the use of existing, more sophisticated techniques such as Supple [11], Artistic Resizing [7] or graceful degradation [10]. Designers would then be able to try out several transformation engines to generate an initial design and pick the one they like best. As mentioned in Sect. 6, it is necessary to investigate transformation algorithms that take into account

more than just the available screen size. Interaction modelling approaches [1, 8] might be used to tackle this problem while still preserving the generality of our approach.

More information on GUMMY and an executable of the tool can be found at `http://research.edm.uhasselt.be/~gummy/`.

Acknowledgments

This paper would not have been what it is today without the help of the other researchers at EDM. We warmly thank everyone who helped us test the tool and provided useful insights during the writing of this paper. Part of the research at EDM is funded by ERDF (European Regional Development Fund) and the Flemish Government. The AMASS++ (Advanced Multimedia Alignment and Structured Summarization) project IWT 060051 is directly funded by the IWT (Flemish subsidy organization).

8. REFERENCES

[1] Renaud Blanch and Michel Beaudouin-Lafon. Programming rich interactions using the hierarchical state machine toolkit. In *Proceedings of AVI '06*, pages 51–58, New York, NY, USA, 2006. ACM.

[2] Luca Cardelli. Building user interfaces by direct manipulation. In *Proceedings of UIST '88*, pages 152–166, New York, NY, USA, 1988. ACM.

[3] Bernoît Collignon, Jean Vanderdonckt, and Gaëlle Calvary. An intelligent editor for multi-presentation user interfaces. In *Proceedings of SAC '08*, New York, NY, USA, 2008. ACM.

[4] Karin Coninx, Kris Luyten, Chris Vandervelpen, Jan Van den Bergh, and Bert Creemers. Dygimes: Dynamically generating interfaces for mobile computing devices and embedded systems. In *Mobile HCI*, volume 2795 of *Lecture Notes in Computer Science*, pages 256–270. Springer, 2003.

[5] Adrien Coyette, Suzanne Kieffer, and Jean Vanderdonckt. Multi-fidelity prototyping of user interfaces. In *Proceedings of INTERACT '07*, volume 4662 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 2007.

[6] Jr. Dan R. Olsen. Evaluating user interface systems research. In *Proceedings of UIST '07*, pages 251–258, New York, NY, USA, 2007. ACM.

[7] Pierre Dragicevic, Stéphane Chatty, David Thevenin, and Jean-Luc Vinot. Artistic resizing: a technique for rich scale-sensitive vector graphics. In *Proceedings of UIST '05*, pages 201–210, New York, NY, USA, 2005. ACM.

[8] Pierre Dragicevic and Jean-Daniel Fekete. Support for input adaptability in the icon toolkit. In *Proceedings of ICMI '04*, pages 212–219, New York, NY, USA, 2004. ACM.

[9] Jacob Eisenstein, Jean Vanderdonckt, and Angel Puerta. Applying model-based techniques to the development of uis for mobile computers. In *Proceedings of IUI '01*, pages 69–76, New York, NY, USA, 2001. ACM.

[10] Murielle Florins, Francisco Montero Simarro, Jean Vanderdonckt, and Benjamin Michotte. Splitting rules for graceful degradation of user interfaces. In *Proceedings of AVI '06*, pages 59–66, New York, NY, USA, 2006. ACM.

[11] Krzysztof Gajos and Daniel S. Weld. Supple: automatically generating user interfaces. In *Proceedings of IUI '04*, pages 93–100, New York, NY, USA, 2004. ACM.

[12] Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of UIST '07*, pages 231–240, New York, NY, USA, 2007. ACM.

[13] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[14] James Helms and Marc Abrams. Retrospective on ui description languages, based on eight years´ experience with the user interface markup language (uiml). *International Journal of Web Engineering and Technology (IJWET)*, 4(2), 2008. To appear.

[15] James Lin and James A. Landay. Damask: A Tool for Early-Stage Design and Prototyping of Multi-Device User Interfaces. In *Proceedings of DMS '02*, pages 573–580, 2002.

[16] James Lin and James Landay. Employing patterns and layers for early-stage design and prototyping of cross-device user interfaces. In *Proceedings of CHI '08*, New York, NY, USA, 2008. ACM. To appear.

[17] Kris Luyten, Kristof Thys, Jo Vermeulen, and Karin Coninx. A generic approach for multi-device user interface rendering with uiml. In *Computer-Aided Design Of User Interfaces V*, pages 175–182. Springer Netherlands, 2007.

[18] Brad Myers, Scott E. Hudson, and Randy Pausch. Past, present, and future of user interface software tools. *ACM Trans. Comput.-Hum. Interact.*, 7(1):3–28, 2000.

[19] Angel Puerta and Jacob Eisenstein. Towards a general computational framework for model-based interface development systems. In *Proceedings of IUI '99*, pages 171–178, New York, NY, USA, 1999. ACM.

[20] Mark Weiser. The computer for the 21st century. *Scientific American*, 265(3):66–75, September 1991.

Interactive Querying and Retrieval

KMVQL: a Visual Query Interface Based on Karnaugh Map

Jiwen Huo
David R. Cheriton School of Computer Science
University of Waterloo
jhuo@cs.uwaterloo.ca

ABSTRACT

Extracting information from data is an interactive process. Visualization plays an important role, particularly during data inspection. Querying is also important, allowing the user to isolate promising portions of the data. As a result, data exploration environments normally include both, integrating them tightly.

This paper presents KMVQL, the Karnaugh map based visual query language. It has been designed to support the interactive exploration of multidimensional datasets. KMVQL uses Karnaugh map as the visual representation for Boolean queries. It provides a visual query interface to help users formulate arbitrarily complex Boolean queries by direct manipulation operations.

With KMVQL, users do not have to worry about the logic operators any more, which makes Boolean query specification much easier. The Karnaugh maps also function as visualization spreadsheets that provide seamless integration of queries with their results, which is helpful for users to better understand the data and refine their queries efficiently.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

query formulation, visual query, visualization, Karnuagh Map, direct manipulation

1. INTRODUCTION

The massive volume and the huge variety of large knowledge bases make information exploration and analysis difficult. An important activity is data filtering and selection, which breaks up the large data set into meaningful, more manageable subsets. A query specified by a user defines how to partition the data and which data parts are required.

Most commonly data set queries are built from simple terms, combined using Boolean operators, which have convenient formal properties. Unfortunately, users have difficulty using Boolean logic [10, 9]. Some query interfaces try to alleviate this difficulty by expressing queries using a visual query language, which amounts to a visualization method for queries.

Visualization also plays an important role in data analysis, supporting both inspection and testing. As a result, interfaces for data exploration environments normally include both data visualization and visual query, integrating them as tightly as possible [11] [7].

This paper presents KMVQL, the Karnaugh map based visual query language. KMVQL has been designed to support the interactive exploration of multidimensional datasets. It incorporates dynamic query techniques and provides a visual query interface that allows users to formulate arbitrary Boolean queries by direct manipulation methods. It relieves users from the task of specifying Boolean logic operators so that query specification is easier.

KMVQL uses Karnaugh map(K-Map) [5] as the visual representation of Boolean queries. The tabular representation of K-Maps utilizes the human brain's pattern-matching capability for query comprehension, which makes KMVQL intuitive to use. The K-Maps in KMVQL also function as visualization spreadsheets [12] that provide seamless integration of queries with their results. The tight coupling of query and query result visualization makes it easier for users to analyze the results and refine their queries.

The paper is organized as the following: section 2 reviews related research in query interface design, focusing on representations of Boolean queries and visualization of query results; section 3 presents the software domain model based on which KMVQL was designed; section 4 introduces two visual representations for Boolean queries and describes how they can be integrated with query result visualization; section 5 and 6 introduces KMVQL and shows how it can be used for data exploration; finally the paper concludes with a discussion of future work.

2. BACKGROUND

2.1 Boolean Query Specification

Traditional database query systems require users to provide database queries written in command languages based on Boolean logic. Such systems are powerful and expressive. Unfortunately, Boolean logic is difficult and error-prone for large sections of the population [10, 9].

Many techniques have been developed as alternatives to command languages. Network(or graph) representations emerge as popular visualizations for queries, with nodes and links representing components and their relationships. But the logic operators are still explicitly displayed in the diagrams, with which users are still facing the difficulty of using Boolean logic. Form-based query interfaces and dynamic query techniques [1] provide predetermined query structures, usually pure conjunctions, which make the inter-

AVI '08, 28-30 May , 2008, Napoli, Italy.

face easy to use at cost of inflexibility.

Visual query languages show Boolean relationships using visual features, enabling users' visual reasoning skills for understanding and specifying Boolean combinations. In Venn diagram based representations [9], each query term is associated with a ring or circle, with intersections of circles indicating conjunctions of terms. Flow diagram representations [13, 14] use sequential flows for conjunction, parallel flows for disjunction. Iconographic representations, such as InfoCrystal [2], visualize the minimal polynomials of Boolean queries as graphical icons. To specify a query, the user finds the graphical icons that represent her information need and selects them. User studies on such query interfaces [10, 9, 2] show that good interfaces eliminate the explicit specification of logical operators, rely on recognition, and maximize concreteness.

KMVQL allows users to specify Boolean queries by selecting cells in a Karnaugh map, eliminating the necessity of using logic operators. Compared with similar visual query representations, such as Venn diagram, or InfoCrystal, the regular layout of Karnaugh map makes it easier to view, navigate, and interact with. Its tabular representation is familiar and intuitive to use. All these features facilitate the formulation and comprehension of Boolean queries.

2.2 Query Result Display

In traditional query interfaces, only data items that exactly meet the query are found and displayed, and those are often too few to give hints for refinement, or too many to browse efficiently. With large and unfamiliar data sets, users find it difficult to find the desired data. Researchers have endeavored to give better visual feedback for users' queries, helping them to understand the query results. The coupling of queries with results is especially important in exploratory search [15] [3].

For example, dynamic query interfaces provide immediate updates as the query develops. Dynamic query histograms, influence explorer [4], and similar techniques provide context information in the form of data distributions of each attribute. Influence explorer [4] provides additive encodings for sensitivity information [15], in which color coding is applied to indicate how many limits are satisfied. VisDB [8] displays tuples that do not satisfy the query, indicating their "distance" from the query. These approaches help users avoid missing important data points that fall just outside the selected query parameters.

Visualization techniques have also been applied to make the connection between the query terms and result data items explicit. For example, in TileBars [16], a graphical bar beside the title of each retrieved document shows the degree of match for each term. In InfoCrystal [2], and VQuery [9], the number of data items matching each query term is presented. Flow diagrams [13, 14] show quantity information of data flowing through the filters. In Tableau(Polaris) [11, 17] and Magic Lens Filters [6], data subsets are visualized in the context of the query terms they satisfy.

The KMVQL system presented in this paper provides seamless integration of queries with their results. It provides context information of data distribution along query terms so that users can better analyze the results, which is helpful for users to understand their queries and refine them efficiently.

3. THE SOFTWARE DOMAIN MODEL

This section introduces a software domain model based on which KMVQL was designed. It describes the basic components and behaviors of visual query interfaces. The components include data, queries, query results, and the visualizations of them. To be general, the granularity of the modules and their responsibilities is coarse, as shown in Figure 1.

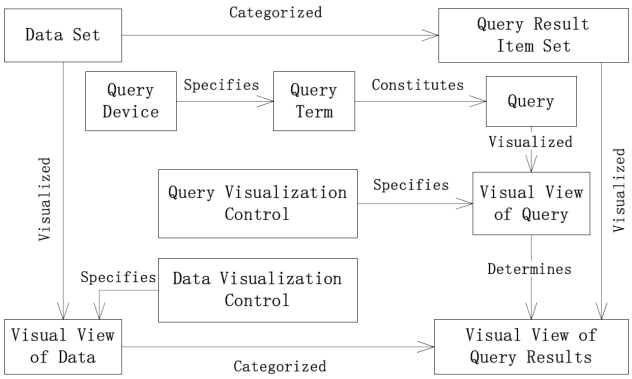


Figure 1: Software Domain Model

Deciding the responsibilities of the modules is a critical step in the design process. The following contents briefly described the responsibilities of some of the important modules.

- Data set module is responsible for data access, data transformation, data manipulation and meta-data management;
- The query device modules are user interface widgets provided for users to specify restrictions on data, which are used as query terms;
- The visual view of query module visualizes Boolean query structure and provide interaction mechanisms for users to specify query structures;
- The visual view of query results provide visualization and interaction mechanisms for users to read the data and operate on it;
- The data visualization control module provides definitions for data visualizations. It may also provide interfaces allowing users specifying the visual mappings. There is a spectrum of choices, with a tradeoff between simplicity and expressibility;
- The query visualization control module provides definitions for visually representing queries;
- The query term modules manage Boolean query expressions that specifies constrains on data attributes.

The software model can be used as a reference for designing practical visual query interfaces. Based on it, KMVQL has been implemented.

4. BOOLEAN QUERY VISUALIZATION

This paper presents two visual representations for Boolean queries that are used in KMVQL. They are visualizations of truth table, which is a mathematical table used to compute the functional values of Boolean logical expressions.

4.1 Truth Table

A Boolean query is often expressed as a logical expression composed of query terms connected by the Boolean operators, or recursively composed expressions. The query terms, $T_1, T_2, \ldots, T_N$, are combined using Boolean operators: AND, denoted by T_1T_2, OR, denoted by $T_1 + T_2$, and NOT, denoted by $\overline{T_1}$.

A truth table is based on a canonical form of Boolean expressions: any Boolean expression using N terms can be written as the disjunction of 2^N query fragments in which all terms are written either negated or not. These query fragments are known as minimal polynomials, and it is elementary to show that any Boolean expression can be written as a disjunction of them. Thus, for example:

$T_1T_2 + T_1T_3 + T_2T_3 = T_1T_2T_3 + \overline{T_1}T_2T_3 + T_1\overline{T_2}T_3 + T_1T_2\overline{T_3}$

Only a subset of the $8 = 2^3$ fragments occur in this expression, and formally we could uniquely define each expression by listing the minimal polynomials that appear in it. Figure 2 shows the truth-table representation for this example.

	T_1	T_2	T_3	isTrue
$\overline{T_1}\overline{T_2}\overline{T_3}$	0	0	0	0
$T_1\overline{T_2}\overline{T_3}$	1	0	0	0
$\overline{T_1}T_2\overline{T_3}$	0	1	0	0
$\overline{T_1}\overline{T_2}T_3$	0	0	1	0
$T_1T_2\overline{T_3}$	1	1	0	1
$T_1\overline{T_2}T_3$	1	0	1	1
$\overline{T_1}T_2T_3$	0	1	1	1
$T_1T_2T_3$	1	1	1	1

$T_1T_2 + T_1T_3 + T_2T_3 =$
$T_1T_2T_3 + \overline{T_1}T_2T_3 +$
$T_1\overline{T_2}T_3 + T_1T_2\overline{T_3}$

Figure 2: Example of a Truth Table

Each unique query term creates a column in the truth table. All the attributes are Boolean values, each of which is "true" or "false" (here we use 1 to represent the "true" value, use 0 for "false") depending on whether the corresponding term is negated or not.

Each possible minimal polynomial forms a row in it. There is an extra column used for indicating whether a minimal polynomial occurs in the expression or not. "true" if the minimal polynomial is included in the expression, "false" otherwise. Any Boolean logical combination of the query terms can be represented using a truth table.

4.2 Iconic Representation for Boolean Queries

The truth table construction puts query in the same formal structures as a multidimensional data set. Therefore, techniques for visualizing multi-dimensional data can be applied to visualize queries.

Figure 3 presents an example of using graphical icons to visualizing a truth table which has 3 query terms. Each query fragment is represented as an icon with colored petals. The presence of a petal is determined by the value of the related attribute. If the *isTrue* value of a query fragment equals to 1, then the icon is surrounded by a blue circle. For example, if a query fragment is $< 1, 1, 0, 1 >$, then its associated icon has a red petal, a green petal, and is surrounded by a blue circle. In this example, the query is $T_1T_2 + T_1T_3 + T_2T_3$. It is a disjunction of four query fragments: $T_1T_2\overline{T_3}$, $T_1\overline{T_2}T_3$, $\overline{T_1}T_2T_3$, and $T_1T_2T_3$. Therefore, only the icons associated with the four query fragments are surrounded by blue circles.

4.3 Karnaugh Map

A Karnaugh map [5] is a 2-dimensional tabular representation of a truth-table that facilitates management of Boolean algebraic expressions. The rows and columns of the map are ordered according to the principles of Gray code, which makes it unique in that only one variable changes value between squares.

	T_1	T_2	T_3	isTrue
$\overline{T_1}\overline{T_2}\overline{T_3}$	0	0	0	0
$T_1\overline{T_2}\overline{T_3}$	1	0	0	0
$\overline{T_1}T_2\overline{T_3}$	0	1	0	0
$\overline{T_1}\overline{T_2}T_3$	0	0	1	0
$T_1T_2\overline{T_3}$	1	1	0	1
$T_1\overline{T_2}T_3$	1	0	1	1
$\overline{T_1}T_2T_3$	0	1	1	1
$T_1T_2T_3$	1	1	1	1

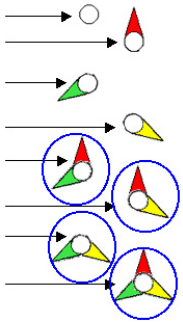


Figure 3: Iconic representation for Boolean query

	$T_2 = 0$		$T_2 = 1$	
	$T_3 = 0$	$T_3 = 1$		$T_3 = 0$
$T_1 = 0$	$\overline{T_1}\overline{T_2}\overline{T_3}$	$\overline{T_1}\overline{T_2}T_3$	$\overline{T_1}T_2T_3$	$\overline{T_1}T_2\overline{T_3}$
$T_1 = 1$	$T_1\overline{T_2}\overline{T_3}$	$T_1\overline{T_2}T_3$	$T_1T_2T_3$	$T_1T_2\overline{T_3}$

Figure 4: A Karnaugh map with 3 query terms

Karnaugh maps make use of the human brain's excellent pattern-matching capability for query comprehension. Its tabular representation is familiar and intuitive to use. Each query fragment is visualized as a cell in the map.

The regular tabular layout of the cells makes the Karnaugh map easy to view, navigate, and interact with. It can be used for specifying Boolean queries: a user need only to find the cells that meet her information requirement and select them. Thus, specifying Boolean queries is transformed into visual search. Users need not worry about the boolean operators.

4.4 Visualizing Query Results

The iconic and Karnaugh map representations can also be seamlessly integrated with query result visualization. Figure 5 shows an example of visualizing the data set based on the query.

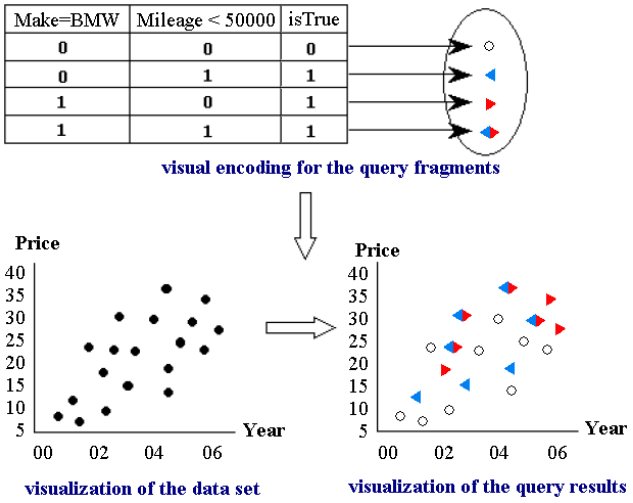


Figure 5: Example: Visualizing Query Results

In the example, the Boolean query is **(Make = BMW) OR (Mileage < 50000)**, which is composed of two query terms. Icons are colored by query fragments, an addition to the original data visualization. In this example, the icons are composed of two sets of attributes: the original visual attributes (x-axis and y-axis positions), and the visual coding for query fragments. In this way, the visualization provides feedback about the data distribution by query terms so that users can better understand the result.

Figure 6 uses a Karnaugh map to visualize the query. Data items associated with a specific query fragment are displayed in the corresponding cell. Each cell is a small visualization.

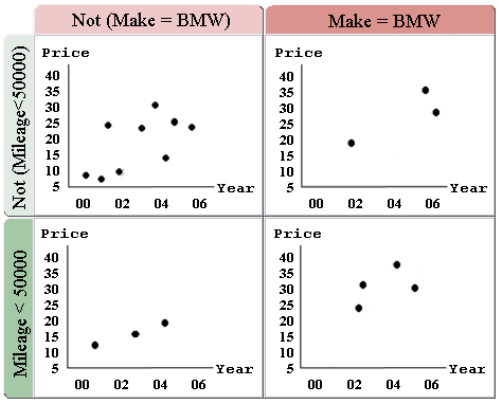


Figure 6: Visualizing Query Results with Karnaugh Map

In the above examples, query results are visualized, showing simultaneously both data set and query structure. This method helps users make sense of the data and facilitates query refinement and modification.

5. OVERVIEW OF KMVQL

KMVQL is designed based on the software domain model presented in Section 3. Its interface is mainly composed of three parts: a visual view that provides graphical presentations of the data, a Karnaugh Map(K-Map), and a set of query device widgets for data value selection. This section introduces these components and how they work together to help users formulating Boolean queries and exploring the data.

5.1 Visualization of Query and Query Results

Two visual query representations are supported in KMVQL: K-Map and iconic representation. Section 4 has described how to use them for Boolean query specification and query result visualization.

Using K-Map to visualize query results makes it possible to incorporate various data visualization mechanisms. Therefore, it works as a visualization spreadsheet [12], which allows users make comparisons between cells. Users can specify query structures by directly selecting the cells related with their information need, which is easy and intuitive.

On the other hand, K-Map splits the visualization into discrete cells. The context information for each cell is not explicit. Therefore, iconic representations are used so that query results are visualized in a single visual view. Corresponding to the two visual representation types, the visual view of query results can be in two modes: K-Map mode, and iconic mode, as shown in Figure 7 and 8.

In iconic mode, the iconic representation is not convenient for query specification. Therefore, a separate K-Map is provided for

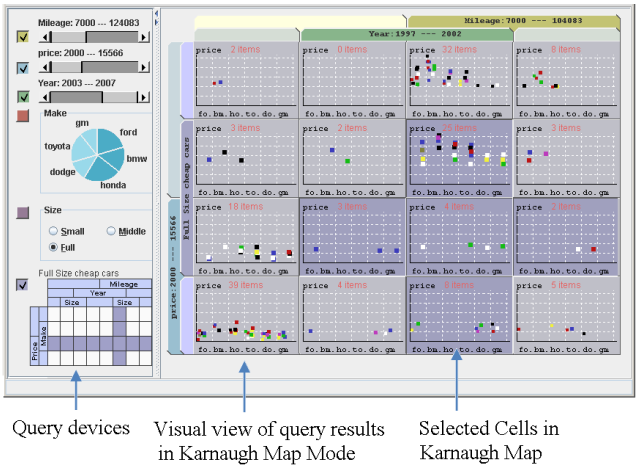


Figure 7: KMVQL User Interface in K-Map Mode

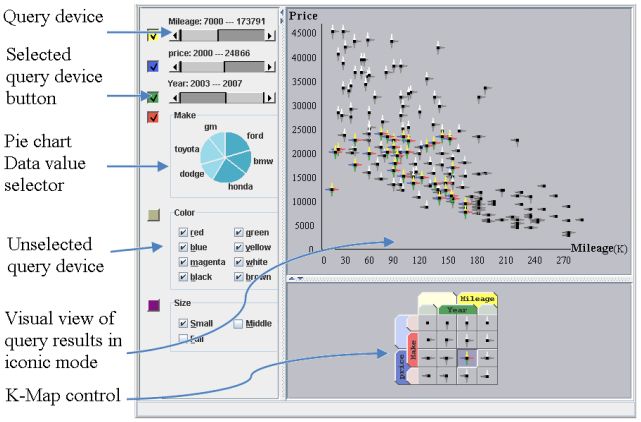


Figure 8: KMVQL User Interface in Iconic Mode

Boolean query structure specification. The K-Map also provides a legend for the visual codings of the query fragments, which can be modified by users. The icons with colored petals presented in Figure 3 is a typical example of iconic coding for query fragments. Such visual coding is complex which might result in clutter when the number of query terms or data items is large. Other simpler coding methods are also supported in KMVQL. For example, visual query fragments can simply differ in color or size. Such visual codings have already been adopted by some systems such as Attribute explorer [4]. On the other hand, it is hard to compare different query fragments with the simple encodings. KMVQL provides the flexibility allowing users to define what visual encodings to use based on their own need.

5.2 Data Visualization

KMVQL provides several predetermined visualization structures: x-y-plot, bar-chart, pie-chart, and parallel-coordinates. Each visualization structure is constituted by a set of visual features. For example, an x-y-plot is constituted by X-axis position, Y-axis position, icon color, icon size, and icon shape. A parallel-coordinate is constituted by a set of axis and color of icons. User can define the mapping relationship between the data attributes and visual features with a tool provided in KMVQL.

5.3 Query Device

In KMVQL, each query device is composed of two parts: a data value selector and an associated button displayed in front of it (see Figure 7 and 8). Simple data value selectors are dynamic query widgets, such as rangeslider, checkbox, radiobutton, combobox, and pie-chart.

Normally, rangeslider are used for ordinal and quantitative values. Checkboxes, radiobuttons, comboboxes and pie-chart are used for discrete values. Query devices in KMVQL are adjustable. Users can specify what type of widget is to be used for a data attribute. KMVQL also allows users to dynamically remove unnecessary query devices from the interface, or add new ones when needed. This feature solves the problem that the screen is occupied by a lot of useless widgets, which is a common problem in many query interfaces.

Interacting with a data value selector generates a Boolean function which is used as a query term. If the associated button of a query device is selected, the query term generated by the query device is added into the K-Map and used for query definition. Releasing an already selected button removes the associated query term from the K-Map.

5.4 Information Seeking Node

The information seeking process consists of a series of interconnected but diverse searches, which can be described as a sequence of query and visualization operations on data. To describe a search status in the process, we introduce the term **information seeking node**, which is defined as a tuple $< D, Q, VD, VQ >$.

Among them, D denotes the current working data set. Q describes the query being specified, which includes the status information of the query devices and the K-Map. VD describes how the data is visualized, and VQ describes the visual representation of the query.

At any state of information seeking, user can save current state as an information seeking node for later review, modification, or for new query creation. When a user load an information seeking node, the working data set and the interface components (including the query devices displayed on the screen, the K-Map, and the visual view of query results) are deployed based on the information stored in the node. This feature allows users to reuse previous query results.

5.5 Formulate Boolean Queries in KMVQL

The scenarios of formulating a query in KMVQL are:

1. load data from a data source. The data is visualized and displayed. KMVQL creates query devices based on the properties of data attributes.

2. specify query terms by interacting with the query devices. The associated button of the active data value selector is automatically selected. Selecting a button adds a new query term to the K-Map; releasing a button removes the related query term.

3. specify query structure by selecting or unselecting cells in the K-Map. The default setting of the K-Map is to automatically select the cell representing the conjunction of the query terms. If a user wants to specify pure conjunction queries, she does not need to operate on the K-Map.

4. double click the visual view of query results, the set of data items that exactly fulfill the query is used as the new working data set, on which users can make further exploration by repeating step 2 and 3. User can also save the query results to use them later, and open a previous information seeking node for analysis.

Upon any above operations, the visual view of query results updates its display immediately to give feedbacks. The number of query terms in the K-Map equals the number of selected query devices. As shown in Figure 7, there are 4 selected query devices, thus the query is composed of 4 query terms.

As shown by the examples in Figure 7 and 8, the K-Map tabs are displayed in different colors. The colors of the map tabs play two roles: (1) the positive examples of a query term can be easily differentiated from the negative ones; (2) the relationship between a map tab and a query device can be easily identified. Each query device is assigned a color, in which its associated button is displayed. Each map tab is associated with a query device. The same color coding is used for the K-Map tabs. By using the same color, the connection between the query devices and the K-Map tabs is explicitly revealed to users.

Of necessity, the query devices, the K-Map, and the visual view of query results are tightly coupled. The K-Map acts as a middle-ware joining the other components. In traditional dynamic query systems, no such middle-ware exists, the resulting query is limited to the conjunction of predetermined query devices. But using K-Map, arbitrary Boolean queries can be easily formulated.

5.6 Compound Query Based on Intermediate Results

With KMVQL, users are allowed to formulate compound queries, which is accomplished by using previous query results to specify data constraints for a new query. KMVQL supports three types of query devices:

1. widgets for simple value selection, such as rangeslider, radio buttons, checkboxes, etc.

2. small visual view of a data set, and

3. small Karnaugh map.

Query devices can be created based on previous information seeking nodes. User can specify the type of the query device to be either a small K-Map or a small visual view, which is created based on the information stored in the node.

KMVQL also allows users to directly select data from the visualization. For example, dragging out a rubber-band in x-y-plot, or brushing an axis in parallel-coordinates are common direct data selection mechanisms in visualization systems. In response to a selection operation, the selected items are highlighted. If user drag-and-drop the selection into the frame of query devices, a new query device is created and added into the interface. The new query device is a small visual view of the data set, with the selected items being highlighted.

The query devices are used in the same way for query structure specification, as described in Section 5. Therefore, users can reuse intermediate query results and specify Boolean combinations of them.

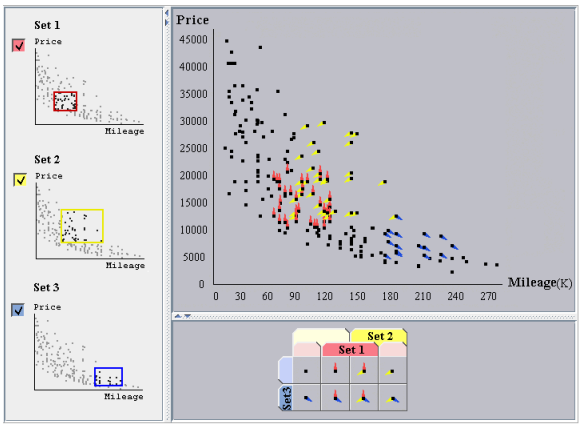


Figure 9: Create query devices based on direct visual item selections in KMVQL

6. EXAMPLE OF EXPLORING DATA WITH KMVQL

To describe the capabilities of KMVQL as an information exploration interface, we use an example. Suppose the user, Iris, was interested in buying a second-hand car. But she was not familiar with the car market, and she had no particular preference of the car. This example shows how KMVQL helps her in seeking for the desired one.

At the beginning of the exploration, Iris loaded into KMVQL a database of second-hand cars. When the data was loaded, KMVQL created a set of query devices based on the data type of the attributes. For example, the Makes of the cars are strings, which are nominal. KMVQL uses a combobox to list all the string values allowing users to select the preferred car Makes. But Iris was not satisfied with the automatically generated query device for this attribute. She chose pie-chart as the data value selector, which provides easier data selection mechanism as well as reveals the portions of the Makes.

Then Iris choose to visualize the data with a x-y-plot. With a form-filling dialog, she specified to map the Mileage attribute to the x-axis, and map the Price attribute to the y-axis. Then the data visualization is displayed on the screen. By operating on the query devices, Iris specified query terms. In the K-Map the cell representing the conjunction of the query terms was automatically selected.

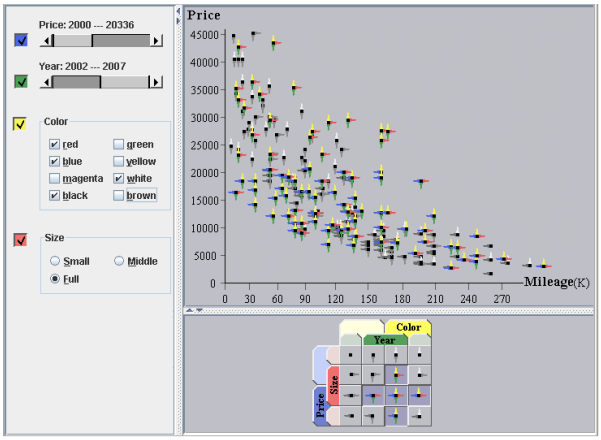


Figure 10: Screen shot of the query "meet at least 3 query criteria"

The data items associated with the selected cell were displayed in colored icons, while other items were displayed in grey scale. This makes the selected items stand out as well as providing context information.

When Iris analyzed the visual view she found that there were few items meet the pure conjunction. So she clicked other cells in the K-Map to broaden the query so that more data items were added as the results. Thus she would not miss promising candidates. Iris also kept interacting with the query devices to modify her query. Upon any operations, the visual view was updated immediately to to give her visual feedbacks so that she can understand the data more efficiently.

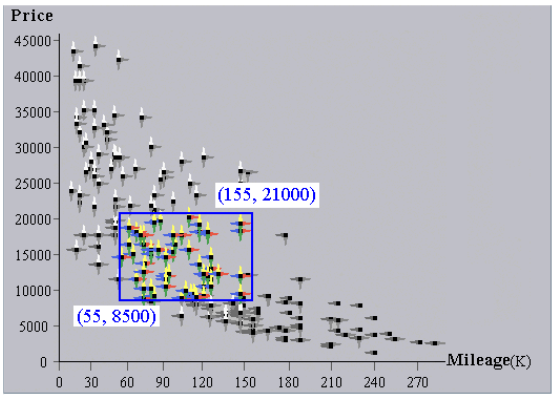
When she felt satisfied with the query and the results, she saved the current state as an information seeking node N_1 so that she could analyze or use it later. The query being specified was Q_1: "satisfy at least 3 query criteria among the four", as presented in Figure 10. Thus she named the node as "Meet at least 3 criteria".

Then Iris dragged out a rectangular box on the x-y-plot, the second query Q_2 was formed. She dragged-and-dropped the rectangular box to the frame of query devices. Thus a new query device was created and added into the window, showing the small visual view of the selection. Iris named the new query device as "Price-Mileage Selection".

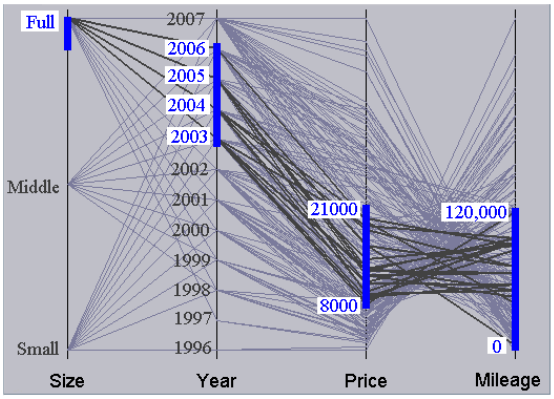
Iris was not sure whether the last two queries filtered out some important candidate cars. She wanted to get more information about the data set. So she reset the interface and analyzed the data set with a parallel-coordinates visualization. She brushed the axis to indicate the interested value ranges, then the third query Q_3 was formed. She saved the current state as a second information seeking node N_2 and named it as "FullSize-After 2003". (as we can see, although the user-defined names are not accurate, they allow users to remember and recognize the intermediate results.)

Then Iris reset the interface again using x-y-plot to visualize the data set. This time the Year attribute was mapped to the x-axis so that she could analyze the data from a different perspective. She added two new query devices into the interface. One shows a small parallel-coordinates for information seeking node N_2, another is a small K-Map for node N_1. To make the interface looks clean and simple, she removed unused widgets from it. She chose the K-Map mode for the interface and selected four cells in the K-Map to form a query. As shown in Figure 12, the query means "meet at least 2 queries among Q_1, Q_2, and Q_3".

Then Iris double clicked the visual view of query results to ana-



(A) Data selection on X-Y-Plot



(B) Data selection on Parallel-Coordinates

Figure 11: Direct data selection on the visual views

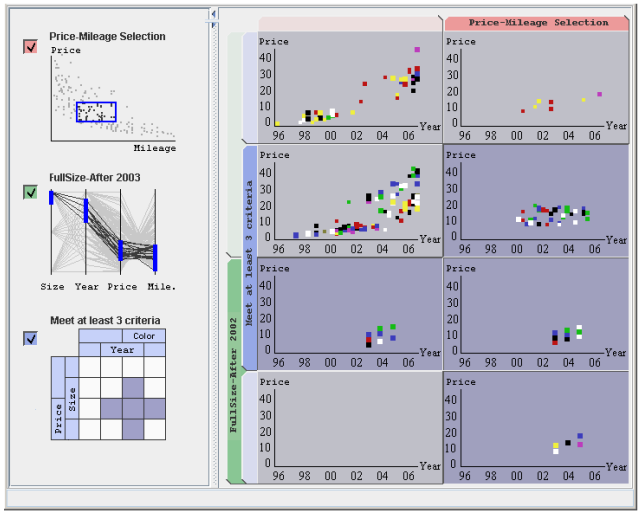


Figure 12: Screen shot of the Compound Query in KMVQL

lyze the data items that fulfill the query. The other items were filtered out so that she could focus her analysis on a relatively small data set. Iris saved the current interface status to a file so that when she explores the same data in the future, the interface will be automatically configured based on it. All the intermediate results will be restored. She could start from this point to continue her search until she found the desired car to purchase.

7. DISCUSSION

The kernel of KMVQL is to use K-Map for both Boolean query presentation and query specification. Thus it is natural to ask "is K-Map really easy to understand and easy to use?" To answer the question, we conducted an experiment comparing the comprehensibility of K-Map with textual representations of Boolean expressions.

From the experimental results, we found that for simple queries, such as queries with only one or two query terms, or queries only involve pure conjunction or pure disjunction, K-Map shows little advantage over textual Boolean expressions. But when the complexity of the Boolean query increases, K-Map increasingly outperforms textual expressions. This indicates that K-Map is promising for dealing with complex problems.

Since more complex interactions between users and data are the future of information exploration, and more complexity requires tools that support it. Thus, K-Map is sure to be important in the future.

However, the scalability of K-Map is problematic. When the number of query terms grows large, the number of cells in a K-Map grows exponentially. For example, the number of cells in a K-Map with 20 query terms is 2^{20}. It is hard to identify each individual cell, which makes query specification and comprehension difficult. Therefore, formulating complex queries that involve large number of query terms using a single K-Map is not efficient.

The mechanism of formulating compound queries based on intermediate results provides a solution for this problem. Intermediate query results can be reused. Boolean combinations of them can be specified. In fact, decomposing complex data searching tasks into stages is a natural approach of information exploration. KMVQL provides intuitive methods that support it.

8. CONCLUSIONS AND FUTURE WORK

This paper presents a new visual query interface KMVQL that supports interactive exploration of multidimensional datasets. Important features of KMVQL include:

- it adopts innovative visualizations for Boolean queries, the K-Map and iconic representations, which can be seamlessly integrated with the visualization of query results to help users better understand the data and accomplish query refinement and modification efficiently.
- it provides innovative methods for specifying Boolean queries. To specify a Boolean combination of query terms, users simply need to select the corresponding cells in the K-Map. They do not have to worry about the logic operators any more, which makes Boolean query specification much easier. User queries are no longer restricted to pure conjunctions of predetermined query devices;
- it provides flexible management of the query devices: new query devices can be dynamically added to the interface when needed; they can also be removed from the interface to make the screen less crowded;

- users can directly select data by interacting with the visual views. Boolean combinations of multiple selections can be specified easily by operating on the K-Map;
- important search status can be saved and reloaded for review, modification, or for new query creation. Intermediate query results can be reused.

These features are important to make the information exploration process easier and more efficient. In the future, we plan to implement new visual representations for Boolean queries in KMVQL to offer users more choices for query specification and data analysis. We also plan to design more informative and intuitive query devices, allowing users to specify visualizations and queries easily.

9. REFERENCES

[1] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11:70–77, 1994.

[2] A. Spoerri. *InfoCrystal: A Visual Tool For Information Retrieval*. PhD thesis, MIT, 1995.

[3] G. Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, 2006.

[4] L. R.Spence. The attribute explorer: information synthesis via exploration. *Interacting with Computers*, 11:137–146, 1998.

[5] M. Karnaugh. The map method for synthesis of combinational logic circuits. *Transactions AIEE, Communications and Electronics*, 72:593–599, 1953.

[6] K. Fishkin and M. C. Stone. Enhanced dynamic queries via movable filters. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–420. ACM Press/Addison-Wesley Publishing Co., 1995.

[7] W. Martin. Interactive visual analytics - put your smart brains in the driver's seat. 2007.

[8] D. A. Keim and H. Kriegel. Visdb: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 14:40–49, 1994.

[9] S. Jones. Graphical query specification and dynamic result previews for a digital library. In *In Proc. of UIST'98, ACM Symposium on User Interface Software and Technology*, pages 143–151, San Francisco, USA, 1998.

[10] S. Greene, S. Devlin, P. Cannata, and L. Gomez. No ifs ands or ors: A study of database querying. *International Journal of Man-Machine Studies*, 32:303–325, 1990.

[11] P. Hanrahan. *Visual Thinking for Business Intelligence - A White Paper of Tableau Software*. 2005.

[12] E. H. Chi, P. Barry, J. Riedl, and J. Konstan. A spreadsheet approach to information visualization. In *INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, page 17. IEEE Computer Society, 1997.

[13] T. Hansaki, B. Shizuki, K. Misue, and J. Tanaka. Findflow: visual interface for information search based on intermediate results. In *APVis '06: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation*, pages 147–152, 2006.

[14] D. Young and B. Shneiderman. A graphical filter/flow model for boolean queries: An implementation and experiment. *Journal of the American Society for Information Science*, 44(6):327–339, 1993.

[15] R. Spence. Sensitivity encoding to support information space navigation: a design guideline. *Information Visualization*, 1(2):120–129, 2002.

[16] M. A. Hearst. Tilebars: Visualization of term distribution information in full text information access. In *In Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems*, 1995.

[17] C. Stolte, D. Tang, and P. Hanrahan. Query, analysis, and visualization of hierarchically structured data using polaris. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 112–122. ACM Press, 2002.

Query-through-Drilldown
Data-Oriented Extensional Queries

Alan Dix
Computing Department, InfoLab21
Lancaster University
Lancaster, LA1 4WA
+44 1524 510 319
alan@hcibook.com

Damon Oram
Corporate Information Systems
Information Systems Services
Lancaster University
Lancaster, LA1 4WA
d.oram@lancaster.ac.uk

http://www.hcibook.com/alan/papers/avi2008-query-through-drilldown/

ABSTRACT
Traditional database query formulation is intensional: at the level of schemas, table and column names. Previous work has shown that filters can be created using a query paradigm focused on interaction with data tables. This paper presents a technique, Query-through-Drilldown, to enable join formulation in a data-oriented paradigm. Instead of formulating joins at the level of schemas, the user drills down through tables of data and the query is implicitly created based on the user's actions. Query-through-Drilldown has been applied to a large relational database, but similar techniques could be applied to semi-structured data or semantic web ontologies.

Categories and Subject Descriptors
H.2.3 [**Database Management**]: Languages – *query languages.* H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *query formulation.* H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces – *graphical user interfaces, interaction styles.*

General Terms
Design, Human Factors

Keywords
database query, data-oriented interaction, SQL, tabular interface, extensional query, data structure mining, query-by-browsing

1. INTRODUCTION
Traditional database query formulation is intensional, users are forced to formulate their queries in terms of schemas, table and column names. This often involves users in very abstract thinking, Boolean logic for defining filters and trying to understand the way that tables are linked together in joins – especially challenging for well-normalised databases. While languages and tools for this may be a powerful for experts, less experienced users may find them unnatural. Indeed the most successful end-user interaction techniques: web browsing and spreadsheets both keep the user focused on the data itself not meta-level descriptions of the data.

This paper takes the position that for many users a more extensional paradigm based on interacting with data is more easily understood.

Previous work on Query-by-Browsing has shown that it is possible to create filters using a query paradigm focused on data; users interact with and extensional view of a query and a query is inferred through machine learning. This paper presents a technique, Query-through-Drilldown, to enable join formulation in a data-oriented paradigm. Instead of formulating joins at the level of schema, the user drills down through tables of data and the query is implicitly created based on the user's actions.

In the next section, the paper begins by discussing the concept of extensional/data-oriented access. As Query-by-Browsing [5] was the initial inspiration for this work, we describe this in detail and analyse some of the generic issues it highlights. In particular QbB enables the creation of filters by simply allowing the user to select desired rows from a table of data. However, QbB does not have any way for the user to create joins.

Section 3 presents Query-through-Drilldown (QtD), a tableau-based interaction that allows complex multi-table queries to be created without explicit joins. The technique depends on an entity-relationship structure, so we also describe techniques to automatically derive this. Section 4 presents our experiences in implementing and evaluating a prototype of QtD and section 5 compares QtD with other data-oriented forms of browsing including semantic web ontologies. Finally, we discuss planned future work and possible extensions to less structured data.

2. DATA-ORIENTED ACCESS
2.1 Intensional vs. extensional data access
In database semantics, following other areas such as logic, a distinction is drawn between intensional and extensional forms of description. The intensional form is the query in terms of the schema, in relational databases usually expressed in SQL whilst the extensional form is the collection of records.

AVI'08, 28-30 May , 2008, Napoli, Italy

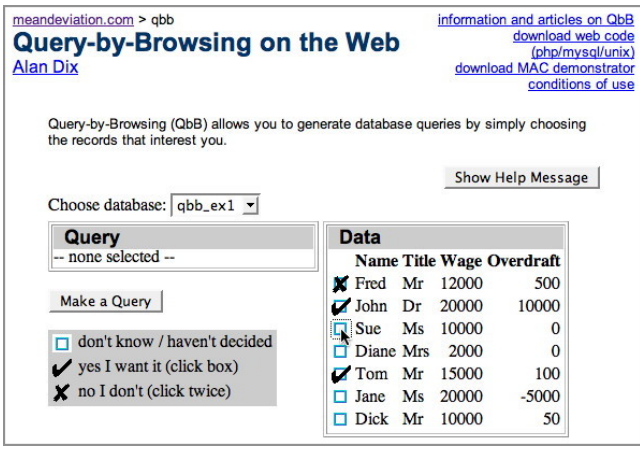


Figure 1. QbB (web interface) – user selects records.

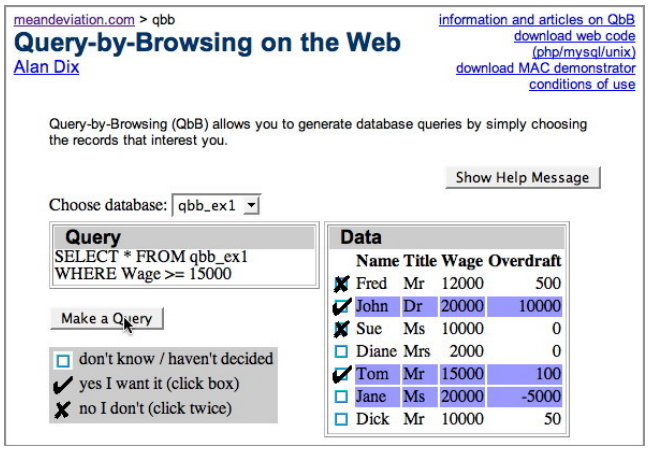


Figure 2. QbB generates SQL and highlights query results.

We see similar patterns in other forms of formally structured data, in particular for semantic web ontologies stored in RDF we have SPARQL queries (intension) and a corresponding graph/set of triples (extension) as an output [14]. The powerful thing about intensional descriptions is that they can be reapplied to new data to obtain precise results, however they are often only usable by experts. Indeed even early studies of database query mechanisms showed that those that incorporated some form of tabular interface outperformed purely textual interfaces such as SQL [7].

Even in information retrieval (IR) systems or web search there is often a Boolean query (intension) giving rise to a set of pages or documents (extension), although the distinction is less sharp. For simple search the distinctions become more problematic as the search terms used are themselves part of the content of the document, however search terms can be re-interpreted over a different collection, so have an intensional aspect – indeed part of the skill of a good web user is knowing which terms to use rather than which pages to visit. In web search and certain forms of bibliographic search, the focus is much more in skimming the data of the results to choose appropriate ones, rather than necessarily tuning the search terms to be precisely correct.

Web and hypertext browsing is perhaps more complex still as the 'schema', such that there is, is at best node + link. The user's focus here is almost solely on the content except in sophisticated systems with multiple link types. This is also true of many forms of graph or tree browsing, although in such case the content may be represented simply by a name or icon.

Similarity-based or recommender systems are also more data oriented, for example, Amazon recommendations are specific books, not specifications of interesting books. Similarly, the Scatter-Gather Browser [11] clusters documents, but presents the clusters in terms of generated summaries – while these are not instances, the summaries are focused on the data content.

In general intensional descriptions are more precise and generalisable, but correspondingly more complex and hard to understand. In contrast extensional descriptions are simpler and more comprehensible, but cannot be easily generalised and hard to be sure of unless checked exhaustively.

The challenge is to use both effectively where they are strong.

2.2 Query-by-Browsing

Query-by-Browsing precisely addresses this issue by effectively turning the traditional query processing pipe on its head, starting with an extensional description and generating an intensional description from it.

QbB was first described in a concept paper in 1992 and later implemented [4,5]. However it is also available as a web demo and the screenshots are taken from that[1].

Figure 1 shows the first stage of use. The user has selected a number of records that are either wanted (ticks ✓) or not wanted (crosses ✗). In this initial stage the user's focus is entirely on the list of records; that is extensional; all the user is doing is selecting positive and negative examples.

After a period the user clicks "Make a Query" and the system generates an SQL query (Figure 2):

```
SELECT * FROM qbb_ex1 WHERE Wage > = 1500
```

In the initial paper this step was described as occurring when the system had sufficient confidence in its inferred query, but in all the implemented systems this is at the user's request.

Looking in detail at Figure 2, we can see that there is both the SQL query in the left hand area and also highlighted items in the listing on the right. The highlighted items are those that would be retuned by the SQL query. That is, the result is *both* intensional (the SQL) and extensional (the highlighted items).

The QbB papers emphasise the importance of this dual representation. Whilst a user may find it hard to produce syntactically correct SQL they may be able to recognise whether it is correct. For more complex Boolean queries the dual representation may make it easier for a user to make sense of the connective – for example the confusing difference between 'and' as used in Boolean logic and its everyday use.

The highlighted records (extensional output) make it easy for the user to verify the query, selecting the appropriate records from

[1] "Query-by-Browsing on the Web". accessed 19 Dec 2007. http://www.meandeviation.com/qbb/qbb.php

those that can be seen. However, the SQL query itself (intensional output), allows the user to verify that the query will also apply correctly to unseen records. This would be important if, for example, the selected records were to be updated in some way … perhaps awarding a pay increase!

QbB uses machine-learning to create the query. The algorithm in the original implementation and the web interface is a variant of Quinlan's ID3 [15], but alternative algorithms are also described [6]. The algorithm used in the extant implementations is guaranteed to give a consistent result – that is the records selected by the query will include all the positive examples and none of the negative ones. However, there may be several queries that are consistent with a given set of positive and negative examples, so while the algorithm is consistent it may not accord with the intention of the user. The user may detect this either because the highlighted rows are not as expected or because the query does not seem right (e.g. the query says "`Wage>=15000`", but the user knows that the key value is really a tax threshold of 14250).

If the user is not satisfied with the inferred query, more positive and negative examples can be given. The highlighted records are again useful as any that are highlighted but not wanted are obvious candidates to be explicitly excluded and vice versa. The user then requests a fresh query and iterates until the returned query is satisfactory. The QbB papers also suggest that the user should be able to interact with the query – particularly easy if the query return format is a Relational Query by Example tableau [20]. That is, the user's input to the system could be a mixture of intensional and extensional elements. However, again this is not implemented in the extant systems.

2.3 Table-based interaction

As well as being data-oriented, QbB is table based. In fact, the basic principles of data-oriented querying could be applied to non-tabular interfaces, it is no accident that in a system designed to be easy for non-experts tables were chosen as a reference implementation. Early studies comparing end-user performance with several database query facilities (including SQL and QBE) found that those facilities that included a tabular interface outperformed those based on a purely textural SQL interface [7]. While there are many times when various forms of graphical or network representations can be useful, tables, however, mundane, are at the heart of many data-intensive interfaces not least the ubiquitous spreadsheet.

While tables are often the output format of choice, they are also used as a central part of almost any information rich interactive environment including lists of messages in email clients, files in a directory or classes in an IDE. They have also been used in various forms of interactive visualization, notably for exploring patterns, correlations and trends in Table Lens [16]. Even Scatter-Gather [11] can be seen as partially table/list focused. In all of these cases it is the records actually selected by the user that are of interest (the extension) rather than any inferred query of criteria.

An interesting exception to this is Query-by-Excel [19]. Here the user uses a spreadsheet that includes extracts from several tables in the full database. Standard spreadsheet functions and formulae are used to link the data in the different table extracts. When the user is satisfied that the Excel spreadsheet it is uploaded into the Query-by-Excel system and the formulae on the extracts are generalised into a full database query or procedure.

Arguably this use of Excel (and indeed much ordinary use) is intensional as the user manipulates formulae. Indeed the power of spreadsheet use is the rapid and incremental turnaround between intensional formulae-focused steps and extensional reflection on the values in the cells.

Query-by-Excel is also particularly interesting as the system can use the formulae to create linkage between tables as well as calculation/selection within them. That is Query-by-Excel can create *joins*, one of the weaknesses of Query-by-Browsing.

3. QUERY-THROUGH-DRILLDOWN

3.1 The concept

Query-by-Browsing demonstrates how data-oriented, table-based interaction can be used to create generic queries. However, a clear weakness is its lack of provision for joins. This raises the question as to whether a similar philosophy of extensional querying can be used to create inter-table joins.

When tables are used in standard interactive applications they may be used to select multiple items for some operation (e.g. to which classifications an uploaded paper belongs) or to allow drilldown to further information. In the latter case this may result in the selected item being opened in its own window (as when an email or file opens) or some sort of hierarchical expansion in place or an adjoining frame.

We will effectively use a form of the last of these. However, typically when rows are 'expanded' the focus is on a *single row*, which already represents a single item. In contrast in a database table listing, it is some of the *columns* that represent foreign keys or shared values that form a point of potential connection to another table. We use columnar drilldown as a way for users to view particular information linked to a given set of records and in so doing implicitly create a join between those tables. For example, if there is a "City Name" column in a table, it could be connected either a table of tourist information about the city or local government. The user's *choice* of which of these to follow effectively creates a join.

3.2 Scenario

To see how this works we will work through a simple scenario.

We assume as a start point that a form of entity-relation structure already exists for the database. That is we know which columns in any table connect to which others. This might have been created by hand, or may be mined automatically. In section 3.4 we will describe methods to achieve the latter, but for now will simply assume it exists.

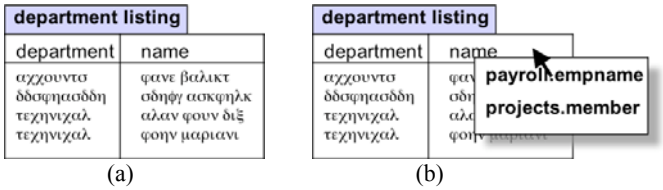


Figure 3. Selecting a column to drilldown through

Figure 3.a shows a listing a single table of department staff. In Figure 3.b the user selects the `name` column in order to find out more about the people. The system offers two options as there are two tables that have columns that are linked to the `name` field in `department listing`. In the figure these are named by the table name and column they are linked to, but part of a hand-crafted entity-relation structure might include more meaningful names for the relationships.

When the user selects one of the links from the `name` column the columns from the selected table are appended to the table. Figure 4 shows this in the case where the user has selected to expand the `payroll` record. In this case we have assumed there is a unique payroll item for each person so the table simply gets wider.

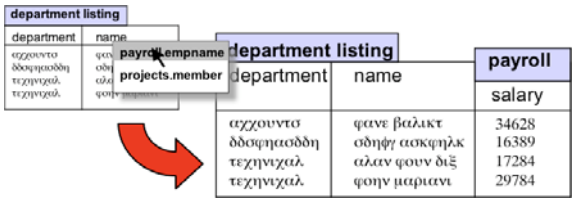


Figure 4. Selected column expands

Only one column is shown corresponding to the case if the `payroll` table had only two columns. In practice tables tend to have many columns and so the user may need to hide unwanted columns. To make this easier the system could default to show the most common columns from the table first (determined by handcrafted meta-data, automated analysis, or personal profile).

Figure 5 shows the SQL generated by this drill down. Unlike Query-by-Browsing we are not currently displaying this to the user in parallel to the tabular interface, but for experts this may be useful in order to generate the query, perhaps alongside a client or end user, and then copy the SQL for later use.

```
SELECT  d.department,
        d.name,
        p.salary
FROM    department d
INNER JOIN payroll p
ON d.empname = p.empname
```

Figure 5. Generated SQL

Several linked tables may be opened and Figure 6 shows the results if the user drills through the name to the `projects` table. Note it is shown at the same level as `payroll` to make clear it is a child (drilled from) the original `department listing` table. In this case, the projects are assumed to be in an m-n relationship with the names from the `department listing`. So in some cases there are several projects listed for each individual and in some cases none.

department listing		payroll	projects
department	name	salary	project
αχχουντσ	φανε βαλικτ	34628	φοβ το δο
δδσφηασδδη	σδηφγ ασκφηλκ	16389	προφεχτ
			φοβ το δο
τεχηνιχαλ.	αλαν φουν διξ	17284	φοβ το δο
τεχηνιχαλ.	φοην μαριανι	29784	– NONE –

Figure 6. Additional column for m–n relation.

Note that in the case of m–n relationships a `LEFT JOIN` is generated; that is all rows are retained in the `department listing` table even if there is no corresponding name in the `projects` table (people who are not members of any projects). This is because the user has started with the list of staff members in departments and so it makes sense not to lose these during drilldown. However, it would be equally odd to find extra names appear as it would with a `RIGHT JOIN`. Note that choosing the right kind of join is often confusing even for semi-experienced database users. However, the way in which the user constructs a query makes it obvious which kind of join is required.

Similar techniques can be used to drill down further through the linked tables, to add computed columns, filter and sort [2]. Figure 7 shows the end point of a series of interaction following on from Figure 6. Three computed columns have been added two connected with the `department listing` table and one with the `projects` (indicated by heights of the tabs). The overall table has also been reordered by project name. The relative heights of the table names help the user keep track of the relationship between the tables – the `payroll` table is only indirectly linked to the `projects` through the `department listing`. This is similar to the effect that would have happened if the user had started with the `projects`, drilled through to `department listing` and then to `payroll`.

projects	department listing		payroll			total cost
project	department	name	salary	nos proj = ...	proj cost = ...	= ...
φοβ το δο	αχχουντσ	φανε βαλικτ	34628	1	34628	51464
	δδσφηασδδη	σδηφγ ασκφηλκ	16389	2	8194	
	τεχηνιχαλ.	αλαν φουν διξ	17284	1	8642	
προφεχτ	δδσφηασδδη	σδηφγ ασκφηλκ	16389	2	8194	8194
– NONE –	τεχηνιχαλ.	φοην μαριανι	29784	0	#####	#####

Figure 7. Complex query: added columns and reordered (also see colour plate)

3.3 Relationship Model

The relational structure of a database can be thought of as a labelled graph where the vertices are tables and the labels on edges are relationships between foreign keys or shared values:

Schema = < Tables, Reln >

Reln $\subseteq$ Table × Table × SharedColumnFormula

The SharedColumnFormula will typically be a set of equalities between fields, but may be more complex as in an SQL `JOIN` clause. As noted earlier, for hand-crafted structures the relationships could be given meaningful names in each direction.

[2] A longer scenario with more of these features can be found at http://www.hcibook.com/alan/teaching/projects/workspace-drill-down.pdf

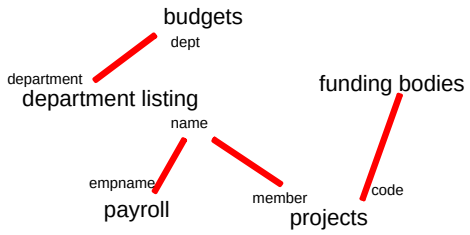


Figure 8. Relationship graph for database

Figure 8 shows an example database relationship structure corresponding to the example in Figures 37.

The query generated by Query-through-Drilldown is effectively a tree where the nodes are tables and the edges relations:

QbBquery = < Tree(Nodes,Edges), NodeMap, EdgeMap >

root ∈ Nodes
parent, child: Edges → Nodes
NodeMap: Nodes → Tables
EdgeMap: Edges → Reln

∀ e ∈ Edges: <t1,t2,c}> = EdgeMap(e)
p = NodeMap(parent(e)) ∧ c = NodeMap(child(e))
⇒ (t1 = p ∧ t2 = c) ∨ (t1 = c ∧ t2 = p)

Note that the mapping between Edges and Tables need not be injective as a table may be returned to during drill down. For example, from the configuration in figure 6 it would be possible to drill down through `projects` back to the `department listing`. This would give for each person in the department a list of the people who are in a project with them. Figure 9 shows a query tree for figure 6 (solid arrows) with the dashed arrow representing the additional drilldown back from `projects` to `department listing`.

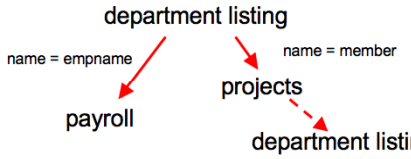


Figure 9. Query tree

3.4 Mining the Model

As noted the relationship model may be constructed by hand in which case meaningful names may be added for many of the relationships. However, for large databases or informal sources (such as a .csv file downloaded from the web) such hand annotation may be infeasible or impossible. Indeed even integrity constraints such as foreign keys are often only maintained implicitly in code and not in the database schema, so it seems likely that some form of automatic structure is needed.

Foreign keys are an obvious first step as they clearly establish a semantic connection between tables. These are most important (and happily most likely to be present) where the keys are simple ids as these are hardest to match implicitly.

Where there is no semantic information available or it is incomplete, the data itself can be used by matching the values in columns across different tables. If there is a high level of overlap between values in two columns then we can infer a relationship. However, this needs to take into account the density of values in their respective domains and especially for integer values. It is common to find id columns in tables consisting mainly of the initial N integers. Without a density check there would be many false positives as columns of ids and similar numbers of elements would overlap even where there is no real relationship. However, ignoring such accidental number range matches does mean that foreign id keys tend to be missed.

The example database that we have been using had a large number of such id fields and so techniques with more semantic information were required. Happily the database in question had large numbers of stored procedures. These procedures can be accessed via a straightforward SQL query (Figure 10).

```
SELECT text
FROM syscomments sc
INNER JOIN sysobjects so
ON sc.id = so.id
WHERE so.xtype = 'P'
```

Figure 10. SQL to access stored procedures

The queries in these formed a rich source to analyse (see figure 11). Wherever a JOIN is found (explicit or implicit in the form of "SELECT …WHERE table1, table2 …") we use the list of fields connecting the two to establish a relationship.

```
SELECT ss.student_id, sname = ss.surname +
', ' + ss.forename, … more fields …
   FROM std s
   INNER JOIN std_snapshot ss
       ON ss.student_id = s.student_id
   INNER JOIN std_address sa
       ON  ss.student_id = sa.student_id
       AND sa.address_type_lid = '000763'
   … 10 more lines  containing 3 more INNER JOINS …
   INNER JOIN org o
       ON ss.org_id = o.org_id
```

Figure 11. Typical SQL in stored procedures

This technique is not guaranteed to find every relationship; indeed in the database we were using with 300 tables it is likely that some potential relationships have never been traversed in previous use of the database. However, where stored procedures are heavily used, they are likely to find the most typical and useful relationships, including most of the important foreign keys.

A full SQL parser could be used for this extraction, but in fact a few regular expressions were sufficient to extract the majority of JOINS and their linkage columns. The exceptions were where aliases were used for table names (which could be captured by more complex regular expressions) and places where the JOIN includes database functions such as SUBSTRING() or INT().

Where stored procedures are not heavily used, the SQL for the queries may be scattered in the source code of many programs. However, databases often have some form of query logging, for example MySQL has a general query log where every query received is recorded [10]. However, compared to the use of stored procedures this is more computationally intensive as there will be many instances of essentially the same query with different parameterisations.

4. PROTOTYPE AND EXPERIENCE

4.1 Implementation

A prototype of Query-through-Drilldown has been created as a web-based interface using .NET framework on the server-side. It was originally hoped that the DataGrid control supplied in Visual Studio.NET web server could be extended. However, it was not possible to modify this to allow the step-down headings and so a custom solution was created using CSS and JavaScript.

The prototype has been developed and tested on our university student information database, which includes over 300 tables demonstrating scalability. However, because of obvious issues of privacy and security, the full and partial screenshots below are all taken from the Northwind, the example database, which forms part of the Microsoft SQL Server 2000.

Figure 12 shows a four table join constructed using the prototype. Note that even in the example database there are a substantial number of rows unlike the simulated screen shots shown earlier.

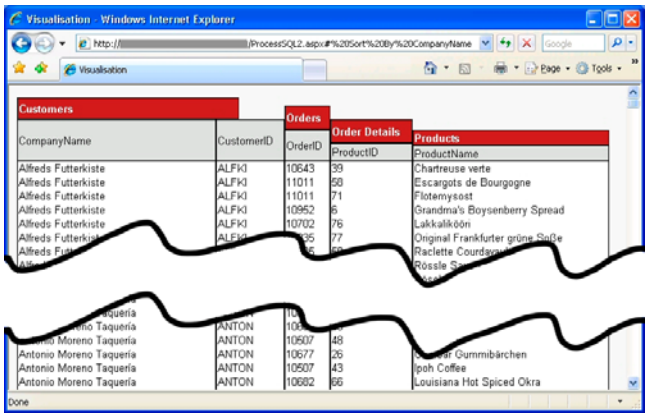


Figure 12. Prototype with four tables joined (also see colour plate)

While there are still many features we would like to add the prototype includes most of the key elements envisioned. For example, Figure 13 shows the query in figure 12 after further interaction adding a computed column and filtering the column based on the `CustomerID` column.

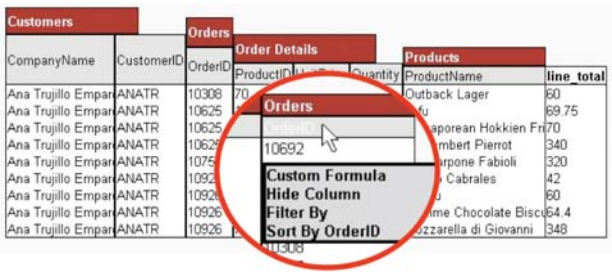


Figure 13. Prototype after filtering and computed column (also see colour plate)

4.2 Evaluation

Formative evaluation has been carried out with two groups of users one non-technical group and one technical group.

4.2.1 Non-technical users

Six non-technical users from an office environment took part in a more formal evaluation. They were initially contacted through their line-manager and then given some information ahead of the session by email describing the purpose of the study, duration and expectations on them. The experiment itself took place in their own premises, but with software installed by one of the authors. Due to security restrictions on the student database and to maintain privacy the Northwind database was used in these experiments. During the evaluation session itself the participants completed a pre-questionnaire to establish prior knowledge and then followed a number of tasks using a written think-aloud protocol (that is, rather than a verbal think-aloud, they were asked to keep notes while working and perform post-task reporting).

None of the non-technical user group had more than passing knowledge of SQL or SQL Server, although they had varying, but not deep, knowledge of desktop databases systems (particularly Access) and, once the term was explained, recognised Query by Example from its use in Access. All had extensive experience in use of spreadsheets.

Many of the user comments referred to fine details of the interface or requests for additional features such as the lack of short-cut keys, sorting on several columns, difficulty of finding certain menus, and confusing error messages. It is always a problem of such evaluations that many user comments relate to superficial interface 'bugs' rather than specific issues relating to the novel aspects. While the former are useful to improve a production system, it is the latter we really need at this formative stage. Happily, some of the comments were indicative of deeper issues.

One such issue was that column names were not regarded as 'user friendly' – they were simply the names of the columns in the database. In desktop databases there is usually provision for having column titles that are more meaningful to users than the column names found in the database schema. In a large commercial database such information is more often embedded in programs or reports. Where a report or UI generator has been used it may be possible to extract the column titles automatically, rather like the JOINS were mined from stored SQL queries. However, even if such column titles were found there may be several such names as the same database row may be presented differently to different kinds of user. This is not just an issue for Query-through-Drilldown, but any system that provides a universal user-interface to databases. In practice this requires semi-automatic user profiling or hand annotation, although this could be inferred if users are allowed to edit the column headings.

Another class of issues were due to the fact that even in the experiment we were using realistic data with substantial numbers of columns and records. When discussing figure 4, we we assumed that unwanted columns had been hidden from the projects table when it was added to the tableaux. However, even when the user only opens essential columns, the tableau grows in width and users complained about horizontal scrolling. This is a problem in any tabular layout, and certainly in more complex spreadsheets. Potentially the focus+context techniques of Table Lens would be useful here [16] or the grouping of columns as used in HyperGrid [8]. Vertical scrolling was also mentioned as a problem, which again might be helped by elision techniques. The

shear number of options created by the database size can also be daunting and may require more structured menus (see Fig 14).

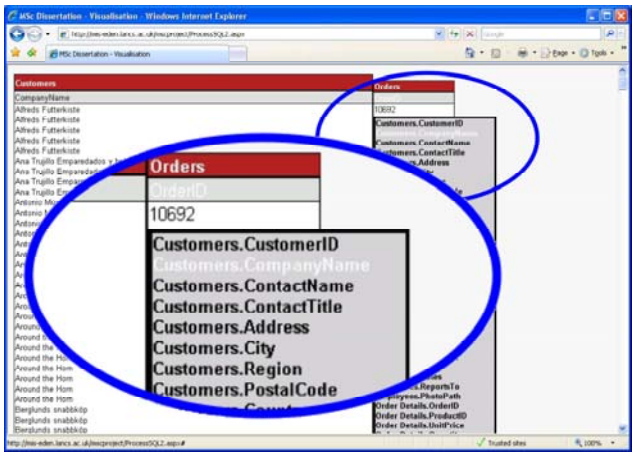


Figure 14. Long menus!

As noted the prototype has been developed using a very large database and, somewhat surprisingly, it has scaled without undue problems. However, users did note some delay on more complex refinements. This is because with a few interactions users were able to create complex queries with multiple joins and very large result sets. If this were submitted as an SQL query a delay a few seconds would seem reasonable, but in an interactive setting second or sub-second responses are expected. The current prototype is completely transaction based and stateless. However, if the interface were delivered as a stand-alone application or if the web interface used AJAX, then it would be possible to know which records the user was currently viewing. This would enable queries to be executed against the sub-selection of visible records substantially increasing the speed and in most cases making query processing proportional to the number of viewed records rather than the total table size. Again these response issues are ones that affect any highly interactive visualisation or query technique.

4.2.2 Technical users

Four technical users, two from an academic support environment and two from a commercial company were recruited for a form of focus group. These users all had high levels of database knowledge and of SQL in particular.

These users were treated very much in a co-designer/participative role. They were given access to the complete source code of the system before the session (in order to allow comments at a system architecture level) and were given a short presentation as to the purposes and vision of the system followed by hands-on time during the discussion session.

As with the non-technical users some of the discussion related to issues that, while important for practical deployment, were not directly related to the fundamental nature of the new technique; for example where configuration options should be stored, better use of the status line and window title, and browser-specific features. However, as these expert users had more knowledge of the purposes of the system they were also able to give more specific remarks about the system concept including the ER mining techniques. In particular they highlighted some of the limitations noted in section 3.4 regarding the regular expressions used to analyse stored procedures.

A major problem they noted, again common to most data visualisation systems, was how to connect to a server, choose a database and an initial table. Once started it becomes easier to navigate based on context, but how does one get started?

The group also noted that it would be useful for users to be able to bookmark states of the system. The ease of interaction meant it was easy to try out something, make a mistake and lose track of where one had been. Even implementing undo when very large SQL statements are being executed 'under the hood' is problematic, certainly requiring either caching or localisation techniques similar to those discussed to improve interactive performance in the previous section. However, explicit bookmarks would be useful too, not just during a single interactive session, but also to return to later. Sharing such useful queries would be one way to alleviate the 'blank screen' problem.

The knowledge of the technical users meant they could question the detailed semantics of Query-by-Browsing. In particular they were interested in the semantics of aggregation when columns were hidden. Interestingly the most intuitive semantics for a user interacting with the system is not the most 'obvious' SQL. Indeed some forms of sorting may require embedded `SELECT`s to create the 'right' answers for a user. The danger of this is that it may end up being confusing for the expert users.

The group suggested adding (the option of) an SQL window to show the actual query being constructed, as is found in Query-by-Browsing. This would fit more closely to QbB's paradigm of optimally combining intensional and extensional representations and also clarify expert users' questions about the semantics of more complex queries.

5. RELATED TECHNIQUES

We have already discussed several table-based interaction techniques in section 2.3. In addition, forms of drilldown or click-through have been used extensively for navigating data, from file browsers to the web, and in some places to aid query constructions.

Some uses of drill down operate at the level of instances of data, such as with links in web pages. Often, like web pages, these replace the current view so that the user 'moves' through the information space. However, rather as we have done with tables, this act of movement can be used to derive more generic queries. In the PESTO [2] system an OO database is browsed by drilling through properties of individual objects instances and each object (or object collection) is opened in a separate window connected to its parent in a graph. However, the path taken effectively forms a generic query and so if the parent object is changed then all the ancestors change accordingly. While Query-by-Browsing shows all the data in a tableau, PESTO focuses on the equivalent of a single row. Each has advantages and there would be arguments for being able to move back and forth between such representations.

Drilldown techniques are an obvious way to interact with hierarchical classifications and have been used extensively in mundane interfaces such as file browsers and also ones involving multi-faceted data or polyarchies [12, 3,17]. Drill-down has also

been used for database queries; one system, also called Query by Browsing [13], uses a file-system-like folder representation where each folder is effectively a table or class, and drilling down through a folder reveals not the rows of the table (instances), but other folders that are linked to the chosen one through the relational structure. While in some ways similar to our system this operates entirely at the schema (intensional) level.

There has been a long tradition of visual query languages [1], but most focus on schema-level constructions. An interesting example is a recent US patent which describes a table-oriented query formulation technique [9] using the relative positioning of tables to represent different forms of relationship, so, whilst displaying data, this is still schema focused.

Query-through-Drilldown uses the relational structure of a database and could easily be used on similar structures, notably semantic web ontologies. In that area m-Spaces [18] are perhaps most closely related as they also tabular layout of instances of classes to perform multi-faceted selections in related classes. However, in m-Spaces the equivalent of the JOIN, that is the specification of relationships between classes, is performed in a configuration step that requires more expertise than the selection interactions.

6. CONCLUSIONS

We have demonstrated how a data-oriented interaction paradigm can be used to create complex queries including joins. Whilst most comparable methods focus on the schema, that is extensional definitions of the query, the focus in Query-through-Drilldown is on the data, that is intensional.

While Query-through-Drilldown was envisaged as an end-user technique, from the evaluation it emerged that it would also be of value to experts in helping them rapidly create complex queries, but to do this would require a more explicit representation of the query, as in QbB.

Query-through-Drilldown has been described here and prototyped as a database interface. However, it was originally envisaged some years ago as a method to operate over other forms of tabular data as found ubiquitously in spreadsheets, word-processor documents and web pages, allowing integration of semi-structured data with fully structured databases. In the future we would like to create some form of adaptors to link such data with more structured databases and semantic web sources.

Given the inspiration for the extensional paradigm is Query-by-Browsing we also intend to integrate QbB filtering with Query-through-Drilldown giving an end-to-end data-oriented query platform.

7. ACKNOWLEDGMENTS

We are grateful to the subjects who gave their time during this project and a special mention for the baby who arrived in the middle. Also thanks to Andrew and Russell for rich discussions way back in 1998 when the first germs of this concept began.

8. REFERENCES

[1] Batini, C. Catarci, T. Costabile, M.F. Levialdi, S. 1991. Visual strategies for querying databases In Proc. of IEEE Workshop on Visual Languages (Kobe, Japan, 8-11 Oct 1991), 183-189.

[2] Carey, M., Haas, L., Maganty, V., and Williams. J. 1996. PESTO : an integrated query/browser for object databases. In Proc. of the Int. Conference on Very Large Databases (VLDB), (Mumbai, India, August 1996). 203-214

[3] Conklin, N., Prabhakar, S., and North, C. 2002. Multiple Foci Drill-Down through Tuple and Attribute Aggregation Polyarchies in Tabular Data. In Proc. of the IEEE Symposium on information Visualization (InfoVis'02) (October 28 - 29, 2002). IEEE Comp. Soc., 131–134

[4] Dix, A. 1992. Human issues in the use of pattern recognition techniques. In Neural Networks and Pattern Recognition in Human Computer Interaction Eds. R. Beale and J. Finlay. Ellis Horwood. 429-451.

[5] Dix, A. and Patrick, A. 1994. Query By Browsing. In Proc. of IDS'94: The 2nd International Workshop on User Interfaces to Databases, P. Sawyer, Ed. Springer Verlag. 236-248.

[6] Dix, A. 1998. Interactive Querying - locating and discovering information. Second Workshop on Information Retrieval and Human Computer Interaction, (Glasgow, 11th Sept. 1998). http://www.hcibook.com/alan/papers/IQ98/

[7] Greene, S. L., Gomez, L. M., and Devlin, S. J. (1986). A Cognitive Analysis of Database Query Production, In Proc. of the Human Factors Society, 9-13.

[8] Jetter, H.-C., Gerken, J., Konig, W., Grun, C. and Reiterer, H. (2005): HyperGrid - Accessing Complex Information Spaces. In: Proc. of the HCI05 Conference on People and Computers XIX 2005. 349-364.

[9] Liang, G. 2007. Method and System for Visual Query Construction and Representation. United States Patent 20070260582. Publication Date: 11/08/2007. http://www.freepatentsonline.com/20070260582.html

[10] MySQL 5.1 Reference Manual, Section 5.2.3. The General Query Log. Accessed 19th December 2007. http://dev.mysql.com/doc/refman/5.1/en/query-log.html

[11] Pirolli, P., Schank, P., Hearst, M., and Diehl, C. 1996. Scatter/gather browsing communicates the topic structure of a very large text collection. In Proc. CHI '96. ACM, New York, NY, 213-220.

[12] Pollitt, A. S., Ellis, G. P., and Smith, M. P. 1994. HIBROWSE for bibliographic database. J. Inf. Sci. 20, 6 (Nov. 1994), 413-426.

[13] Polyviou, S., Evripidou, P. and Samaras, G. 2004. Query by Browsing: A Visual Query Language Based on the Relational Model and the Desktop User Interface Paradigm. The 3rd Hellenic Symposium on Data Management, (HDMS04), (Athens, Greece, 28-29 June 2004).

[14] Prud'hommeaux, E. and Seaborne, A. (eds.) 2007. SPARQL Query Language for RDF. W3C Recommendation, 12 November 2007, http://www.w3.org/TR/2007/PR-rdf-sparql-query-20071112/. Latest version available at http://www.w3.org/TR/rdf-sparql-query/.

[15] Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106.

[16] Rao, R. and Card, S. K. 1994. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In Proc. CHI '94. ACM, New York, 318-322

[17] Robertson, G., Cameron, K., Czerwinski, M., and Robbins, D. 2002. Polyarchy visualization: visualizing multiple intersecting hierarchies. In Proc. CHI '02. ACM, New York, NY, 423-430.

[18] schraefel, m. Karam, M., and Zhao, S. 2003. mSpace: interaction design for user-determined, adaptable domain exploration in hypermedia. In Proc. AH2003 Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems,, 217–235

[19] Witkowski, A., Bellamkonda, S., Bozkaya, T., Naimat, A., Sheng, L., Subramanian, S., and Waingold, A. 2005. Query by Excel. In Proc. of the 31st international Conference on Very Large Data Bases (Trondheim, Norway, August 30 - September 02, 2005). Very Large Data Bases. VLDB Endowment, 1204-1215.

[20] Zloof, M. (1975). Query by example. Proc. AFIPS National Computer Conf. 44, AFIPS Press, New Jersey. 431-438.

Automatically Adapting Web Sites for Mobile Access through Logical Descriptions and Dynamic Analysis of Interaction Resources

Fabio Paternò, Carmen Santoro, Antonio Scorcia
ISTI-CNR
Via Moruzzi 1, 56124 Pisa, Italy
{fabio.paterno, carmen.santoro, antonio.scorcia}@isti.cnr.it

ABSTRACT
While several solutions for desktop user interface adaptation for mobile access have been proposed, there is still a lack of solutions able to automatically generate mobile versions taking semantic aspects into account. In this paper, we propose a general solution able to dynamically build logical descriptions of existing desktop Web site implementations, adapt the design to the target mobile device, and generate an implementation that preserves the original communications goals while taking into account the actual resources available in the target device. We describe the novel transformations supported by our new solution, show example applications and report on first user tests.

Author Keywords
Multi-device Web interfaces, Mobile interfaces, Model-based design, User interface adaptation.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI).

INTRODUCTION
The increasing availability of mobile devices has stimulated interest in tools for adapting the great number of existing Web applications originally developed for desktop systems into versions that are accessible and usable for mobile devices. In carrying out this adaptation it is important to consider the main characteristics of the target device, in this case the mobile devices. For example, one aspect to consider is that often mobile devices have no pointing device, thus users have to navigate through 5-way keys, which allow them to go left, right, up, down and select the current element. There are also softkeys, which are used to activate commands, but their number and purpose vary depending on the device. In addition, text input is slow and users often have to pay to access the information, and thus prefer short sessions.

In general there are various approaches to authoring multi-device interfaces:

- *Device-specific authoring*, which means that a specific version is developed separately for each target platform. One example is the Amazon Web site, which has a separate version (http://www.amazon.com/anywhere) for mobile devices. This approach is clearly expensive in terms of time and effort.
- *Multiple-device authoring*, which is similar to the previous one but utilises a single application that has separate parts for different platforms. An example is the use of CSS depending on the platform.
- *Single authoring*, in which only one version of the application is built and then adapted to various target platforms. There are two main possibilities for this purpose: either to include the authors' hints or to specify an abstract description, which is then refined according to the target platform (see for example SUPPLE [8]).
- *Automatic re-authoring*, which means an automatic transformation of a version for a given platform, usually the desktop, into a version for the target platform.

Our work falls into the last category, with the aim of obtaining a solution that does not require particular effort in terms of time but is still able to produce meaningful results. Various automatic re-authoring solutions have been proposed. A first distinction can be made depending on where the re-authoring process occurs: the client device, the application server or an intermediate proxy server. We prefer the last solution because performing the transformation on the client device can lead to performance issues with limited capabilities devices, while a solution on the application server would require duplicate installations in all the applications of interest. The feasibility of proxy-

AVI'08, 28-30 May , 2008, Napoli, Italy

based solutions is also shown by its widespread use in tools, such as Google for mobile devices (www.google.com/xhtml) [9], which converts the Web pages identified by the search engine into versions adapted for mobile devices.

In addition, we think that effective solutions for transforming desktop Web sites for mobile access should be based on semantic aspects. Unfortunately, so far the semantic Web has mainly focused on the data semantics through the use of ontologies and languages that allow for more intelligent processing. We would also like to consider the semantics of interaction, which is related to the tasks to support in order to reach the users' goals.

In particular, after discussion of the related work we provide some background information regarding the XML-based language that we use for representing logical user interfaces. Next, we introduce our approach and the general architecture of our tool, and explain how we automatically obtain logical descriptions of existing implementations through a reverse engineering process. We describe in detail how our semantic redesign algorithm works for transforming desktop versions for mobile devices, show example applications, and report on first user tests, which have provided useful suggestions to further improve the tool. Lastly, we draw some conclusions along with indications for future work.

RELATED WORK

Several solutions for automatic re-authoring from desktop-to-mobile have been proposed in recent years. The simplest one just proposes resizing the elements according to the size of the target screen. However, it often generates unusable results with unreadable elements and presentation structures unsuitable for the mobile device. Thus, research work has focused on transformations able to go further, to modify both the content and structure originally designed for desktop systems to make them suitable for small screen displays. Also in this case various possibilities have been explored. The most common transformation supported by current mobile devices is conversion into a single column (the narrow solution): the order of the content follows that of the mark-up file starting from the top, the images are scaled to the size of the screen, and the text is always visible and the content compacted without blank spaces. It eliminates horizontal scrolling, though it greatly increases the amount of vertical scrolling. For example, Opera SSR (Small Screen Rendering, www.opera.com/products /smartphone/smallscreen/) uses a remote server to pre-process Web pages before sending them to a mobile device, Web content is compressed to reduce the size of data transfers. In general, in this solution content requiring a good deal of space such as maps and tables can become unreadable; and often it is difficult to understand that the corresponding desktop page has changed because the initial parts of several desktop pages are indistinguishable from each other. Digestor [4] and Power Browser [5] have been solutions that use proxy-based transformations (as in our case) in order to modify the content and structure of Web pages for mobile use. However, they do not use logical descriptions of user interfaces in order to reason about the page re-design or apply analysis of the sustainable costs of the target device, as happens in our case.

Various approaches have considered the application of information visualization techniques to address these issues. Fish-eye representations have been considered, for example Fishnet [3], which is a fisheye Web browser that shows a focus region at a readable scale, while spatially compressing page content outside the focus region. However, generating fish-eye representations of desktop Web pages in mobile devices can require excessive processing. Overview + detail splits a Web page into multiple sections and provides an overview page with links to these sections. The overview page can be either a thumbnail image, or a text summary of the Web page. Within this approach various solutions have been proposed. Smartview [12] provides a zoomed-out thumbnail view of the original Web page fitting it on the screen horizontally. The approach partitions the page into logical regions; when one is selected, content is presented inside the screen space in detailed view. Summary Thumbnails [10] uses the same thumbnail approach but the texts are summarized enabling good legibility (fonts are enlarged to a legible size and characters are cropped from right to left until the sentence fits in the available area). The main issue with this type of approach is that it works well in some cases, less in others, because it mainly focuses on the transformation of specific elements (for example, Summary Thumbnail mainly works on text snippets). Another contribution in this area is MiniMap [13] a browser for Nokia 6600 mobile phones developed at Nokia Research. The user interface is organised in such a way that text size should not exceed the screen space and provides an overview+detail representation. The overview is given by an area dedicated to showing where the current mobile page is located in the original desktop page. However, this solution is effective only with mobile devices with relatively large screens.

We believe that model-based approaches can provide a more general solution. They are based on the use of logical descriptions that capture the main semantic aspects of the user interface and hide low-level details. Some first studies of how to apply them in this context have already been proposed (see for example [6][7]). These works were useful to provide solutions to specific issues raised by supporting mobile user interactions, but they did not address the issue of providing a general solution for taking Web sites originally developed for desktop systems and dynamically transforming them into accessible and usable versions for mobile devices while users are accessing them.

TERESA XML

Several approaches for representing logical descriptions of user interfaces have been proposed. (e.g. UIML [1], UsiXML[11]). We use TERESA XML because it provides

not only abstractions of the single interface elements but also various ways to compose and structure them, which is an important aspect in user interface design. In this section we summarise the main features of this language in order to make the reader in a position to understand how we exploited it in our new tool. TERESA XML supports the various possible abstraction levels (task, abstract and concrete user interface). The task level describes the activities that should be supported. An abstract user interface is composed of a number of presentations and connections among them. While each presentation defines a set of interaction techniques perceivable by the user at a given time, the connections define the dynamic behaviour of the user interface, by indicating what interactions trigger a change of presentation and what the next presentation is. There are *abstract* interactors indicating the possible interactions in a platform-independent manner: for instance we just indicate the type of interaction to be performed (e.g.: selection, editing, etc.) without any reference to concrete ways to support such an interaction (e.g.: selecting an object through a radio button or a pull-down menu, etc.). At this level we also describe how to compose such basic elements through some composition operators. Such operators can involve one or two expressions, each of them can be composed of one or several interactors or, in turn, compositions of interactors. In particular, the composition operators have been defined taking into account the type of communication effects that designers aim to achieve when they create a presentation. They are: *Grouping*: indicates a set of interface elements logically connected to each other; *Relation*: highlights that one element has some effects on a set of elements; *Ordering*: some kind of ordering among a set of elements can be indicated; *Hierarchy*: different levels of importance can be defined among a set of elements.

The *concrete* level is a refinement of the abstract interface: depending on the type of platform considered there are different ways to render the various interactors and composition operators of the abstract user interface. The concrete elements are obtained as a refinement of the abstract ones. For example, a navigator (an abstract interactor) can be implemented, either through a textlink, or an imagelink or a simple button, and in the same way, a single choice object can be implemented using either a radio button or a list box or a drop-down list. The same holds for the composition operators: for example an abstract grouping operator can be refined at the concrete level in a desktop platform by a number of techniques including both unordered lists by row and unordered lists by column (apart from classical grouping techniques such as fieldsets, bullets, and colours). The small capability of a mobile phone does not allow for implementing the grouping operator by using an unordered list of elements by row, thus this technique is not available on this platform.

THE ARCHITECTURE OF SEMANTIC TRANSFORMER

Our tool (Semantic Transformer) acts as a server, which includes proxy functionalities. When a request from a mobile device is detected, it is forwarded to the application server, which provides the corresponding page, and then the proxy server manipulates it through three stages: reverse, redesign, and page generation (see Figure 1 for the overall architecture). While in previous work we considered similar underlying techniques to support user interface migration, here we present a novel solution from many respects. Previous solutions [2] were able to handle only HTML pages, while here we are also able to consider the associated CSS files and some JavaScripts. In addition, previous work [2] had rigid criteria to decide how to split desktop pages for mobile access, while here we present a new solution able to dynamically calculate the cost of a desktop page and compare it with the cost sustainable by the mobile device at hand and exploit such information in deciding whether and how to split the desktop pages. Lastly, previous work [2] was not validated by any user test, while here we also report on the results of a user test.

The purpose of the reverse engineering phase is to build the logical description corresponding to the desktop page accessed by the mobile device. Its result is passed on to the semantic redesign module, which transforms it into a logical description suitable for the mobile device at hand, and which is used as starting point to generate the corresponding page(s) for the mobile device. The next sections provide detailed descriptions of this process.

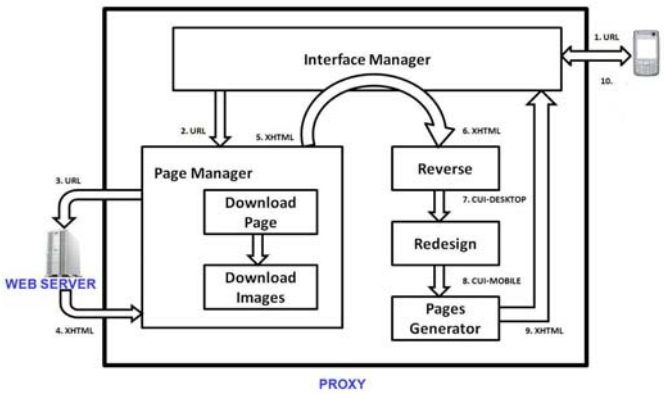


Figure 1: The Architecture of Semantic Transformer.

REVERSE ENGINEERING

The main purpose of the reverse engineering part is to capture the logical design of the interface, which is then used to drive the generation of the interface for the target device. The reverse transformation can reverse Web sites implemented in (X)HTML, including their associated CSS stylesheets. It works by considering one page at a time and reversing each into a concrete presentation. When a page is reversed into a presentation, its elements are reversed into different types of concrete interactors and combinations thereof by recursively analysing the DOM tree of the X/HTML page. In order to work properly it requires well formed X/HTML files as input. However, since many pages available on the Web do not satisfy this requirement, before reversing the page, the W3C Tidy parser is used to correct

features such as missing and mismatching tags and returns the DOM tree of the corrected page. The algorithm of the reverse phase analyses each XHTML element and if such element can be mapped onto a concrete interactor then we have a recursion endpoint. The appropriate interactor element is built and inserted into the XML-based logical description. For example, DOM nodes corresponding to the tags <img>, <a> and <select> cause the generation of concrete objects respectively of type *image*, *navigator* and *selection*. The properties of the objects in the source page considered are also used to fill in the attributes of the corresponding concrete user interface elements, independent of the peculiarities used to implement the page of the source device. For instance, the *italic* attribute of a text concrete element is set to true, though in the HTML implementation it might appear as either <i> or .

If the XHTML node corresponds to a composition operator then, after creating the proper composition element, the function is called recursively on the XHTML node subtrees. The subtree analysis can return both elementary interactors and compositions of them. In both cases the resulting nodes are appended to the composition element from which the analysis started. For example, the node corresponding to the tag <form> is reversed into a *Relation* composition operator, <ol> into an *Ordering*, <ul> into a *Grouping*. Depending on the considered node to be reversed, appropriate attributes are also stored in the resulting element at the concrete level (e.g. typical HTML desktop <ul> lists will be mapped at the concrete level into a *grouping* expression using *vertical bullets*).

If the node does not require the creation of an instance of an interactor in the concrete specification (for example, if the Web page contains the definition of a new font, no new element is added in the concrete description) then if the node has no children, no action is taken and we have a recursion endpoint (this can happen for example with line separators such as
 tags). Otherwise, if the node has children, each child subtree is recursively reversed and the resulting nodes are collected into a *grouping* composition which is in turn added to the result.

In the reversing process, the environment first builds the concrete description and then derives the abstract logical objects corresponding to the different concrete interaction elements. This is a simple matter in TERESA XML because the concrete languages are a refinement of the abstract one, which means that they add a number of attributes to the higher level elements defined in the abstract description. Thus, the process for reversing a concrete description into the corresponding abstract one consists of removing the lower level details from the interactor and composition operators specification, while leaving the structure of the presentations and the connections among presentations unchanged. In practice, there is a many-to-one relation between the elements of the concrete languages and the abstract one (for both the interaction objects and the composition operators).

SEMANTIC REDESIGN

In general, the semantic redesign transformation changes the logical description of a user interface for a given platform into a logical description for a different platform, aiming to support a similar set of tasks and communication goals, but providing indications for an implementation that adapts to the interaction resources available. In our case, the redesign module analyses the input from the desktop logical descriptions and generates an abstract and concrete description for the mobile platform, from which it is possible to automatically generate the corresponding user interface. Previous approaches to semantic redesign [2] were unable to dynamically calculate the cost sustainable by the target device or that of the resources consumed by the Web pages under consideration, thus providing rather limited results in terms of adaptation.

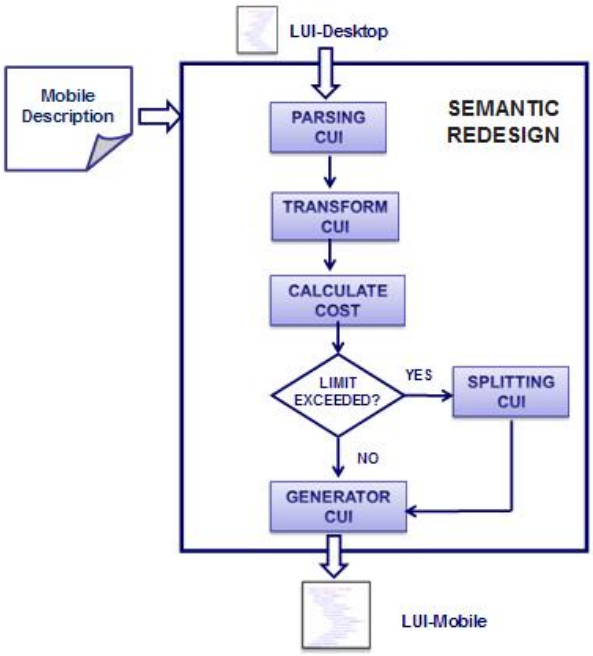


Figure 2: The Semantic Redesign Algorithm.

Figure 2 shows the various phases of semantic redesign in the case of desktop-to-mobile transformations. After parsing the logical desktop description, there are three main phases: transforming the desktop logical interface into a mobile logical interface, calculating the cost of such a new user interface in terms of resources, and splitting the logical interface into presentations that fit the cost sustainable by the target device. In the first transformation, we mainly change the concrete elements of the desktop description into concrete elements that are supported by the mobile platform (for example, a radio-button with several elements can be replaced with a pull-down menu that occupies less screen space). In this transformation the images are resized according to the screen size of the target device, keeping the same aspect ratio. In some cases, they may not be rendered at all because the resulting resized image would be too small or the mobile device does not support them. Text

and labels can be transformed as well, since they may be too long for the mobile device. In converting labels we use tables able to identify shorter synonyms.

In order to automatically redesign a desktop presentation for a mobile device, we need to consider semantic information and the limits of the available resources. If we consider only the physical limitations, we may end up dividing large pages into smaller ones that are not meaningful. To avoid this, we also consider the composition operators indicated in the logical descriptions.

To this end, our algorithm tries to maintain interactors that are composed together through some composition operator in the same final presentation, thus preserving the communication goals of the designer and obtaining consistent interfaces. In addition, the page splitting requires a change in the navigation structure with the need for additional navigator interactors that allow access to the newly created pages. More specifically, the algorithm for calculating the costs and splitting the presentations accordingly is based on the number and cost of interactors and their compositions. The cost is related to the interaction resources consumed, e.g.: number of pixels of images, font sizes. After the initial transformation, which replaces the desktop concrete elements with mobile concrete elements (for example, a text area for the desktop could be transformed into a simpler text edit on the mobile), the cost of each presentation is calculated. If it fits the cost sustainable by the target device, then no other processing is applied. Otherwise, the presentation is split into two or more pages following this approach. The cost of each composition of elements is calculated. The one with the highest cost is associated to a newly generated presentation and is replaced in the original presentation with a link to the resulting new presentation. Thus, if the cost of the original presentation after this modification is less or equal the maximum cost that can be supported by a single mobile presentation, then the process terminates, otherwise the algorithm is recursively applied to the remaining composition of elements. In case of a complex composition of interface elements that might not be entirely included in a single presentation because of its high cost for the target device, the algorithm aims to distribute the interactors equally amongst presentations of the mobile device.

In the transformation process we take into account semantic aspects and the cost in terms of interaction resources of the elements considered. The cost that can be supported by the target mobile device is calculated by identifying the characteristics of the device through the *user agent* information in the HTTP protocol, which can be used to access more detailed information in its description, which is stored in the adaptation server through the WURFL (wurfl.sourceforge.net/), a Device Description Repository containing a catalogue of mobile device information. Initially, we considered UAProfiles but sometimes such descriptions are not available and they require an additional access to another server where they are stored.

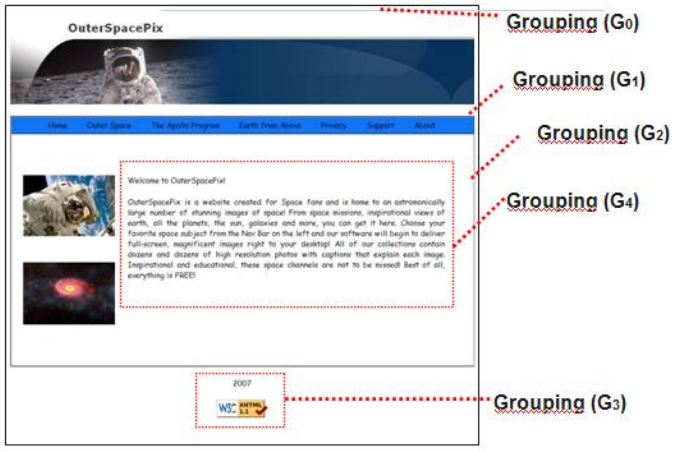


Figure 3: An Example Web Page.

As we already mentioned, examples of elements that determine the cost of interactors are the font size (in pixels) and number of characters in a text, and image size (in pixels), if present. One example of the costs associated with composition operators is the minimum additional space (in pixels) needed to contain all its interactors in a readable layout. This additional value depends on the way the composition operator is implemented (for example, if a grouping is implemented with a fieldset or with bullets). Another example is the minimum and maximum interspace (in pixels) between the composed interactors.

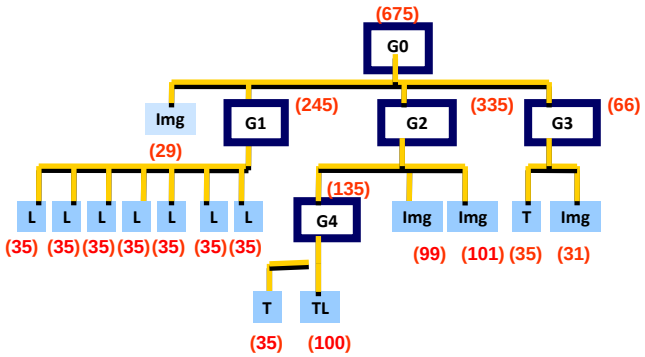


Figure 4: Structure and Cost of the Example Web Page.

EXAMPLE APPLICATIONS

In order to understand how our tool works, we can consider the example shown in Figure 3. For the sake of clarity, we have intentionally chosen an example that is not particularly complex. Through the reverse engineering transformation it is possible to automatically identify five element groupings: one associated with the overall page, one with the navigation bar, one corresponding to the text area, which also belongs to another grouping that includes two side images, and one with some interface elements lined up vertically at the bottom of the page (see Figure 3). Then, the semantic redesign module calculates the space cost of each element, composition of elements, and lastly of the entire page to check whether the page can be sustained by the target device at hand. Figure 4 shows the logical structure

of the page in which the costs of the compositions of elements and their basic elements have been calculated after being transformed for the mobile platform. In Figure 4, T = Text, TL = Long text, Img = image, L = link, Gi (with i=0 …,4)=Grouping.

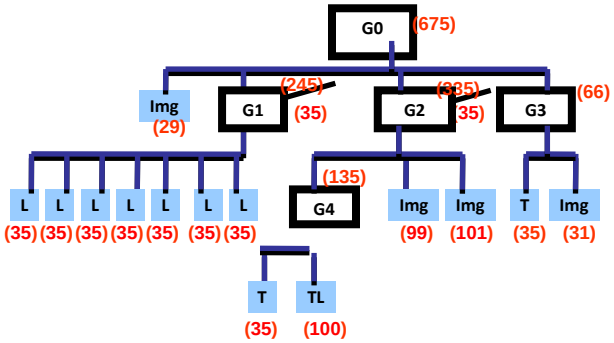


Figure 5: Reduction of Cost of the Example Page.

Since the overall cost is higher than that sustainable by the current mobile device, the transformation process identifies groupings of elements (i.e. the costliest), which are to be replaced with a link (which has much less cost) and allocated to newly created mobile Web pages (see Figure 5). This stage is accomplished through an iterative process, which stops when a satisfactory fit is achieved.

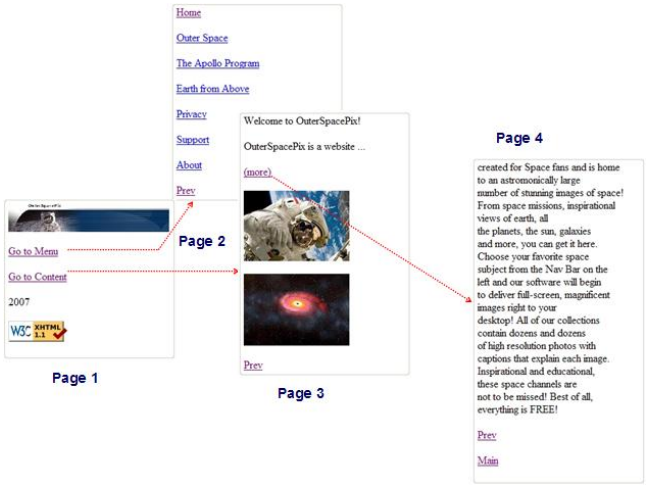


Figure 6: The Corresponding Mobile Web Pages.

In our example, this generates two new mobile pages, one associated with the navigation bar and one with the main content part (see Figure 6). It is worth noting that the names of the newly generated links are meaningful (for example Go to Menu or Go to Content) because they are derived from the id (Menu or Content) of the tag (DIV in this case) used to group the elements. Since one page contains a long text, this is processed in such a way that the initial part appears in the content page and then a "More" link is included to access the corresponding complete passage.

The next example shows how our transformation works in the case of a desktop page containing long text and an interactive form (see Figure 7) and the resulting mobile Web pages (Figure 8). The long text is presented with the initial sentence in the first page and made accessible through a specific link.

Figure 7: Another Example Desktop Web page.

USER TEST

We performed an evaluation test to assess the usability of the Web pages obtained by the transformation. The test involved a group of 10 users, with an average age of 28.6, most of them with a high level of education (80% had a university degree), half of them had previous experience in using a mobile device for accessing the Web. Moreover, users had good experience in using desktop PCs, although they had never heard about redesign tools. During the test the users were shown some Web pages originally implemented for the desktop platform. Then, after having analysed them, they used three different redesign tools in order to automatically obtain a version of the page redesigned for the mobile platform: *Google* [www.google.com/xhtml][9], *SemanticTransformer* (the prototype developed in our laboratory) and *Skweezer* [www.skweezer.net], another commercially available tool. The order of use of the tools was counterbalanced in order

to avoid learning effects, which could have an impact on the evaluation.

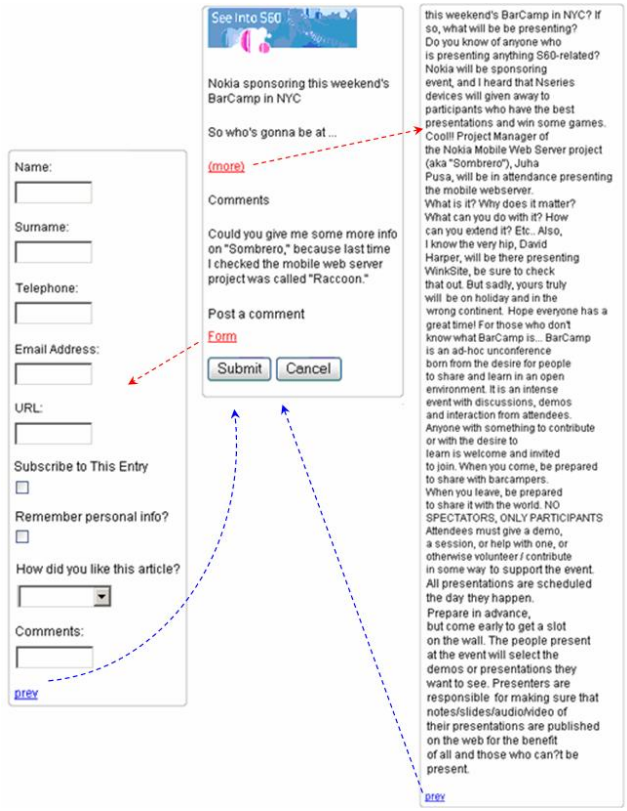


Figure 8: The Resulting Mobile Web pages.

The users were expected to compare the results produced by the different redesign tools proposed, and assess advantages and disadvantages of the considered systems. At the beginning of the test the users read an introduction explaining the goals of the system. The users were requested to navigate/analyse the pages produced by the three tools. The tools used for comparing the results produced by our tool (SemanticTransformer) presented some common characteristics. Figure 9 shows one of the Web pages used in the evaluation test. Afterwards the users analysed the pages obtained as a result of the redesign tools.

As you can see from Figure 10, the pages produced by Google and Skweezer had some common characteristics: a vertical scrolling is available for the navigation of the page, whereas SemanticTransformer produced more than one page in the target mobile platform, in order to reduce the need of vertical scrolling. After having analysed the different pages, each user had to fill in an evaluation questionnaire. The questions concerned: the easiness of putting in correspondence the page of the mobile platform with the associated page for the desktop; the way in which redesign tools supported the transformation of the images in the desktop page; the way in which redesign tools supported the transformation of the *long texts* in the desktop page; the transformation of other user interface objects (for instance: radio button, text area, drop down list); the overall usability of the pages produced by the different tools analysed during the test; assessing the split of a desktop page into several mobile pages; the convenience of splitting a desktop page into several pages of a mobile platform.

Most of the questions the user had to answer had a 1-5 scale (1 means the worst value, e.g.: very difficult, ineffective,… while 5 means the best one). Regarding the easiness of putting in correspondence the page of the mobile platform with the associated page for the desktop (Google: M=3.8; SD=1.03; SemanticTransformer: M=3.4; SD=0.97; Skweezer: M=4; SD=1.05), on the one hand, the pages produced by Google and Skweezer resulted to be easier to be associated due to the direct correspondence between the desktop version and the mobile one. On the other hand, the pages produced by the SemanticTransformer have a less direct mapping because of the splitting of the pages, which can even disorient the user, especially when the page included several links. The page splitting used by SemanticTransformer, made more difficult putting in correspondence the pages of the two platforms with respect to the technique using the vertical scrolling, since the user had to get accustomed to the division of the pages. As for the transformation of the images, the evaluation did not reveal big differences among the various tools (Google: M=2.6; SD=1.17; SemanticTransformer: M=3.8; SD=1.03; Skweezer: M=3.1; SD=1.19). To the question regarding the transformation of the long texts most of the users replied that they preferred to have the text divided into several parts with respect to presenting the text compressed vertically, as it happens with Google and Skweezer (Google: M=3.9; SD=0.88; SemanticTransformer: M=4; SD=0.67; Skweezer: M=4; SD=1.05).

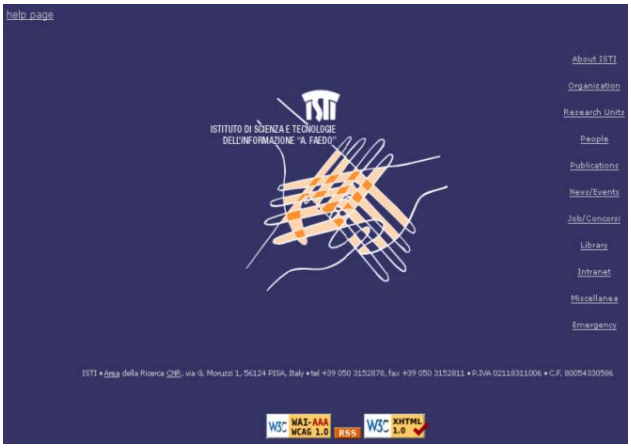


Figure 9: A Web page used in the test (www.isti.cnr.it)

Also, the users appreciated the transformation of some elements of the form (for instance the radio button was transformed in a drop down list or the textarea in a textfield) done by SemanticTransformer (Google: M=2.7; SD=1.64; SemanticTransformer: M=3.9; SD=0.99; Skweezer: M=2.6; SD=1.71).
Regarding the overall usability of the pages produced by the different tools analysed during the test, by analysing the

comments of the users it came out that the approach of the page splitting was appreciated by the users with respect to the vertical scrolling (Google: M=2.7; SD=1.64; SemanticTransformer: M=4.2; SD=0.92; Skweezer: M=3.3; SD=1.41); between Google and Skweezer the latter was the preferred one because it presented the elements of the form in a more compact manner: indeed, the pages generated by Skweezer include a reference to a CSS file which contains a number of rules (eg: re-definition of fontsizes for H1, H2, H3 etc.) aimed at using more efficiently the space available on the mobile device. Some expert users highlighted the possible issue of the number of requests that the mobile device has to send to the proxy server when several pages are produced by the splitting algorithm (it should be analysed whether it would be better to have several requests of small pages or one single request of a bigger page). The average evaluation regarding the splitting of a desktop page into several mobile pages done by SemanticTransformer for the mobile device was 3.8 (Google: M=2.9; SD=1.45; SemanticTransformer: M=3.8; SD=0.79; Skweezer: M=3.1; SD=1.37). As possible suggestions, the users indicated to visualise the path followed by the user to reach the current page, which should be useful especially when the splitting algorithm generates many pages; insert in every page a link to come back to the main page and also the possibility for the users to move between the two different redesign modalities (page splitting and vertical scrolling).

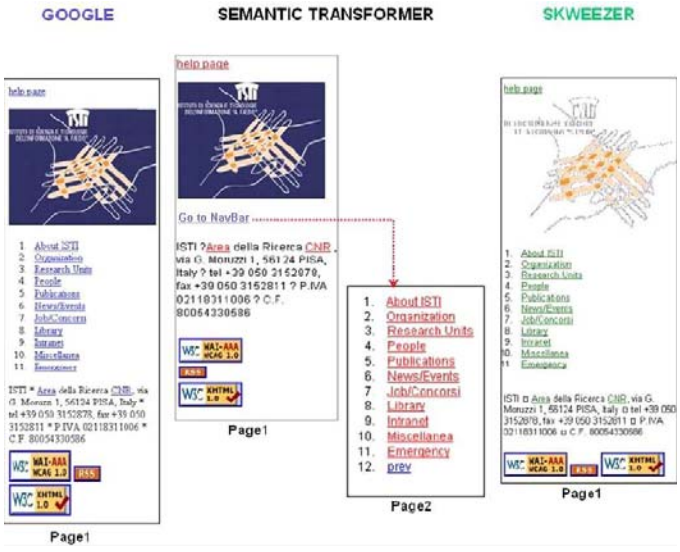


Figure 10: The resulting pages produced by the three tools.

The users also highlighted the need for more meaningful link labels for navigating between the pages that are added by the algorithm, so that they can have a clearer idea of the associated page content. Regarding the convenience and the opportunity of the splitting technique, 70% of the users answered positively, so provided encouraging feedback for pursuing this approach.

CONCLUSIONS AND ACNOWLEDGMENTS

We have presented a tool for the automatic transformation of desktop Web sites for mobile access exploiting logical user interface descriptions. We carried out a user test, which provided positive feedback and suggestions for small refinements.

We thank Agazio Gregorace for help in the implementation of the tool.

REFERENCES

1. Abrams, M., Phanouriou, C., Batongbacal, A., Williams, S., Shuster, J. UIML: An Appliance-Independent XML User Interface Language, Proceedings of the 8th WWW conference, 1999.
2. Bandelloni, R., Mori, G., Paternò, F., Dynamic Generation of Migratory Interfaces, Proceedings Mobile HCI 2005, ACM Press, pp.83-90, Salzburg, Sept. 2005.
3. Baudisch, P., Lee, B., Hanna L.: Fishnet, a fisheye Web browser with search term popouts: a comparative evaluation with overview and linear view. AVI 2004: pp.133-140.
4. Bickmore T., Girgensohn A., Sullivan J., Web-page Filtering and Re-Authoring for Mobile Users, The Computer Journal, Vol.42, N., 1999, pp.534-546.
5. Buyukkokten O., Kaljuvee O., Garcia-Molina H., et al.., Efficient Web Browsing on Handheld Devices Using Page and Form Summarization, ATOIS, Vol.20, N.1, January 2002, pp.82-115.
6. Calvary, G., Coutaz, J., Thevenin, D., Bouillon, L., Florins, M., Limbourg, Q., Souchon, N., Vanderdonckt, J., Marucci, L.,, Paternò, F. and Santoro, C. 2002. The CAMELEON Reference Framework, Deliverable D1.1.
7. Florins, M., Vanderdonckt J.: Graceful degradation of user interfaces as a design method for multiplatform systems. Intelligent User Interfaces 2004: pp.140-147.
8. Gajos K., Christianson D., Hoffmann R., Shaked T., Henning K., Long J. J., and Weld D. S.. Fast and robust interface generation for ubiquitous applications. In UBICOMP'05, pp. 37–55. Springer Verlag LNCS 3660.
9. Kamvar M., Baluja S.. A Large Scale Study of Wireless Search Behavior: Google Mobile Search. Proceedings CHI 2006, ACM Press.
10. Lam H., Baudisch P., Summary Thumbnails: Readable Overviews for Small Screen Web Browsers, ACM CHI'05, Portland, pp. 681-690, ACM Press, 2005.
11. Limbourg, Q., Vanderdonckt, J., UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence, Engineering Advanced Web Applications, Rinton Press, Paramus, 2004.
12. Milic-Frayling N., Sommerer R.. Smartview: Enhanced document viewer for mobile devices. Technical Report, Microsoft Re-search, Cambridge, UK, November 2002.
13. Roto, V., Popescu, A., Koivisto, A., Vartiainen E.: Minimap: a Web page visualization method for mobile phones. Proceedings CHI 2006: pp.35-44, ACM Press.

Advanced Visualization

A Physics-based Approach for Interactive Manipulation of Graph Visualizations

Andre Suslik Spritzer
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Caixa Postal 15064 91.501-970 Porto Alegre, RS
Brazil
spritzer@inf.ufrgs.br

Carla M.D.S. Freitas
Instituto de Informática
Universidade Federal do Rio Grande do Sul
Caixa Postal 15064 91.501-970 Porto Alegre, RS
Brazil
carla@inf.ufrgs.br

ABSTRACT
This paper presents an interactive physics-based technique for the exploration and dynamic reorganization of graph layouts that takes into account semantic properties which the user might need to emphasize. Many techniques have been proposed that take a graph as input and produce a visualization solely based on its topology, seldom ever relying on the semantic attributes of nodes and edges. These automatic topology-based algorithms might generate aesthetically interesting layouts, but they neglect information that might be important for the user. Among these are the force-directed or energy minimization algorithms, which use physics analogies to produce satisfactory layouts. They consist of applying forces on the nodes, which move until the physical system enters a state of mechanical equilibrium. We propose an extension of this metaphor to include tools for the interactive manipulation of such layouts. These tools are comprised of magnets, which attract nodes with user-specified criteria to the regions surrounding the magnets. Magnets can be nested and also used to intuitively perform set operations such as union and intersection, becoming thus an intuitive visual tool for sorting through the datasets. To evaluate the technique we discuss how they can be used to perform common graph visualization tasks.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *graphical user interfaces, interaction styles.*

General Terms
Design, Human Factors.

Keywords
Graph visualization, Interaction

1. INTRODUCTION
Graphs are present in many different application areas, ranging from biology to social network analysis and software engineering. While information organized in graph-like structures can be explored textually, through tools such as query languages, this usually requires an expert user, being too complex and not intuitive enough for most people, who have little experience with such instruments. Therefore, many of these areas make use of applications that visualize their inherent graphs in order to make it easier for their users to grasp and manipulate the information they require.

By far, the most popular and intuitive visual representation of a graph is the node-link diagram. A large community of researchers is dedicated specifically to the study of how to compute the best possible layout. The problem they attempt to solve can be simply stated as how to find the geometric positions of the nodes that are aesthetically more interesting for the better comprehension of the graph and its structure. Its solution, though, has proven to be quite complex.

Different algorithms for the layout of node-link diagrams have been created, each favoring certain aesthetic criteria, such as edge crossings, edge bends, graph symmetry, etc., in detriment of others. Some of these techniques are better for certain applications while some are better for others, but all have their limitations, which range from computational cost to visual clutter. A good source on the field of graph drawing is the book written by Di Battista et al. [2].

To deal with the limitations of these layout algorithms, many approaches have been experimented [11]. While some have applied navigation and interaction schemes on the traditional layouts, others have built 3D visualization techniques, changed from node-link diagrams to alternative visual representations, or combined existent techniques into new hybrid ones.

While some of these techniques might be well suited for certain applications, no technique is generally applicable. Also, while many techniques can build aesthetically pleasing layouts, few take into account the semantic information contained in the attributes of the nodes and edges of a graph, focusing only on the topological characteristics. Some use interaction tools such as filtering, to achieve some visual customization based on the semantic information, but very few are attribute-aware and, among those one finds that most are too application-specific to be generally applicable.

The main information associated to a graph is the relationships represented by its topology, but it is often the case that the attributes represented in its nodes and edges are just as important to the user of a graph visualization application, who might be missing out important data and even relationships that are not explicitly expressed.

AVI'08, May 28–30, 2008, Napoli, Italy.

In this work, we present a physically-based technique for the interactive manipulation of graph visualizations. Our technique consists of providing the user with virtual magnets associated with user-defined criteria, which can be topology or attribute-based, and that allow the interactive manipulation of the visualization of a graph in order to make it semantically more interesting and valuable. Magnets can also be used to intuitively perform set operations such as union and intersection, becoming a visual tool for exploring the datasets, and allowing the user to discover new relationships that were previously invisible due to the techniques that focused exclusively on the topology. To illustrate our technique, we show how it can be applied to perform common graph visualization tasks identified by Lee et al. [13].

In the following section, we present an overview of related works. Section 3 describes our approach, and gives some details of its implementation. Section 4 illustrates how it can be applied and in Section 5 we present our conclusions and draw some comments on future work.

2. RELATED WORK

The works related to the technique presented in this paper are mostly in four subjects: force-directed graph layouts; interactive graph layout reorganization; use of "false" elements for layout re-organization; and evaluation and design of graph visualization techniques.

2.1 Force-directed Layouts

Force-directed graph layouts are amongst the most popular in graph visualization due to their pleasing visual results, relative implementation simplicity and flexibility as to the inclusion of new aesthetic criteria. The basic idea of a force-directed algorithm is to treat the graph as a physical system, assigning forces to the nodes and edges, and minimizing its energy until reaching a stable layout. The forces will work to rearrange the positions of the nodes until the system finds itself in a state of mechanical equilibrium. Di Battista et al.'s book [2] presents a good survey of force-directed methods.

One of the first force-directed algorithms was proposed by Eades [6]. It takes the intuitive approach of treating the graph as a mass-spring system, with nodes being steel rings and edges springs that connect them. Despite the physical metaphor, it does not aim for physical accuracy, not employing Hooke's law for the springs and with forces affecting velocity instead of acceleration. It produces visualizations of uniform edge length and allows representation of graph symmetry. The algorithm consists of randomly positioning the nodes and simply running the simulation for a number of iterations, which is one of its drawbacks, since not all graphs converge at the same time.

Many other algorithms extended the basic idea presented by Eades. One of those is Kamada and Kawai's [12], which aims at positioning nodes in a way that their geometric distance is equal to their graph-theoretic distance. It does so by having the simulation assuming that between every two nodes there is a spring with length equal to the theoretical distance between them. Another interesting algorithm is Davidson and Harel's [5], which introduced the idea of using simulated annealing to minimize the system's energy function, which takes into account vertex distribution, edge length and edge crossings. This algorithm can be very time consuming, but can produce better results.

One of the most popular force-directed algorithms is the one proposed by Fruchterman and Reingold [8]. This algorithm consists on calculating all the forces that attract and repulse nodes, at each iteration. Nodes connected by edges exert an attraction force between them, while all nodes exert a repulsion force on all others. From the forces, the position displacement a node will suffer during each iteration is calculated and limited by the current value of an attribute (usually used as temperature), which is progressively decreased. This algorithm is relatively fast and produces nice visual results.

Another interesting approach is Noack's LinLog energy model [14], which attempts to reveal clusters of highly connected nodes. This technique is particularly useful for datasets such as social networks, and was proposed in two variations, the node-repulsion LinLog model [14][15] and the edge-repulsion LinLog model [16]. Both variations produce similar drawings, but the latter avoids dense accumulations of nodes with high degrees for graphs with non-uniform degrees.

One interesting property of force-directed algorithms is that most of them support the application of constraints. A position constraint can be established by forcing nodes to remain within a certain region, while other types of constraints can be used if they can be expressed with forces. Examples of this include the use of magnetic fields to impose orientation constraints [17] and the utilization of dummy nodes to force groupings.

For many years, force-directed algorithms have suffered dramatically from a scalability problem: the more nodes and edges we have, the slower it is for the system to converge. Thus, it was only possible to use these algorithms in real-time with smaller graphs due to their high computational cost. However, with the advent of faster, multi-core processors and powerful, programmable graphic processing units (GPUs) this reality is changing fast. Mass-spring algorithms can now deal with hundreds of thousands nodes and edges in real-time, and many different applications, such as real-time cloth simulation, are already making use of that [9, 18]. Recent works on GPU-based force-directed layout include Frishman and Tal [7], reporting a multi-level graph layout algorithm.

Aside from the scalability problem, force-directed algorithms also suffer greatly from a predictability problem. Two different runs of an algorithm over similar (or even the same) input graphs might generate two completely different layouts, which is not very helpful in allowing the user to create and maintain a mental map of the visualization. One approach that has been used to minimize this is to run another layout algorithm first, and afterwards execute the force-directed technique on that.

2.2 Interactive Layout Reorganization

Most graph visualization techniques usually use interaction and navigation techniques to explore static, pre-computed layouts. Well-known techniques include filtering; fish-eye views; scrolling and panning; zooming and even coordination of two or more visualizations (see Herman et al. [11], for a wider review on navigation and interaction techniques for graph visualizations). Very few techniques, though, allow for dynamic interactive reorganization of graph layouts.

Some applications allow for simple layout reorganization by letting the user move around nodes in force-directed layouts, which will cause an alteration in the balance of energy of the

force-directed system, thus triggering a repositioning of the nodes, which will move until equilibrium is again reached. Another known technique is to find clusters of nodes and transform them into cluster-nodes that can be expanded and collapsed by the user. Clusters can also be used to perform cluster-based semantic zooming, which allows for a level-of-detail-like approach to the visualization, letting the user incrementally explore the graph by zooming in or out.

Considering the few techniques that allow for dynamic graph layout reorganization, we find the work of Henry et al., NodeTrix [10], which is a hybrid of matrix and node-link visualizations. NodeTrix allows the user to turn clusters of nodes of node-link visualizations into matrices, which are then displayed within the node-link diagram. The layout itself is computed with the previously mentioned LinLog algorithm.

2.3 Use of False Elements

The next few related works are not exactly devoted to graph layout reorganization, but they introduced ideas that we found inspiring and somehow proved the feasibility of using magnets to allow users to re-organize visualizations in a more powerful and easy way.

Fidg't[1] is an application developed for the management of social networks that includes an interactive visualization tool that allows users to iteratively explore their networks by creating tag magnets for pictures (from Flickr) or music (from Last.fm), and observing how its nodes are attracted or repelled.

Although devoted to a different data domain (multivariate information), the Dust & Magnet information visualization technique proposed by Yi et al. [19] also uses a magnet metaphor.

Finally, it is important to mention that the technique presented in our work was partly inspired by Bier and Stone's Snap-Dragging [4], which is an interactive technique that aims at helping the user make precise line drawings.

2.4 Evaluation and Design of Graph Visualization Techniques

Evaluating such a variety of graph layout techniques and interaction techniques is a huge problem, because there are both perceptual and functional issues involved. Few works deal with evaluation of graph visualization techniques. A very useful task taxonomy for graph visualizations has been proposed by Lee et al. [13]. In their article, the authors provide a list of tasks that users might need to perform while using a graph visualization application. In Section 4 we will use this taxonomy to evaluate how our technique fares when the user tries to execute the defined tasks.

3. OUR TOOL

The goal of our technique is to aid users in interactively reorganizing the layout of a graph to better fit their needs by providing them with tools that allow the manipulation of graph visualizations based on the topological and semantic attributes that better interest them. To do so, we build on the physics metaphor of force-directed algorithms by allowing the placement of virtual magnets, which attract nodes that fulfil certain user-defined criteria. While we follow a magnet metaphor, though, physical accuracy is not one of our aims.

In our technique magnets can be placed on the scene in order to reorganize the graph. They can be set to attract nodes based on their values for certain criteria, which can be topological (such as nodes that have a certain degree or that have a path to another node with a certain length) or attribute-based (i.e. all users that come from the UK). Also, boundary shapes can be applied to magnets to keep the nodes they attract bound to certain regions of the scene.

3.1 Basic Graph Layout

In our technique, the user is given two options for the computation of the layout: a mass-spring algorithm similar to Eades's [6] or an adapted version of Fruchterman and Reingold's technique [8].

In the first option, the layout of the graph is computed assuming that all nodes have equal mass. The parameters of the algorithm, such as time step, edge rest length, damping factor and node repulsion force can be changed by the user to produce a visualization that is more satisfying aesthetically. The rest length of the edges can be either a fixed value provided by the user or computed based on the degree of the nodes that it links (the higher the degree, the longest the length). The layout is dynamic and is always being recomputed; therefore, any change in the position of a node will trigger a subsequent reorganization of the layout. Figure 1 shows a small graph with layout computed using this algorithm.

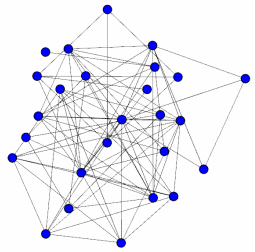


Figure 1. Graph of the largest component (largest connected subgraph) of the AVI coauthorship network drawn using a mass-spring algorithm.

In the second option, Fruchterman and Reingold's technique is combined with the Barnes and Hut algorithm [3], and slightly adapted to better fit our needs. Our modifications were the addition of a small gravitational force that pulls all nodes slightly towards the centre of the workspace and the alteration of the manner in which the algorithm runs (we re-evaluate it every frame instead of running it for a given number of iterations). The user can set several parameters, such as time step, damping, a constant that is used for the computation of the optimal distance between two vertices, maximum attraction force (to make it easier for the simulation to reach stability), attraction and repulsion exponents and central gravitation factor. Figure 2 shows the same graph as Figure 1 computed using this algorithm.

[1] http://fidgt.com

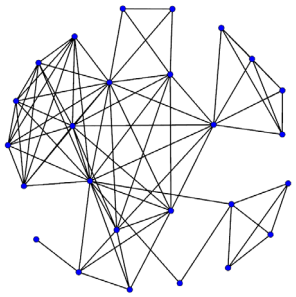


Figure 2. Graph of the largest component of the AVI coauthorship network drawn using our variation of Fruchterman and Reingold's technique.

In both cases, Verlet integration is used to compute node positions in every frame due to its stability and area preserving properties. To allow for easier navigation, the user can pause and resume the simulation at any time.

3.2 Magnets

Magnets are special objects that can be added to the scene which have the ability to attract nodes of a graph that fulfil certain user-defined criteria. Figure 3 shows how magnets are visually represented in the prototype we developed.

Figure 3. Visual representation of a magnet.

A magnet works by exerting onto each of these nodes an attraction force that will progressively move them towards it, thereby building a cluster of semantically-related nodes around it. When these nodes move, the force-directed layout algorithm ensures that all the other nodes that are connected to them by edges will be pulled along, reorganizing the whole layout of the graph in the process.

To each magnet users should associate one or more attraction criteria, which can be set as requirements of attraction or simply criteria. To be attracted a node must fulfil all requirements and at least one of the defined criteria. These requirements and criteria can be based on the topology of the graph, the attributes of its nodes and edges or even other magnets that have been placed on the scene.

Topology-based criteria use the structure of the graph to attract nodes. It is possible to attract nodes based on properties such as degree, path length (i.e. all nodes that are within a specified path length from another node or group of nodes), connected subgraph (i.e. subgraphs with a given number of nodes), connected components (maximally connected subgraphs with a given number of nodes) . Figure 4 shows an example of two magnets with topology-based criteria in action.

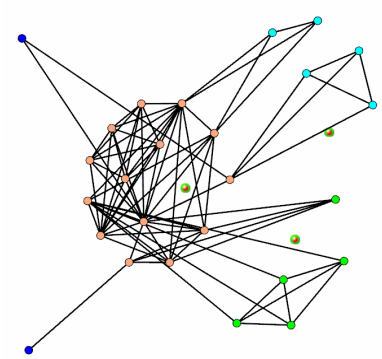


Figure 4. A small graph with three magnets - one targeting nodes of degree 3 (light blue nodes); the second attracting nodes of degree 4 (green nodes) and the third one attracting nodes with degree greater than 5 (light orange nodes).

Attribute-based criteria use the semantic properties contained in nodes and edges in order to attract nodes. Users can set a magnet to attract all the nodes in which a certain property exists, or not only exists but is also equal to a certain value or is within a certain value range (if it is numerical). Users can do the same for edges, with the magnet then attracting the nodes linked by edges that fulfil the defined criteria. An example of a magnet with an attribute-based criterion can be observed in Figure 5.

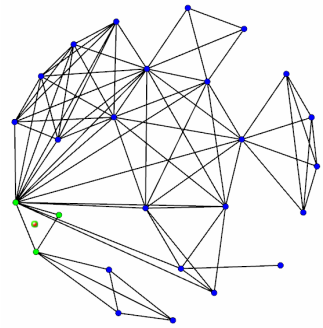


Figure 5. Graph of the largest component (largest connected subgraph) of the AVI coauthorship network with a magnet set to attract authors cited more than once (all nodes that have attribute "citationsnb" greater than 1).

Magnet-based criteria use the sets of attracted nodes of each magnet that was included in the scene by the user. With magnet-based criteria, one can set a magnet to attract all the nodes that another magnet also attracts, all the nodes that another magnet does not attract, all the nodes that no magnet attracts or all the nodes that are attracted by a combination of magnets. This allows for set-based operations on the graph visualization, which usually will end up in a graph reorganization.

Each criterion has a properties dialog through which it can be properly set up and configured. They can be added, removed and edited at any time, with users also being able to add multiple criteria and requirements to each magnet. This makes it possible, for instance, to set a magnet to attract all nodes that are not attracted by any magnet, have degree higher than a certain number and include the property that is called *x*. Figure 6 shows an example of a magnet with different types of criteria.

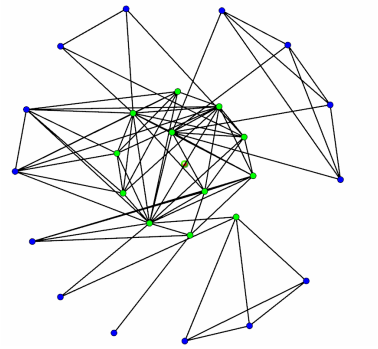


Figure 6. Magnet attracting authors with at least two papers that have written papers with more than three other authors (nodes with the attribute "papersnb" greater than or equal to 2 and degree greater than 3).

In case one might wish to perform a union of the set of nodes attracted by two or more different magnets, it is possible to combine such magnets. The combination operation will create a new compound magnet with the combined criteria of the ones selected. The new magnet will have its force magnitude set to the average of the original ones', which will be subsequently set to 0.

Within the physics metaphor, a magnet works simply, on each frame, by applying to all attracted nodes a force vector in its direction with the specified magnitude. Also, to keep the magnet from being overlapped by its attracted nodes and to keep the attracted nodes from staying all bundled together too close to each other, the magnet also exerts a repulsion force on each of the attracted nodes. This repulsion force is the same as with common nodes, working like a reverse gravity by being inversely proportional to the distance of the node to the magnet and proportional to the magnitude of the force of attraction, so that it is stronger with the nodes that are near the magnet and weaker with the ones that are progressively further away from it. The magnitude of the repulsion force can be increased by the user to change the minimum distance the nodes ought to have from the magnet.

Occasionally it might be cumbersome to see which nodes that are positioned close to a magnet are in fact attracted by it. To deal with this situation, it is possible to assign a colour to all the nodes that it attracts or to create a boundary shape around the magnet to limit the region in which such nodes can move about.

3.3 Boundary Shapes

A boundary shape is simply a geometric shape (a circle in our current implementation), which can be placed around a magnet and have the function of bounding the nodes that such magnet attracts to the region that the shape delimits. At the same time that all attracted nodes are kept within the boundary shape, all other nodes are kept out, with the shape exerting a repulsion force similar to the one exerted on the other nodes by the nodes themselves (a reverse gravity force).

To allow for a better distribution of space within a boundary shape, the magnitude of the attraction force of the magnet is reduced for the nodes that are inside. Also, when a node enters the region of a boundary shape, the direction of the force that pulls it is "refracted" by the application of Snell's Law. So, instead of there being a force that pulls the nodes straight to the magnet, each node is pulled towards a nearby direction, which makes for more evenly distributed nodes. Figure 7 shows boundary shapes in action.

Once a node finds itself inside a magnet's boundary shape, it cannot escape that area, unless it is also attracted by another magnet that is placed outside such shape.

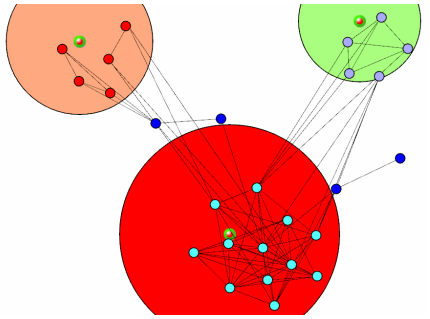


Figure 7. Graph with boundary shapes.

3.4 Magnet Hierarchy

A magnet effectively creates sets of related nodes and ensures that they remain near a certain physical region. Occasionally it might be useful to refine this set of nodes into subsets. To allow for that, it is possible to define magnets that act only on the subset of the graph that is already attracted by another magnet. To do this, the user must simply create a magnet and define another one as its parent. It is interesting to note that children magnets might children magnets of their own; creating thus a hierarchy of magnets that might be helpful for incremental exploration of a graph. Figure 8 illustrates the use of magnet hierarchy to achieve a better organization of the layout.

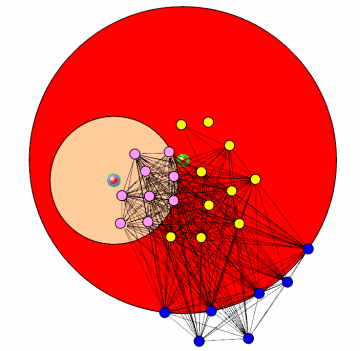


Figure 8. Graph of the CHI conference citations with a magnet hierarchy. All nodes within the red boundary shape are proceedings of the conferences that were published before 2000, with the ones inside the beige shape being from after 1990.

If the parent magnet does not have a boundary shape, or has one, but the child magnet is outside of it, a dashed line in the same colour as the parent's nodes appears between them. If there is a boundary shape and the child magnet is within it, no line appears.

3.5 Magnet Intersections

Figure 9. Visual representation of an intersecting node.

Occasionally it may happen that a node fulfils the criteria of two or more magnets. In such a case, the node will be attracted to all of these magnets and will thus have a tendency to stabilize in the middle of them with a bigger lean towards the ones with the strongest attraction forces. If there is intersection between magnets that have boundary shapes, their position constraints are ignored, and they are allowed to escape the region they were previously bound to.

To make the intersections more apparent visually, the common nodes are drawn as in Figure 9. The user may also at any time choose to display dashed lines from the intersecting nodes to their 'parent' magnets (Figure 10). Each line assumes the colour that was defined by the user for the nodes that are attracted by its respective magnet.

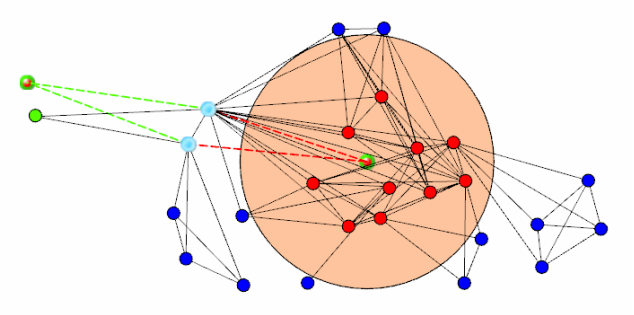


Figure 11. Largest component of the AVI coauthorship network with a magnet that attracts all authors that have more than one paper (node attribute "papersnb" greater than 1) and another that attracts all authors with more than one citation (node attribute "citationsnb" greater than 1).

3.6 Implementation Details

A proof-of-concept prototype was developed to test and evaluate our approach. It was initially developed with Python 2.5 and Qt 4.3.1, using PyQt and later ported to C++. The prototype takes as input GraphML files and displays the graphs with the previously described layouts.

Users are able to insert magnets, whose attributes (including shapes and criteria) can be manipulated through a panel on the right side of the graphical user interface. Each criterion has its own properties dialog, which can be accessed by picking it from the selected magnet's requirement and criteria lists. The prototype was built with extensibility in mind, so that the creation of new tools and types of criteria is straightforward.

The panel on the right side of the user interface is used to provide layout and magnet options to the user. When no node or magnet is selected, information about the graph and the layout configuration interface is displayed. If a magnet is selected, the magnet editor is launched, allowing the user to set and edit the magnet criteria and boundary shape. When the user selects a node, the panel displays the attributes of that node.

As the goal of this prototype was to test our technique, performance considerations were not taken into account in the development of the application. Therefore, the current implementation is completely on software and with only few optimizations. Nevertheless, on the machine used to run it, a single-core Mobile AMD Athlon 64 3000+ (2.0 GHz) with 2 GB of DDR 333 MHz memory and an ATI Radeon 9700 graphics card with 128 MB of memory, it was already possible to deal with graphs of several hundred nodes and edges at interactive rates.

4. DISCUSSION

Lee et al. [13] have proposed a useful task taxonomy for graph visualization in which it is defined a list of tasks that are commonly performed while exploring a graph. They divide these tasks into general low-level tasks, graph-specific tasks and complex tasks, with the latter being further categorized into topology, attribute-based, browsing and overview tasks.

To examine how our technique can contribute to the visualization of a graph, in this section we show how it can be used to better carry out many of the tasks on Lee et al.'s taxonomy.

Table 1. Low-level tasks inherently covered by our technique

Task	Description
1. Filter	Given some conditions on attribute values, find data cases satisfying those conditions.
2. Find Extremum	Find data cases possessing an extreme value of an attribute over its range within the data set.
3. Sort	Given a set of data cases, rank them according to some ordinal metric.
4. Determine Range	Given a set of data cases, rank them according to some ordinal metric.
5. Characterize Distribution	Given a set of data cases and a quantitative attribute of interest, characterize the distribution of that attribute's values over the set.
6. Find Anomalies	Identify any anomalies within a given set of data cases with respect to a given relationship or expectation.
7. Cluster	Given a set of data cases, find clusters of similar attribute values.
8. Correlate	Given a set of data cases and two attributes, determine useful relationships between the values of those attributes.
9. Find Adjacent Nodes	Given a node, find its adjacent nodes.
10. Set Operation	Given multiple sets of nodes, perform set operations on them.

Most of the higher-level tasks are built on combinations of the 10 general low-level visual analytic tasks described by Amar et al. [1] and also three other operations proposed by themselves (with one of them being exclusive to graphs). It is interesting to note how our technique already inherently deals with several of these lower-level tasks. Table 1, partially taken from Lee et al.'s paper,

contains a listing and description of the low-level tasks that our approach is able to cover.

As can be clearly seen from Table 1, these tasks can be performed through our technique simply by adding magnets to the scene with the proper combination of criteria, and allowing the graph to reorganize itself. Tools such as magnet boundary shapes and the ability to operate on magnets themselves (through magnet combination and magnets that have magnet-based criteria) make carrying out these tasks a natural fit, with clustering and set operations being some of the most natural applications for the tools we propose.

Regarding to graph-specific higher level tasks, our technique also provides adequate support to the user in accomplishing several of them. In most cases the tasks can be easily carried out by relying simply on the placement of magnets with the proper combination of topology and attribute-based criteria followed by (if necessary) the proper operations on the magnets themselves (such as magnet-based criteria and magnet combination).

Lee et al. divide graph complex tasks into topology-based tasks, attribute-based tasks, browsing tasks and overview tasks. Our technique is especially useful for the first two categories and can be easily integrated into applications that provide ways to accomplish the other two types of tasks.

Topology-based tasks were further subdivided into a few categories: adjacency, accessibility, common connection and connectivity. Adjacency tasks include finding the set of nodes adjacent to a node, a node's degree and the node with the highest degree. Accessibility includes issues such as finding all the nodes accessible from another one and the set of nodes with distance from another node within a certain range. Common connection corresponds to finding a set of nodes that are connected to all the nodes of a given set, while connectivity includes finding the shortest path between two nodes, finding connected components (defined by Lee et al. as a maximal connected subgraph) and clusters (defined by the same authors as a subgraph of connected components whose nodes have high connectivity).

Attribute-based tasks can work on nodes or edges and include operations such as finding the nodes that have a specific attribute value or that are linked by edges that have a certain attribute or a certain attribute value in a specified range.

Browsing tasks include operations such as following a given path or revisiting a previously visited node.

Finally, overview tasks correspond to exploratory operations performed in order to quickly get an estimate of a certain value, such as the size of a graph or subgraph, or patterns that the graph tends to have.

As can be seen from the previous description of the different types of tasks, magnets apply directly to topology and attribute-based tasks. Such tasks can be accomplished simply by creating magnets with the proper criteria. Browsing and overview tasks can also be helped by the magnets, by making it easier to find nodes on the scene and providing the visualization with some node position predictability, since magnets can be inserted to make sure that nodes that fulfill certain criteria are within a certain region. For the tasks that our tools are unable to cover, the solution is simply a matter of combining it with other techniques, such as fish-eye-like visualizations, overview windows, node search, etc.

One interesting aspect of our technique is that it can be used to easily explore graph datasets by building queries through the specification of magnets and their criteria, and performing set operations on them, becoming thus an intuitive and simplified alternative to query languages or filtering operations, which can be too complex for most end-users that do not have advanced programming and computer skills.

5. CONCLUSIONS AND FUTURE WORK

Even though there is a multitude of graph layout algorithms, there is no one which fits to all types and sizes of graphs. With the work presented in this paper, we aimed at developing a technique that would help circumvent this fact by providing the user with tools that could allow him to shape a layout into one of his/her needs.

On the contrary of most graph layout techniques, which work solely based on the topological structure of the graph, ours also takes into account the information contained in attributes of the nodes and edges. This allows the user to dispose the graph in a layout that can be semantically more interesting.

Our tools, in great part due to the metaphor we employ, make it possible for the user to intuitively navigate through the graph and perform many common graph visualization operations.

One of the biggest drawbacks of force-directed algorithms is that the layouts they produce tend to be unpredictable – different runs on similar graphs (or even with the same one) might generate completely different layouts, which is quite a hindrance for maintaining a mental map of the graph. Our technique helps minimize this limitation, allowing for a level of predictability in otherwise unpredictable drawings. In two runs of the application on the same graph, two magnets will always attract the same nodes to the same place. It is not guaranteed that the nodes will be at the same exact position, but their general location can be easily known, since it is indicated by the users themselves.

Another interesting aspect of the presented technique is that it is not bound to a specific layout algorithm: it can work with any that allows for forces to be applied to nodes.

There is still work to be done in order to improve the technique presented in this paper. Amongst the planned work is an efficient implementation of the technique using the GPU on top of a more sophisticated layout algorithm that more clearly separates clusters of highly connected nodes, such as LinLog [14]. This implementation will allow the use of the technique with larger and more complex datasets, permitting its validation and better adaptation for huge graphs, which have shown up quite frequently lately due to the growing interest in the visualization of social networks.

Some new features are also planned for the technique itself, such as new types of criteria, arbitrarily-shaped magnet boundaries, the possibility of making a magnet work only on the nodes that are within a certain area (i.e. with a certain radius around it), and the ability to collapse the nodes attracted by a magnet into an expandable and collapsible cluster-node to allow for a better iterative visualization. Also planned is a special magnet that applies weighed forces to the nodes it attracts, allowing for a visual sorting of such nodes (the closer the node is to the magnet, the more it has of a certain property). This sorting magnet would

allow operations such as visually browsing by date, alphabetically or any numerical value.

Planned work also includes user experiments for better validation of the technique as well as its integration into a complete graph visualization application that supports other features such as node search, overview windows, coordination with different visualizations, filtering and fish-eye-like views.

6. ACKNOWLEDGMENTS

We gratefully acknowledge the helpful comments of our colleagues as well as Jean-Daniel Fekete and Nathalie Henry. We also thank J.-D. Fekete and N. Henry for providing the datasets we use in the paper. This work is partially sponsored by CNPq (Brazilian Council for Research and Development), specially the CNPQ/INRIA cooperation program (grant nr. 490087/2005-1).

7. REFERENCES

[1] Amar, R., Eagan, J. and Stasko, J. 2005 Low-Level Components of Analytic Activity in Information Visualization. In Proceedings of 11th IEEE Symposium on Information Visualization (2005), 111-147

[2] Battista, G., Eades, P., Tamassia, R., and Tollis I.G. 1999. Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall, New Jersey.

[3] Barnes, J. and Hut, P. 1986. A hierarchical O(N log N) Force-calculation Algorithm. Nature, 324(4).

[4] Bier, E. and Stone, M. 1986 Snap-dragging. ACM Computer Graphics, 20 (August 1986), 233-240.

[5] Davidson, R. and Harel, D. 1996. Drawing Graphs Nicely Using Simulated Annealing, ACM Transaction on Graphics, 15 (4), 301–331.

[6] Eades, P. 1984. A Heuristic for Graph Drawing, Congressus Numerantium, 42(1984), 149–160.

[7] Frishman, Y. and Tal, A. 2007. Multilevel Graph Layout on the GPU. IEEE Transactions on Visualization and Computer Graphics, 13 (Nov-Dec 2007), 1310-1319.

[8] Fruchterman, T.M.J. and Reingold, E.M. 1991. Graph Drawing by Force–Directed Placement. Software - Practice & Experience, 21 (Nov 1991), 1129–1164.

[9] Georgii, J., Echtler, F. and Westermann, R. 2005 Interactive simulation of deformable bodies on GPUs. In Proceedings of Simulation and Visualization, 2005, 247- 258.

[10] Henry, N., Fekete, J.-D. and McGuffin, M. 2007 NodeTrix: Hybrid representation for analyzing social networks. IEEE Transactions on Visualization and Computer Graphics, 13 (Nov-Dec 2007), 1302-1309.

[11] Herman, I., Melancon, G., and Marshall, M. S. 2000.Graph visualization and navigation in information visualization: A survey. IEEE Transactions on Visualization and Computer Graphics, 6 (Jan-Feb 2000), 24-43.

[12] Kamada, T. and Kawai, S. 1989. An Algorithm for Drawing General Undirected Graphs, Information Processing Letters, 31(1989), 7–15.

[13] Lee, B., Plaisant, C., Parr, C., Fekete, J-D., and Henry, N. 2006. Task taxonomy for graph visualization. In Proceedings of the 2006 AVI workshop on BEyond time and errors. (Venice, Italy) BELIV 2006, ACM Press, New York, NY, 81-86, DOI= http://doi.acm.org/10.1145/1168149.1168168.

[14] Noack, A. 2003. Energy Models for Drawing Clustered Small-World Graphs. Technical Report 07/03, Computer Science Reports, Brandenburg University of Technology at Cottbus.

[15] Noack. A. 2004. An Energy Model for Visual Graph Clustering. In Proceedings of the 11th International Symposium on Graph Drawing (Perugia, Italy, Sep. 21-24), GD 2003, Springer-Verlag, Berlin, LNCS 2912, 425-436.

[16] Noack. A. 2005 Energy-Based Clustering of Graphs with Nonuniform Degrees. In Proceedings of the 13th International Symposium on Graph Drawing (Limerick, Ireland, Sep. 12-14), GD 2005, Springer-Verlag, Berlin, LNCS 3843, 309-320.

[17] Sugiyama, K. and Misue, K. 1995 A Simple and Unified Method for Drawing Graphs: Magnetic-Spring Algorithm. In Proceedings of the International Workshop on Graph Drawing, (Princeton, NJ, USA, October 1994), GD'94, Springer-Verlag, Berlin, LNCS 894, 364-375.

[18] Tejada, E. and Ertl, T. 2005 Large Steps in GPU-based Deformable Bodies Simulation. Simulation Modeling Practice and Theory, 13(2005), 703-715.

[19] Yi, J.S., Melton, R. Stasko, J., and Jacko, J. 2005 Dust & Magnet: multivariate information visualization using a magnet metaphor. Information Visualization, 4 (2005), 239-256.

Agent Warp Engine: Formula Based Shape Warping for Networked Applications

Alexander Repenning
AgentSheets, Inc.
6560 Gunpark Dr. Suite D
Boulder, CO 80301
+1 303 530-1773
alexander@agentsheets.com

Andri Ioannidou
AgentSheets, Inc.
6560 Gunpark Dr. Suite D
Boulder, CO 80301
+1 303 530-1773
andri@agentsheets.com

ABSTRACT
Computer visualization and networking have advanced dramatically. 3D hardware acceleration has reached the point where even low-power handheld computers can render and animate complex 3D graphics efficiently. Unfortunately, end-user computing does not yet provide the necessary tools and conceptual frameworks to let end-users access these technologies and build their own networked interactive 2D and 3D applications such as rich visualizations, animations and simulations. The Agent Warp Engine (AWE) is a formula-based shape-warping framework that combines end-user visualization and end-user networking. AWE is a spreadsheet-inspired framework based on Web sharable variables. To build visualizations, users define these variables, relate them through equations and connect them to 2D and 3D shapes. In addition to basic shape control such as rotation, size, and location, AWE enables the creation of rich shape warping visualizations. We motivate the AWE approach with the Mr. Vetro human physiology simulation supporting collaborative learning through networked handheld computers.

Categories and Subject Descriptors
C.2.4 [Distributed Systems]: distributed applications, I.3.5 [Computational Geometry and Object Modeling]: Hierarchy and geometric transformations

General Terms
Design, Human Factors, Languages

Keywords
Real-time Image Warping, Collective Simulations, 3D Graphics, spreadsheets, End-User Programming, End-User Development.

1. INTRODUCTION
End-user computing, including end-user development [9] and end-user programming [13], is a quickly growing field with the number of end-user programmers already exceeding the number of professional software developers [7]. Some end-user development employs Web 2.0[1] frameworks, mostly for collaborative authoring and access to shared repositories of non-computational artifacts like images (Flikr), text (Wikipedia), and movies (YouTube).

End-user programming goes beyond the authoring of images, text, and movies by letting computer users without formal programming background create computational artifacts. Spreadsheets have been an extremely popular end-user programming platform. In the 1990s we witnessed a wealth of research that pushed the boundaries of the spreadsheet paradigm (e.g., Forms/3 [3], NoPumpG [20], Garnet [12], AgentSheets [14]). Most of these systems explored extending the existing number-and-string spreadsheet framework with notions of interactive graphics.

Video games and the Web have been essential drivers of the incredibly rapid evolution of personal computers. Since the 1990s, visualization and networking capabilities of affordable computers have exploded, yet very few of these advancements are accessible to end-user computing. Although spreadsheets use 3D technology such as OpenGL to display pie charts and 3D plots very efficiently, from a conceptual viewpoint, the state of the art has not changed much.

Perhaps more important than the availability of powerful technology, are the concrete needs we have encountered for a new end-user framework that is capable of creating sophisticated, interactive, networked visualizations. Educators presented us with the challenge of teaching about interacting complex systems such as human systems in physiology. As a response, we created a framework that goes beyond regular animations to create medical visualizations and networked simulations [16]. This framework is based on spreadsheet ideas in a way that provides what we call rich *end-user visualizations* and *end-user networking,* with the following requirements.

End-User Visualizations:

- ***End-User Accessible***: End users should not only be able to select from menus of preexisting visualizations such as plots, and simple geometric shapes, they should also be empowered to construct their own. In order to do this, they need to be able to

AVI'08, May 28-–30, 2008, Napoli, Italy.

[1] http://en.wikipedia.org/wiki/Web_2

make or import 2D or even 3D shapes and have the means to control these shapes so they can serve as visualization.

- ***Rich***. To be truly engaging, visualizations need to be rich. Many variables, e.g., physiological variables like heart rate, could be represented by numbers. However, to truly engage learners, such crucial variables should be able to immerse learners audio-visually. Ideally, this would go so far as to invoke an emotional response. A set of variables representing a hyperventilating human being should not just be a list of numbers, but should show a human being with heart and lung functions – even with sounds that provoke emotional responses.
- ***Efficient***. To be perceived as smoothly animated, visualizations need to be highly efficient. In the case of Mr. Vetro we need to be able to perform complex shape warping for the skeleton, heart, and lung in real time and at a frame rate of at least 30 frames per second.

End-User Networking:

- ***Transparent networking***: Similar to a spreadsheet, it should be possible to define computation as a set of variables connected with each other through formulae. Unlike the traditional spreadsheet idea, however, variables should be sharable through the Web. This mechanism should be as transparent as possible. That is, an end user should not have to use network APIs or use any other complex mechanism to implement variable sharing. All they should have to do is declare the variable to be shared when they create that variable.
- ***Real time computation***: To allow the building of sophisticated animations and simulations, it should include a simple model of time similar to the Forms/3 system [3]. Additionally, because of potential delays caused by networking and client implementation, animations and simulations need to be able to work in a frame rate independent way.
- ***Fluid***: End-user networking should exhibit low viscosity [6]. Similar to a spreadsheet, it should be simple for the user to add, change, and remove variables. The environment should facilitate experimental explorations of networked computing.

In the following sections we introduce the Agent Warp Engine technology with a basic example, describe Mr. Vetro as a more sophisticated medical application, compare related work, and share our evaluation results.

2. TECHNOLOGY

The Agent Warp Engine (AWE) is a technical framework for running collective simulations. It is implemented as a thin layer on top of the Open Agent Engine[2] (OAE) which itself is the open source part of the AgentCubes 3D game and simulation-authoring environment [15]. The OAE implements a simple 3D agent-based simulation engine based on four main components:

OpenGL: ***2D/3D Graphics.*** A highly optimized 2D/3D rendering API[3]. Fundamental graphics primitives such as 3D meshes, textures, and shaders [19] are hardware accelerated on most platforms. OpenGL is used to render large numbers of agents with 3D shapes efficiently.

QuickTime: Sounds, images, movies. An API available for OS X, Windows, and Linux[4] provides access to image, movie, and sound files. QuickTime is used to load texture files and to play sound files.

XMLisp: files, knowledge representations. XMLisp [17] is an API mapping XML expressions to object oriented language constructs. An AWE project consists of XML files that are processed through XMLisp.

3D Agents. Autonomous objects that have a 3D position, orientation, size, velocity, and acceleration. Agents can be composed into scenes, animated, displayed and user selected.

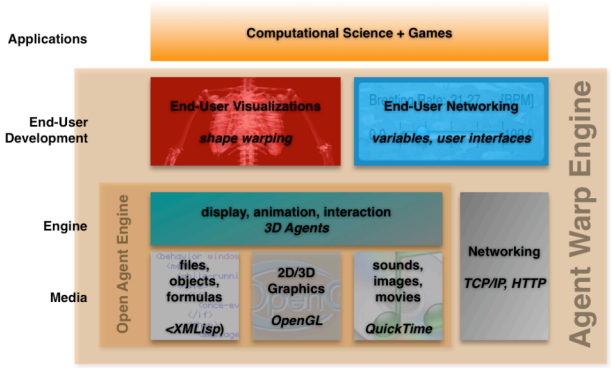


Figure 1. The Agent Warp Engine is layered on top of the Open Agent Engine

AWE adds two main components to the architecture (Figure 1):

- ***End-User Visualization***: end users create custom visualizations by defining 2D or 3D shapes with control points that connect to variables through spreadsheet-like formulas. Employing techniques such as shape warping, users can define sophisticated visualizations such as a beating human heart.
- ***End-User Networking***: an end-user formula language includes the notion of variables and equations connecting them. Variables on different clients can be shared via the network in real time. Network access, established via HTTP, is completely transparent to users.

The following two sections illustrate the notions of end-user visualization and end-user networking in detail.

2.1 End-User Visualization

An important goal of this work is to offer the refined kinds of visualizations necessary to communicate complex dynamic processes. For instance, in our main application called Mr. Vetro, we need to show the function of the heart, the lung, and the human skeleton. All three systems mechanically interact with each other in complex ways. For instance, inhaling air will change the shape of the lung, which in turn will influence the skeleton. Ribs expand and, in the case of deep breathing, even the position of the shoulders and arms can be influenced. AWE offers a number of visualizations, but the most sophisticated one (called "morph") is specifically designed to implement complex

[2] http://www.agentsheets.com/lisp/OpenGL.html
[3] http://www.opengl.org
[4] http://www.openquicktime.org

visualization based on shape warping. Shape warping is a kind of image warping [21, 22].

We start with a basic but detailed example applying a shape warp visualization to the well known image of Leonardo DaVinci's Mona Lisa. While this example is much simpler than the Mr. Vetro visualization described in the next section, it illustrates how even small, easy to program shape warps can have a major effect. It helps that the human perception system is highly sensitive towards emotional clues found in facial expressions.

Experts are still debating if Mona Lisa's facial expression represents sadness or happiness. We will not contribute to this discussion, but we will use Mona Lisa's image to illustrate how to build a simple visualization with AWE. We will use morphs as a means to warp Mona Lisa's image in a controlled way and evoke different emotional interpretations.

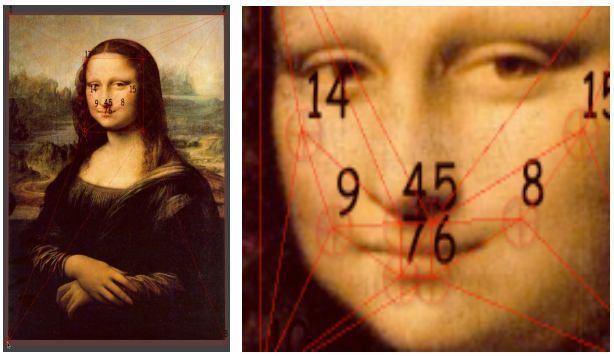


Figure 2: Left: Mona Lisa. Right: detail showing tessellated detail of her face.

An AWE morph can warp images using texture mapping [21]. Explaining the details of texture mapping goes beyond the scope of this paper, but in essence, a texture map is a means of applying a texture or image to a shape. A simple example would be to map a rectangular image such as the entire image of the Mona Lisa onto a rectangular shape. The pixels of the image need to be mapped accordingly, e.g., scaled vertically or horizontally depending on the shape dimensions. Things become substantially more interesting when the target shape is deformed in ways other than just vertical and horizontal scaling. For instance, one can tessellate an image, a process of tiling, by segmenting it into a triangle mesh. Mapping these triangles onto identical target triangles would create the original picture. However, one can warp an image by shifting vertices of this mesh. To render a warped image, texture-mapping needs to be able to warp individual triangles in terms of shape and their image content. This process is computationally intensive but can be done very efficiently in hardware on modern graphical processing units.

The main idea of our simple emotional visualization example is to add a single variable to control the entire spectrum from sad ☹, to neutral 😐, and on to happy ☺. Sophisticated facial expression systems have existed for some time. The Radial Basis Function network approach [8], for instance, uses 2D image warping based on neural nets. Here we want to show a much simpler approach that is readily accessible to end users.

The first step in defining our visualization is the image tessellation. AWE includes a mesh-authoring tool that lets end users define tessellation points and triangles. A simple approach to warping our image emotionally is to focus on Mona Lisa's mouth to make her look happy, sad or neutral. A mesh around her mouth (Figure 2, vertices 4, 5, 6, 7, 8, 9) is a starting point. Additional vertices are needed to be able to define triangles covering the entire image. The selection of vertices requires some experience. Users participating in our evaluation (section 5.2) were able to create a well-working mesh of a person that included warping of mouth and eyes after seeing one example. The key to vertex selection is controlling the scope of the desired effect. To change the mouth by moving vertices 4-9, one needs to make sure that the mouth deformation does not influence too much of the remaining image. For instance, if the only other vertices were the corners of the image itself, then moving vertices 8 and 9 up to make Mona Lisa smile would also partially move the rest of the face in an unnatural way. Instead, we define vertices 14 and 15 as fixed points, roughly at the location of the cheekbones.

To create a *formula-based shape warp* an end user needs to define a mesh and add formulas to vertices. The AWE mesh-authoring tool automatically creates an XML representation of the Mona Lisa mesh that includes the image reference, a list of vertices, and a list of triangles. Vertices 8 and 9 are the left and right corners of the mouth. Vertex 8 in AWE XML notation looks like this:

```
<vertex x="0.520" y="0.712" xt="0.520" yt="0.712"/>
```

X and Y are the vertex positions, whereas xt and yt are the texture coordinates.

The goal is to control Mona Lisa's emotions by adjusting the positions of the left and the right corners of the mouth. Both the x and the y attribute of the vertex are extended by the user from being constants to being formulas:

```
<vertex x="0.520 + 0.0003 * happiness" y="0.712 + 0.0003 *
   happiness" … />
```

Happiness is a user defined variable. For happiness = 0 we get the original image. For happiness > 0 we get an increasingly happy Mona Lisa by pulling her mouth corners up and out. Finally, for happiness < 0 we have her start to frown by pulling her mouth corners down and together.

The real power of a formula-based shape warp appears when the user sees it attached to a variable controlled by a slider and experiences warping in real time. Because of OpenGL hardware acceleration, this simple application with both visualization and formula evaluation runs at hundreds of frames per second—even on moderate hardware. Unfortunately, this experience cannot be completely shared in a static publication. However, Figure 3 shows some variations—including the original image. With this simple mesh and equations used in this example, the limits are visible. For instance, the super happy Mona Lisa no longer looks completely natural. However, the important point is that end users with no background in computer graphics can build formula-based shape warps effectively.

2.2 End–User Networking

End-user networking is based on shared variables that let end users create distributed applications. End users should be able to define, change, and connect these networked variables as easily as they use variables in spreadsheets. For our collective simulations, it is essential that these variables can be shared by different clients running on the same computer or separate networked computers.

For most of our applications, real-time interactivity is essential. In a collaborative simulation all users are physically present in the same room. Being able, as a group, to see how individual users change simulation variables in real time and how the system reacts to change helps the perception of causality [11]. Users

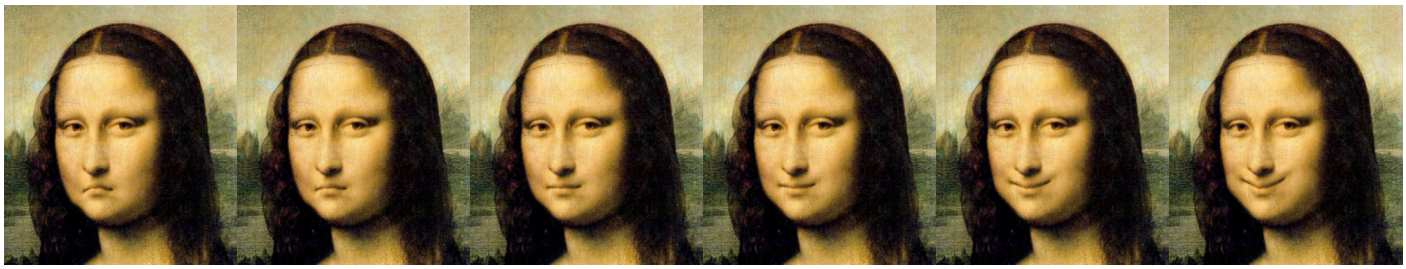


Figure 3: Shape warping of Mona Lisa. Depending on Happiness she is annoyed, concerned, innocent, enigmatic, happy, ecstatic.

typically change the value of variables gradually using slider interfaces to let themselves and everybody else in the classroom experience controlled change and the gradual reactive effect of all the other simulations that depend on a specific variable.

AWE supports *local variables* and *shared variables*. All variables have these properties:

- ***name***: all variable references are by user-defined names.
- ***value***: a value is defined as constant (e.g., value=3.14), a formula (e.g., value="sin(time) + 45.0") or through a user interface such as a slider (Figure 4).
- ***action***: each time the variable is changed an action could be invoked.
- ***user interface***: one or more user interfaces can be attached to a variable to output values or to get values from the user.

Local variables are the simplest kind of variables. They could be compared to cells of a spreadsheet. The scope of local variables is limited to a single AWE client. Other AWE clients may have local variables with identical names without establishing sharing.

Shared variables are used as communication mechanism between multiple simulations. Conceptually speaking, shared variables are similar to shared memory locations in parallel programming. A user simply defines a variable to be shared without the need to explicitly deal with network interfaces. For instance, if the variable heart_rate is defined to be a shared variable, then all clients can simply access that variable by referring to its name. For instance, the equation `sin(time * heart_rate * 6.28 / 60)` will access the current heart_rate set and shared by the heart client.

Typically only one client writes to a shared variable, but any number of clients can read a shared variable. Shared variable synchronization does not prevent multiple clients from writing to the same shared variable, but the shared variable will keep the value set by the last client computing that value. Shared variables are shadowed by local variables with the same name.

The sharing of shared variables is achieved through a Web server. Clients establish fast HTTP 1.1 persistent connections to a shared server that is storing and providing access to variable <name, value> tuples. Each client has a number of caches holding values of local and shared variables in order to minimize networking overhead. When a formula is evaluated, all its variable values are computed. If the value is not found in the right cache then it will be assumed that the variable to be accessed is a shared variable.

User interfaces allow users to control the values of variables. Mona Lisa's shape warp employs a formula that includes a single variable reference: `0.520 + 0.0003 * happiness.` Happiness was defined to be a local variable with a slider user interface. The XML representation

```
<local-variable name="HAPPINESS">
  <slider-interface min-value="-50.0" max-value="50.0"
    label="happy:" units="[haha]"/>
</local-variable>
```

corresponds to the slider in Figure 4, representing a local variable. As the user changes the value of the variable through the slider, the shape warp is recomputed and updated on the screen. Displaying the slider and the shape warp is fast; they render at about 400 frames per second on a 1.67 Ghz Mac PowerBook G4 with an ATI Mobility 9700 GPU.

Figure 4. Variable Happiness with slider user interface

User interface output options include sound. In the Mr. Vetro application (Figure 5) the lung distortion is computed as a function of time, breathing rate, and lung tidal volume. To increase the immersiveness of the visualization we added inhale and exhale sounds that are triggered if the value of the distortion variable begins to increase or to decrease respectively.

```
<local-variable name="DISTORTION" value="0.02 *
  lung_tidal_volume * sin(time * breathing_rate * 6.28 /
  60)">
  <sound-alert-when-value-begins-to-increase
    soundfile="inhale.mp3"/>
  <sound-alert-when-value-begins-to-decrease
    soundfile="exhale.mp3"/>
</local-variable>
```

Next we describe the Mr. Vetro application in more detail. This application was the main driver for developing formula based shape warping.

3. MR. VETRO: AN AWE APPLICATION

Mr. Vetro[5,6] (Figure 5) is a Collective Simulation for human physiology we have developed in collaboration with teachers and medical doctors for use in K-12 science classes [16]. *Collective Simulations* is a conceptual framework that integrates *social learning pedagogical models* with *distributed simulation technical frameworks*. This conceptual framework both enables and actively encourages meaningful learning by supporting a discovery-oriented social learning process. Visualization, animation and simulation as well as networking play a vital role in collective simulations. The first incarnation of Mr. Vetro (described below) incorporated these aspects, but our preliminary evaluation (section 5.1) led to making those aspects accessible to end-users. This

[5] Translated from Italian, "vetro" means "glass". The name is derived from Mr. Vetro's glass skeleton.

[6] An interactive flyer with Mr. Vetro can be found at: http://agentsheets.com/research/c5/documents/interactive%20flier/c5-flier.html.

process led to the development of the end-user visualization and end-user networking components of the AWE architecture.

In the Mr. Vetro application, different human systems are simulated on wirelessly connected handheld computers, while a central simulation aggregates parameters from the organs and computes Mr. Vetro's vital signs.

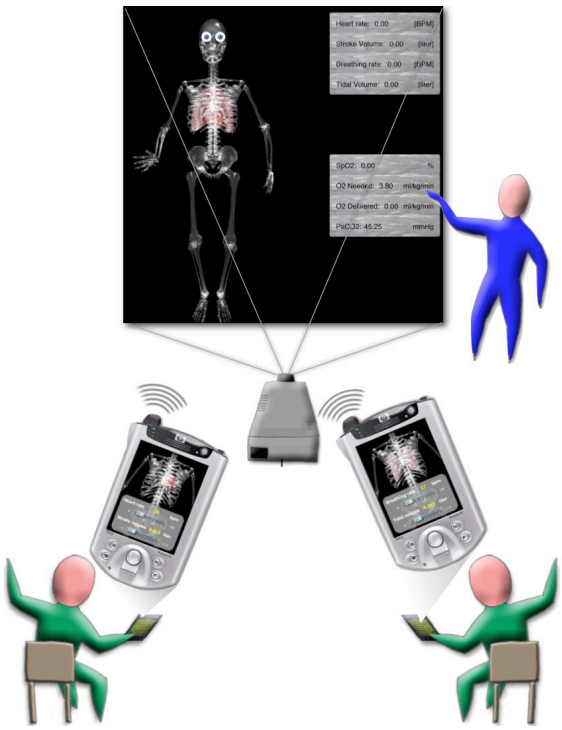


Figure 5: Mr Vetro – the collective simulation aggregating all the input parameters from the distributed organs and calculating vital signs.

In an activity, handheld devices with Mr. Vetro's organs are handed out to students. One group receives the heart, another group the lungs. Students engage in role-playing by being Mr. Vetro's organs. The lung group varies parameters such as breathing rate and tidal volume in response to changing conditions such as exercise or smoking. The heart group can vary parameters such as heart rate and stroke volume. Another group controls decisions such as how intensely to exercise.

A computer connected to a video projector is the server that communicates wirelessly with all the client simulations and aggregates all the inputs into a composite representation of a human that includes visual and audio elements. A vital signs monitor keeps track of Mr. Vetro's vital signs and displays them in the form of graphs or numerical values. Oxygen saturation in the blood, partial pressure of CO_2, oxygen needed and oxygen delivered to tissue are some of the physiological variables that are calculated and displayed.

Using Collective Simulations in the classroom is dramatically different from the typical use of technology in education. A collective simulation cannot be operated without discourse and collaboration among members of the same team and among different teams. This fosters a social style of learning that emphasizes distributed cognition [2].

3.1 Requirements

In a formal feasibility study conducted with the first prototype of Mr. Vetro (section 5.1), we found that collective simulations provide significant improvement over lecture-style teaching. This pilot study provided strong evidence for effective teaching performance and increased motivation. However, this real-world classroom use of the technology also revealed some emerging requirements that are being addressed in the next research phase.

Crude Animations => Improved Medical Content: Animations of the human were crude. Mr. Vetro's skeleton was not moving at all and the lung animation was anatomically wrong. These limitations were pointed out by both medical doctors and students during the evaluation study. Our limited animations needed to be replaced with more anatomically and physiologically correct animations that would be acceptable to medical specialists (doctors & medical illustrators). We therefore designed the AWE engine for end-user visualizations.

Networking challenges => Zero Configuration: The distributed nature of collective simulations presented non-trivial network configuration challenges. Establishing connectivity between clients, servers, and the Web proved to be difficult in educational settings. We found wireless networking to be especially complex for current-generation PocketPC PDAs. In educational settings teachers have no time to spend on elaborate technology configurations. Simplifying this process to the point of zero configuration will be essential to successful adoption of this technology. To simplify connectivity we developed a new approach. Instead of embedding the collective simulation in a server running in the classroom – as in Mr. Vetro I – which required client reconfiguration each time the network environment changed (e.g. typing IP addresses on each client), we designed a client server architecture where everything (individual organs as well as the entire Mr. Vetro) is a client to a Tuple [10] server. The server resides on a machine accessible from anywhere and does not require special configuration for the clients to access it.

Hard-coded system => Flexible authoring mechanisms: Extending or altering the activity to introduce more physiological variables was not easily achievable with the first system prototype. However, easy alteration and system flexibility is needed to extend the repertoire of interactive educational activities. This extension is commonly achieved by customizing existing activities and including new ones that focus on different aspects of human physiology. From the activity developer's point of view the system should have low viscosity (resistance to change) [6]. The effort required to change end-user defined artifacts should not be prohibitive. Instead, flexible mechanisms should let the activity designer create new activities easily. For instance, new organs and many physiological variables could be added to Mr. Vetro. Depending on the educational goals of each scenario, different organs and variables could be involved, but the students would see only a subset. Since human physiology is extremely complex and involves convoluted relationships of many organs and variables, it is often necessary to reduce the complexity to make it manageable for students to infer and understand relationships between variables.

3.2 Mr Vetro II

The need to address the needs uncovered by the feasibility study led to the design of the new Agent Warp Engine infrastructure for our collective simulation framework. End-user visualizations tailored to Mr. Vetro provide the complex animations needed for

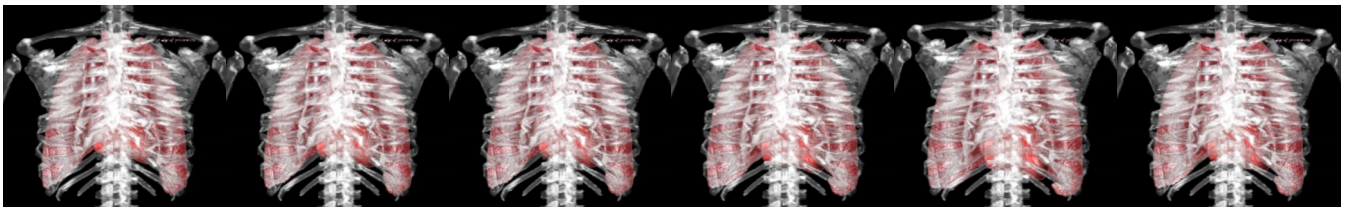


Figure 6: Sequence of frames in Mr. Vetro's animation: the visualization of breathing includes movement of the skeleton (notice the subtle shoulder location changing in each successive frame) and lungs filling up the space at the bottom when the diaphragm recedes to make room for them to expand. The visualization of the heartbeat is illustrated by subtle warping of the heart shape.

realistic and accurate medical applications. These visualizations feature formula-based shape-warping capabilities based on user-defined variables. They also enable flexible authoring of new organs of the human body. End-user networking for collective simulations addresses networking challenges by reconceptualizing the architecture and implementing every organ as a client to a Tuple server. It also enables flexible authoring of new activities with the introduction of new physiological variables.

Mr. Vetro II consists of the following shape warps:

Skeleton: A ray-traced shape with 16,000 polygons is used to create the X-ray look of Mr. Vetro's glass skeleton. The skeleton movement is animated based on a distortion equation that is a function of breathing rate and tidal volume. The effect is a realistic shoulder and ribs movement based on how fast, and how deep Mr. Vetro is breathing (Figure 6).

Heart and Lungs: The classic illustration of the heart and lungs is from Gray's Anatomy book [5]. The organs are both warped independently based on parameters of each organ, and also move synergistically to simulate the mechanical connection between the two organs in the body.

Mr. Vetro includes the following local and shared variables:

Heart parameters: Heart rate and stroke volume are shared variables of the distributed client simulation of the heart.

Lung parameters: Breathing rate and tidal volume are shared variables of the distributed client simulation of the lungs.

Activity parameters: exercise intensity expressed in running speed is a shared variable of the distributed client simulation of Mr. Vetro's brain.

Vital signs: O_2 saturation, partial pressure of CO_2, O_2 needed, and O_2 delivered to tissue as Mr. Vetro exercises are local variables computed based on equations referring to input parameters.

For continuous shape warping (having the animation/warp change in real-time) a special time variable is used in the distortion functions. For instance to simulate the motions of the human body associated with breathing, or the movement and distortions of lungs and heart as the breathing and heart parameters change, we need to express the warping of the vertices common between lung and heart as a function of two different frequencies, as well as time, e.g.:

```
<vertex x="0.735 + distortion_heart" y="0.120 - 4 *
   distortion_lung + distortion_heart" xt="0.735" yt="0.120"/>
```

where distortion_heart is defined as

```
<local-variable name="distortion_heart" value="0.2 *
   heart_stroke_volume * sin(time * heart_rate * 6.28 / 60)" />
```

The sequence of frames in Figure 6 show Mr. Vetro's skeleton, heart and lungs move according to the input parameters. Since it is difficult to capture animation on print media, we created a movie that conveys the full animation experience[7].

4. RELATED WORK

The AWE framework for formula-based shape warping for networked applications is a spreadsheet-inspired approach to end-user visualization and networking.

Since their early inception, spreadsheets and related paradigms, such as data flow, evolved to include visualizations. For example, in the ThingLab system [1], a constraint-oriented simulation laboratory was used for the development of interactive graphics. ThingLab supported the definition of networks of constraints, but because of the bidirectional nature of constraints, the behavior was hard to predict [20]. Moreover, there was no real end-user component: the non-graphical, programming-language notation for the constraints did not let the user author – there was a clear distinction between the "user" of the system and the programmer [20]. NoPumpG [20], a system that combined interactive graphics with spreadsheets, attempted to introduce more end-user authoring aspects to the definition of visualizations (behavior and appearance). Forms/3 [3] extended visualization capabilities with a notion of time used to create animations and simulations.

Networked spreadsheets expanded the spreadsheet paradigm with networking capabilities. Spreadsheet-inspired tools such as WikiCalc[8], a web application that let users share spreadsheets through wiki-style user-editable interfaces, enabled new kinds of collaboration through end-user networking. However, WikiCalc does not include sophisticated visualizations, and more importantly, is not geared for real-time variable sharing.

The AgentSheets Behavior Exchange [18] was an early incarnation of Web 2.0 ideas combining end-user visualizations with end-user networking. It let end users author and share computational components, called agents, through web interfaces. Used mostly for educational applications, the Behavior Exchange let end users, such as children in science education, learn about the fragility of ecosystems by creating and sharing electronic versions of behaving animals. In contrast to text, photos, and movies, the agents that were created with this kind of technology included end-user defined behavior. Today, we see early versions of more sophisticated forms of end-user programming that include the use of web services to execute programs useful to end users.

Mashup platforms such as Yahoo Pipes[9] or Orchestr8[10] are an exciting development that bring web-based computing to end

[7] Movie: http://www.youtube.com/watch?v=39NJJC1Vt18. Alternatively go to YouTube (http://www.youtube.com) and search for Mr Vetro.

[8] http://en.wikipedia.org/wiki/WikiCalc

[9] http://pipes.yahoo.com

[10] http://www.orch8.net

users. End users can create their own mashups by running, inspecting, and editing shared scripts. Yahoo Pipes, for example, lets end users define scripts called pipes to access and process web based information available in RSS format.

5. EVALUATION

Evaluation of AWE technology has occurred at multiple levels. The educational efficacy of the collective simulations approach was evaluated in a formal feasibility study in high school biology classes; the usability of the end-user visualization approach of AWE has been evaluated in an undergraduate Computer Graphics and Visualization class; and end-user networking is currently being evaluated as we work with teachers and content experts to create new activities for physiology classes.

5.1 Educational evaluation

To evaluate the feasibility of the Collective Simulations framework with Mr. Vetro, we designed an experiment comparing a traditional lecture format with an interactive activity using the Mr. Vetro collective simulation. We worked extensively with teachers and doctors to create two complete alternative learning activities aligned with state-level science standards to use in the evaluation study. We employed an improvement-measuring framework [4], to assess the advantages of educational technology. The framework looks for evidence of four different types of improvements: *increased learner motivation*, *advanced topics mastered*, *students acting as experts do*, and *better outcomes on standardized tests*.

We considered the feasibility study to be a success if in the collective simulation condition group we measured performed equal to or better than the lecture group, and we saw evidence for increased student motivation, mastery of advanced topics, and ability to act as experts. Along the four evaluation dimensions the results were as follows:

Learner motivation: Motivation is essential and at the core of this research. We strongly believe that without motivation much of the educational effort is lost. Students in the lecture group were not particularly engaged in the material being presented. In comparison, in the Mr. Vetro group, students were noticeably engaged. The teacher asserted that students stayed completely focused on task and more students were engaged speaking, listening, and asking questions than in any lecture session and most lab sessions.

Mastery of advanced topics: Mastery of a topic is not limited to memorizing facts, but includes the ability to gain deeper understanding of knowledge. The Mr. Vetro groups scored significantly higher (26% higher) than the control group on the deep knowledge questions posed on the retention test. In order to answer these questions, students had to have a strong sense of the interaction of human organs as they had to reason quantitatively about physiological variables at settings they had not experienced during either the lecture or the simulation.

Learners acting as experts: Developing the ability in learners to use problem solving processes similar to those of experts is challenging, but provides powerful evidence that students are gaining the skills they will need to succeed. The inquiry-based approach to pedagogy creates a need to work and communicate in groups and substantially changes the roles of students and teachers. Students engage in cause and effect questions. Rather than behaving in the typical student role of absorbing facts and vocabulary, Mr. Vetro students were grappling with ideas and trying to find answers to their own questions. When students expressed a misguided interpretation of the current situation, other students intervened to correct the misconceptions. Students who normally did not participate in class took on leadership roles in their groups.

***Performance on conventional tests*:** The most difficult type of evidence to provide for the superiority of new, technology-based instructional models is higher scores on conventional measures of achievement. There is no standardized test for human physiology at the high-school level. Instead, we used teacher-generated tests that were guided by the biology curriculum and standardized science tests. Comparing performance scores (total points on the retention test) of both groups we found a slight, insignificant ($p < 0.05$) advantage in the collective simulation condition. However, we found that the collective simulation group had a significantly lower pre-test average score compared to the lecture group. Therefore, in terms of learning gain (ratio of retention score to the pre-test score) we found a significant advantage for the Mr. Vetro group: this group showed a 58.88% learning gain compared to a 50.87% learning gain of the lecture group.

This formal feasibility study provided early indications of the educational efficacy of the collective simulations approach for teaching about the relationships of complex interacting systems.

5.2 End-User Visualization evaluation

The idea of formula-based shape warping was evaluated with a group of undergraduate informatics students with no background in computer graphics. In a Computer Graphics and Visualization class at the University of Lugano in Switzerland, the task was to create a shape warp of faces similar to the Mona Lisa example presented in this paper. In addition to making the faces appear to smile or be sad by warping the mouth, students had to warp the eyes. Moreover, the faces had to be able to open and close their eyes without squeezing the eye pupils. This required a more sophisticated approach of having two layers of images. One layer was the image of the person with the eye masked out, and the other layers, behind the first one, were the eyes as two separate images. All students in this experiment were able to create a running version of the face morph application.

5.3 End-User Networking evaluation

We are currently working with biology teachers and doctors to add new systems to Mr. Vetro. Adding the renal system (the kidneys) would allow us to explore scenarios of blood loss, dehydration, and training in high altitude. For instance, Mr. Vetro could be in situations where blood pressure changes dramatically and needs to be regulated. To accomplish this, the kidneys need to produce Renin to start the chain reaction of turning Angiotensin I to Angiotensin II; Angiotensin II ultimately regulates blood pressure. In these scenarios, blood pressure could be an input to the system. In other scenarios, like subjecting Mr. Vetro to exercise, the blood pressure would be an output that is calculated and visualized for the students. The new AWE architecture allows such variations. Two representations of Mr. Vetro could exist for each scenario with different XML representations of the blood pressure variable – one defined as a shared variable with user input interface (e.g. a slider), and one as a calculated output.

6. DISCUSSION

Our current method of developing these activities involves collaborating with content experts to get enough information to produce the code (XML representations) for the distributed

components of the collective simulation. Our ultimate goal is to give these experts the tools they need to define the system interactions with AWE by themselves. Most of the XML expressions needed to specify a formula based shape warp is automatically generated with the authoring tool. However, it is still necessary to find the proper locations for relevant vertices and to add formula expressions. Surprisingly, we found that the subjects in our study had few problems editing XML files. In part, we can explain this by end users having an increasing familiarity with XML and HTML file formats. Nonetheless, syntactic challenges due to XML editing can and should be eliminated. For this one could use XML syntax supporting editors or better yet visual programming languages generating XML.

Advances in computer graphics and mobile computing make it possible to create a new kind of networked application with strong visualization, animation and simulation components. The Agent Warp Engine not only combines sophisticated 2D/3D visualizations and real-time networking but also makes them accessible to end users. Formula based shape warping is a spreadsheet inspired end-user programming paradigm that can be employed for a variety of applications in need of end-user visualizations and end-user networking. In addition to the technological framework itself we have presented end-user visualization and end-user networking aspects of the Mr. Vetro human physiology collective simulation.

7. ACKNOWLEDGMENTS

This work is supported by NIH Grant 1R43 RR022008-02. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Institutes of Health.

8. REFERENCES

[1] Borning, A. 1981. The Programming Language Aspects of ThingLab, a Constraint-Oriented Simulation Laboratory. ACM Transactions on Programming Languages and Systems (TOPLAS), 3, 4, 353 - 387.

[2] Brown, A.L., Ash, D., Rutherford, M., Nakagawa, K., Gordon, A. and Campione, J. Distributed Expertise in the Classroom. in Salomon, G. ed. Distributed Cognitions, Cambridge University Press, New York, 1997.

[3] Burnett, M., Atwood, J., Djang, R., Gottfried, H., Reichwein, J. and Yang, S. 2001. Forms/3: A First-Order Visual Language to Explore the Boundaries of the Spreadsheet Paradigm. In Journal of Functional Programming (March), 155-206.

[4] Dede, C. 1996. Emerging technologies and distributed learning. American Journal of Distance Education, 10, 2, 4-36.

[5] Gray, H. Anatomy of the Human Body. LEA & Febiger, Philadelphia, 1918.

[6] Green, T.R.G. and Petre, M. 1996. Usability Analysis of Visual Programming Environments: a 'cognitive dimensions' framework. Journal of Visual Languages and Computing, 7, 2, 131-174.

[7] Jones, C. 1995. End-user programming. IEEE Computer, 28, 9, 68-70.

[8] Leung, M.Y.Y., Hung, Y.H. and King, I. 1996. Facial expression synthesis by radial basis function network and image warping. In Proceedings of the IEEE International Conference on Neural Networks (June 3-6). IEEE, Washington, DC, USA.

[9] Lieberman, H., Paternò, F. and Wulf, V. (eds.). End User Development. Springer, 2006.

[10] Matsuoka, S. and Kawai, S. 1988. Using Tuple Space Communication in Distributed Object-Oriented Languages. In OOPSLA '88. ACM Press, San Diego.

[11] Michotte, A. The perception of causality. Methuen, Andover, MA, 1962.

[12] Myers, B., Hudson, S.E. and Pausch, R. 2000. Past, present, and future of user interface software tools. ACM Transactions Computer-Human Interaction, 7, 1, 3-28.

[13] Nardi, B. A Small Matter of Programming. MIT Press, Cambridge, MA, 1993.

[14] Repenning, A. and Ioannidou, A. 2004. Agent-Based End-User Development. Communications of the ACM, 47, 9, 43-46.

[15] Repenning, A. and Ioannidou, A. 2006. AgentCubes: Raising the Ceiling of End-User Development in Education through Incremental 3D. In IEEE Symposium on Visual Languages and Human-Centric Computing 2006 (September 4-8). IEEE Press, Brighton, United Kingdom.

[16] Repenning, A. and Ioannidou, A. 2005. Mr. Vetro: A Collective Simulation Framework. In ED-Media 2005, World Conference on Educational Multimedia, Hypermedia & Telecommunications. Association for the Advancement of Computing in Education, Montreal, Canada.

[17] Repenning, A. and Ioannidou, A. 2007. X-expressions in XMLisp: S-expressions and Extensible Markup Language Unite. In Proceedings of the ACM SIGPLAN International Lisp Conference (ILC 2007). ACM Press, Cambridge, England.

[18] Repenning, A., Ioannidou, A., Rausch, M. and Phillips, J. 1998. Using Agents as a Currency of Exchange between End-Users. In Proceedings of the WebNET 98 World Conference of the WW, Internet, and Intranet. Association for the Advancement of Computing in Education, Orlando, FL, 762-767.

[19] Rost, R.J. OpenGL Shading Language (2nd Edition). Addison-Wesley, 2006.

[20] Wilde, N. and Lewis, C. 1990. Spreadsheet-based Interactive Graphics: From Prototype to Tool. In Proceedings CHI'90. ACM Press, Seattle, WA., 153-159.

[21] Wolberg, G. Digital Image Warping. IEEE Computer Society Press, 1994.

[22] Wolberg, G. 1996. Recent Advances in Image Morphing. In Proceedings of the 1996 Conference on Computer Graphics International. IEEE Computer Society, Washington, DC, 64.

Image Geo–Mashups: The Example of an Augmented Reality Weather Camera

Jana Gliet
Institute for Geoinformatics
University of Münster
Robert-Koch-Str. 26-28
48149 Münster, Germany
gliet@uni-muenster.de

Antonio Krüger
Institute for Geoinformatics
University of Münster
Robert-Koch-Str. 26-28
48149 Münster, Germany
antonio.krueger@uni-muenster.de

Otto Klemm
Institute for Landscape Ecology
University of Münster
Robert-Koch-Str. 26-28
48149 Münster, Germany
oklemm@uni-muenster.de

Johannes Schöning
Institute for Geoinformatics
University of Münster
Robert-Koch-Str. 26-28
48149 Münster, Germany
j.schoening@uni-muenster.de

ABSTRACT

This paper presents the general idea of image geo-mashups, which combines concepts from web mashups and augmented reality by adding geo–referenced data to a perspective image. The paper shows how to design and implement an augmented reality weather cam, that combines data from a steerable weather cam with additional sensor information retrieved from the web.

Categories and Subject Descriptors

H.1.2 [**User Machine Systems**]: Miscellaneous

Keywords

Augmented Reality, Geo–Mashups, Image Composition Processes

1. INTRODUCTION

Until recently the world wide web was made up of pieces of information without explicit reference to locations or the spatial context. Implicit information, e.g. indicated by the language, the IP-address of the server hosting that information and coordinates on the web page, has been present since the beginnings of the web. This information was intended to be used by humans and therefore rather difficult to parse. In the last years, the potential of the geo–referenced web has become apparent. Services such as yellow pages (`http://www.yellowpages.com/`) use geo–referenced content to provide users with information on shop locations. Directions on the web are more and more given in standardized ways, usually by providing a link to one of the numerous web mapping services, e.g. with Google Earth (`http://earth.google.com`), Google Maps (`http://maps.google.com`), Microsoft's Live Search Maps (`http://maps.live.com`) or Yahoo Maps (`http://maps.yahoo.com`) the potential of connecting content on the web through locations has become obvious [1]. Through simple API, mapping mashups can easily be created by integrating individual features, routes and other content (such as images) into an existing digital map. Such 2D Mapping mashups can be considered the first step of geo-mashups, i.e. mashups that use geo–referenced information that is available on the web. Google Earth already provides a sophisticated platform for 2.5-3D mapping geo-mashup, This is exemplified on the right hand side of figure 1. It is notable that mapping geo-mashups just require a weak reference to location of the involved media. By weak refer-

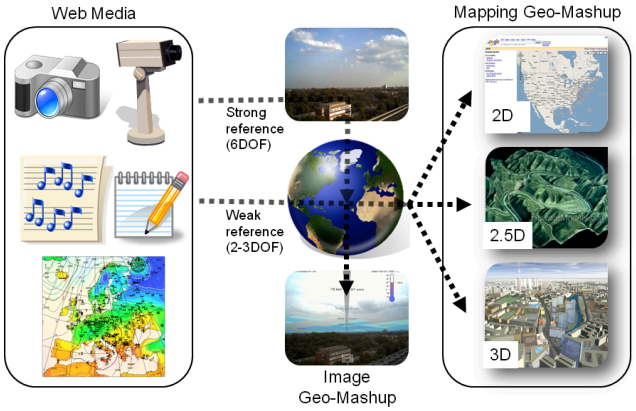


Figure 1: Differences of mapping geo-mashups and image geo-mashups. While mapping mashups just require a weak reference to location (i.e. 2-3 degrees of freedom), image mashups require a strong reference to location and position (i.e. 6 degrees of freedom and more).

AVI '08, 28-30 May , 2008, Napoli, Italy.

ence we mean 2 or 3 degrees of freedom (Dof), i.e. latitude, longitude and height. This paper will explore the next level of geo-mashups: image geo-mashups. These geo-mashups do not use maps as underlying reference graphics, but arbitrary images of locations and places. In order to integrate georeferenced material from the web it is necessary to georeference each pixel in the image. This requires obtaining information on at least 6 degrees of freedom of the camera while taking the picture, i.e. longitude, latitude, height, pitch, roll and yaw. Furthermore information on the optical camera parameters is needed to account for the lens distortion. With such information web content can be integrated directly into the image, and the result is very similar to an Augmented Reality (AR) [3] application, where virtual content is combined with an image of the real world.

The middle column of figure 1 shows an example of an augmented reality weather cam that we will use throughout the paper to illustrate our ideas. Here an image of a location is overlaid with additional geo–referenced information on the weather conditions, such as rain radar data and temperature distributions. In this example rain radar data is integrated into the sky directly, providing users a good overview on the actual rain probability. Additional information on the actual weather at the location and the direction of the camera are integrated in an iconic style as well.

Geo-mashups rely on static and on dynamic web content. Most of the mapping geo-mashups for example combine static content, such as background cartographic material, a street network layer, and selected locations. A few start to integrate dynamic content as well, such as online traffic data or weather information. The Image geo-mashups we would like to address in this paper mostly use dynamic content, i.e. video live streams and online weather data. Of particular interest to us are image geo- mashups that are able to compose aspects of the mashup automatically. Our web cam geo-mashup example relies on an image that is retrieved from a steerable camera. The mashup service itself decides automatically on the heading of the camera dependent on weather data to ensure that the camera is always pointing into the weather direction. Dynamic time and space scales which are integrated into the image, allow users to estimate how far away a rain front is located. From a practical perspective image geo-mashups will develop their full potential, when based on standardized web services. As demonstrated in this paper this allows the implementation of a complex web service that augments an arbitrary web cam image with geo- (and image-) referenced weather data, if the optical and positional parameters of the camera are provided. We will explain in this paper the main concepts of image geo-mashups and then show in detail how to design and fully implemented a weather cam geo-mashup. In the next section we will discuss relevant pieces of literature and explain how our work differs from related areas such as AR. Section three is dedicated to the general concept of image geo-mashups. In section four and five we will introduce an example of an image geo-mashup service: the augmented reality weather cam. We will conclude in section six and discuss open issues and future work.

2. RELATED WORK

Our work is clearly related to the field of Mixed Reality [7] and since a real environment is augmented more specifically to Augmented Reality (AR). In this context the subfield of mobile AR is of particular importance. One of the most prominent and earliest projects of this research field has been the Touring Machine from Feiner [9]. This system delivered mobile augmented reality information related to landmarks on Columbia University Campus. For this purpose digital information was overlaid in real time by see-trough glasses worn by the user. A couple of further mobile AR systems have emerged since then, providing indoor guidances [10], x-ray vision [11, 8] or tourist information [12].

Also related to our work is the concept of see-trough tools, introduced by Bier et al [13]. A special case of see-trough tools are magic-lenses that provide a convenient way to superimpose and interact with dynamic digital information on an underlying static information layer. Several research groups have looked at magic-lenses in the context of cartography as well.

Our own work Wikeye [14, 15], follows a similar principle, but is a true see-trough magic lens approach since it uses a typical mobile phone, a Nokia N95, to be operated above a physical map. The system tracks the 3D position [16] of the camera-display unit over the map and therefore allows to augment the video stream with additional information. This enables users to personalize the content of the static map through their camera display unit, for example, by adding personalized route instructions to the map. With our work on a Virtual Globe 2.0 in [17] we focus how content for such AR applications can be automatically derived from the web.

The most related approach to our work, at least from the technical setup, is Mower's work on Augmented Scene Delivery Systems [6]. He describes a system that augments the image from a steerable camera with labels describing the features in sight. To be able to position the labels correctly in the image a digital elevation model of the environment is used, together with a database of interesting landmarks that should be labeled. Mower worked on a couple of relevant problems related to the visibility of far away objects and their appropriate labeling. However, in this work he focuses on discrete data (e.g. symbols and labels) and not on continuous data (e.g. rain or temperature data). This makes the task of superimposing data over an image much easier, i.e. since occlusion of real world objects is playing a subdominant role.

A number of researchers have investigated the use of virtual environments with GIS data like with the ARVino [18] system or the Augurscope [19]. A good overview of the usage of AR for geographic visualization is provide by [20]. Schall et al. [21] have developed a handheld AR setup combining multiple sensors built around an Ultra-Mobile PC to visualise subsurface infrastructure in 3D extracted from geodata sources.

Another example of related work is the work of Brenner et al, who present a see-through device for geographic information [22]. Their focus lies on achieving sub-centimeter accuracy and is not inspired by the idea of geo-mashups. The Timescope (`http://www.timescope.de`) is a commercial application that combine binocular optics and a live webcam. The viewer can enjoy a 120° panorama view of the city. By hitting a special button she can experience a journey through different historic photos of that area. In this application the focus lies on the content which is presented to the user with 2 degrees of freedom. Related in a broader scope is the work from Naaman et al. [23], who retrieve additional context information from web sources related to

digital photos based on a location and time stamp.

The concept of image geo-mashups that we present in this paper relies on the principle of normal web mashups, geo–referenced data and concepts from AR. The main idea of web mashups is to combine the result of existing web services based on standard web protocols, such as HTTP. The result is a new web page that takes advantage of the synergy of the different data sources [4]. It is important to note that mashups rely on standards. Only standards allow easy reconfiguration of content and the adaptation into a new context. Therefore with the raise of the Web 2.0 framework mashups have recently become very popular. In the next section we will explain how to extend the notion of web mashups to image geo-mashups.

3. CREATING IMAGE GEO–MASHUPS

In this section we will discuss our understanding of the image geo-mashup process as briefly introduced in section 1. Figure 2 gives an overview on the main components of an image geo-mashup process. We distinguish three different layers involved to obtain an image geo-mashup.

The first level is responsible for the acquisition of the various pieces of data and is therefore referred to as the Sensor Level. Sensors can be of various kinds. For a weather meshup for example this could include physical sensors, e.g. temperature, air pressure and humidity sensors as well as virtual sensors that provide information about a state in the future, e.g. a weather forecast. Sensors can be more ore less complex. A temperature sensor for example can be considered rather simple, while a weather forecast sensor could be very sophisticated if it makes use of a complex weather model. For our approach it is just important that the sensor level provides the data within a reference-frame which allows to geo-reference every data item and therefore to combine various data sources. Although an image geo-mashup always relies on an underlying image, it is not necessary that one of the sensors is a camera providing a stream of real-time data. It is also possible to construct image geo-mashups with historic data or a canonical background image or even with a non photo-realistic representation of a landscape, provided that some sort of geo-referencing is possible.

The second level contains the Web Services that encapsulate the sensor data in a standardized way. These web services are both responsible to retrieve data from the sensors as well as to configure the sensors in the most appropriate way. This could include for example services that poll temperature sensors in constant times periods or a service that directs a steerable camera into a desired direction to obtain an image of a specific scene. It is important to highlight the importance of standards at this intermediate level. As pointed out in the introduction general mashups require standards and this is of course also true for geo-mashups. In the next section we will discuss in more detail which standards are suitable for geo-mashups. As with the sensor level also the web service level can consist of services with varying complexity. A service could be a simple wrapper, which ensures that the syntax of a particular standard is met or a complex service that integrates different types of sensor data in a geo- referenced manner. If web services provide geo–referenced data we refer to them as geo-services.

The third level is the Application level, where the final image composition takes place. Depending on the application in mind different web services are parameterized and triggered. The background image, which itself can be obtained by a web service or retrieved from a database, is blended with the results from the involved geo-services. We suggest that on the application level each result of a geo-service is transformed into image coordinates and stored in respective geo-layers. This makes it easier to combine thematic data, as provided for example by a weather radar, with the underlying image. On the application level the designer of an image-geo-mashup is confronted with the following problems: (a) the service selection problem, (b) the layout problem, (c) the registration problem. Problem (a) is highly task dependent, but still complex, i.e. if the mashup application needs to decide on the fly how to compose appropriate geo-services. Problem (b) is related to the challenge to decide where to place data from the geo-layers on the background image and, maybe even more important, where to keep the parts of the background image unaltered. As we will discuss in the next section, our example of an augmented reality weather camera places most information in the sky region of the image, i.e. to keep most of the landmarks as visible as possible. However, to find (semi-automated) ways to solve problem (b) is far from being trivial. The registration Problem (c) is the classical problem of AR, i.e. how to register the virtual information with the data of an image of the real world. The registration problem is often not as forward as it might appear. Even if the camera parameters (such as position and focal length) are known, additional information on the camera environment needs to be modelled with enough accuracy. If certain information cannot be extracted from the environment it is often necessary to work with simplified assumptions, for example values for the maximum visibility or the height of the clouds in an image. In the following section we will describe a particular weather image geo-mashup

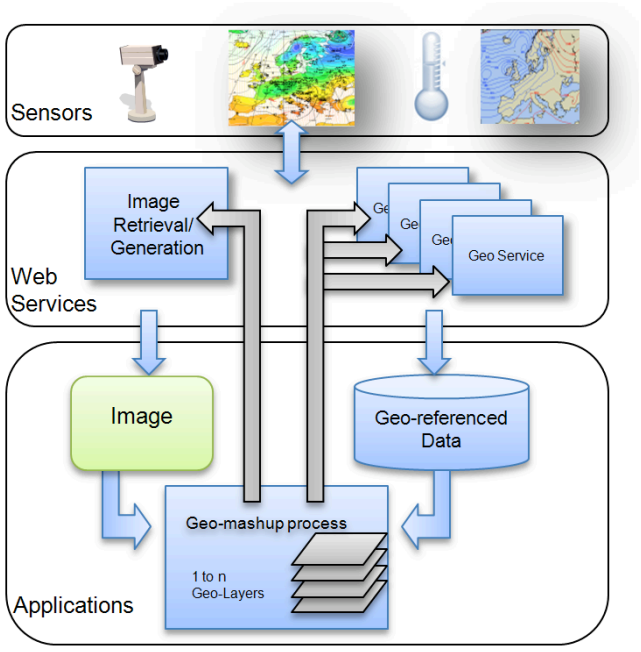


Figure 2: A schematic view on the image geo-mashup process. Three levels can be identified: the sensor level, the web service level and the application level.

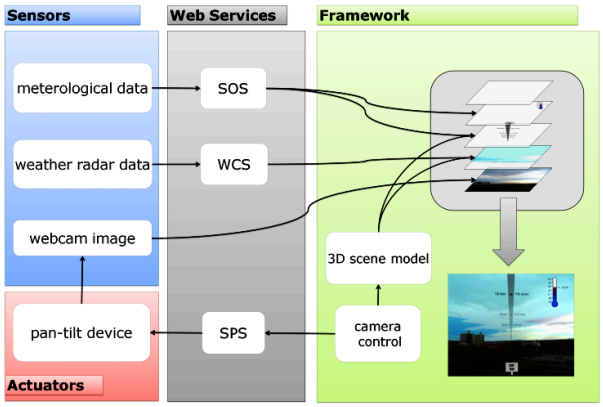


Figure 3: Communication structure generating an augmented weather image geo-mashup.

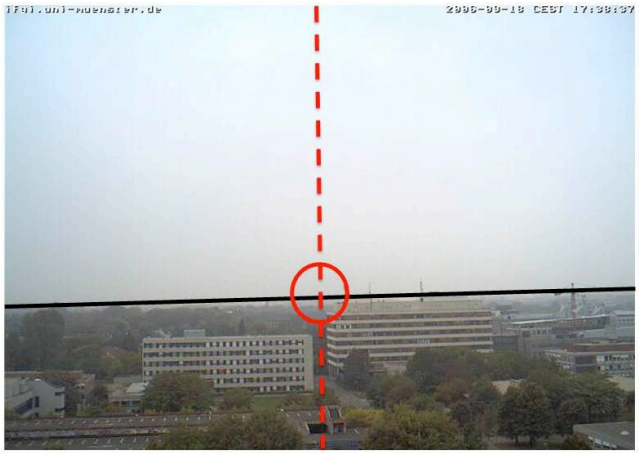


Figure 4: Determination of the theoretic horizon line.

that we have designed and implemented to test the concepts related to image geo-mashups.

4. THE AUGMENTED REALITY WEATHER CAMERA

The Augmented Reality Weather Cam (AR weather cam) is a software framework to create a mashup of a webcam image and additional spatial and textual data such as geo-data (e.g. data retrieved from a rain radar). The main idea of the AR weather cam is to combine a real-world image with sensor data which is visualized in the sky part of the camera image. Depending on the actual weather condition, this sensor data could stem from a weather radar, indicating areas of high rain probability or areas of temperature change. By geo-mashing this information into the sky of the weather camera, users can easily make a straight forward spatial reference between the geo-data and the environment. Since the AR weather cam can be automatically steered into the weather direction, (i.e. the direction where the weather comes from), users can be helped to judge how far away certain weather events from the actual stand point of the camera are. This image geo-mashup can be interpreted as a local short term weather forecast, which can help users for example to make a decision when to leave their offices without getting caught in a rain shower.

4.1 System Component and Architecture

The system consists of three major parts forming a framework to mash up a camera image as a geo- mashup (see figure 3). One part of the framework deals with the communication and data retrieval from external web services. In order to support the exchangeability of the used web services and allow an easy reconfiguration, we have decided to use standardized web services. As the data to be collected is geographical data, the web services used are based on standards defined by the Open Geospatial Consortium (OGC). The OGC is an international consortium of industry, academic and government organizations developing specification and standards to support the electronic interchange of geospatial data. A more detailed description of the web services used by our framework is given in section 4.4.

Another part of the framework deals with the connection between real and virtual camera which is essential to generate correct overlays. In order to combine real and virtual images, the virtual camera has to adopt the intrinsic and extrinsic parameters of the real camera. To calculate these camera parameters there are several approaches, one of them is described by Tsai [24]. One essential parameter (tilt of the camera or the pitch angle) may be difficult to specify. Our idea is to calculate the pitch angle with the knowledge of the theoretic position of the horizon line in the camera image. However, in some situations it may be difficult to calibrate a webcam (as described in [24]) to get the extrinsic orientation parameters roll, pitch and, jaw when the camera is difficult to reach due to the fact that it has been mounted on a pole or a roof. In this case an approximate calculation of the pitch angle (tilt of the camera) is possible when the intrinsic parameters and the y-coordinate of the theoretic horizon line in the camera image are known. The y-coordinate of the theoretic horizon's line is received from the camera image by taking the y-coordinate of the intersection point from the horizon line and the vertical center line as shown in figure 4. If the tilt of the camera is 0, the value of the y-coordinate is exactly half of the image height.

Then, to adapt the geo-data to the perspective image a 3D model of the environment is necessary. The construction of this 3D-model forms another part of the framework. The coordinate system in 3D space relies on the spatial reference system of the spatial data. If several spatial reference systems are used, the coordinates have to be transformed into one main reference system.

4.2 Geo-referenced AR Image Composing

According to the proposed general model for image geo-mashups presented in figure 2, the visualization process is based on a layered structure where each layer represents certain sensor data retrieved from external web services. The camera image forms the basis of the resulting overly image and is therefore placed at the bottom of the layer stack. The other layers contain edited sensor data (e.g. weather radar data, temperature) or additional information (e.g. direction of view). Sensor data consisting of 2D geo–referenced data is visualized by the means of a 3D sky model. We have

experimented with three different types of sky model that differ mainly by their degree of realism. For all models we have used a simple plane in the form of a rectangle specific at a certain height above the scene.

In the first approach the sky plane and the data to be displayed have the same spatial extension so that no additional distortions occur. The extent of the sky plane corresponds to the theoretic visibility (range of sight) which depends on the supposed height (static) of the sky and the height over ground of the real camera. Since the data is geo–referenced it can be easily added to the camera image. If the real value for the height of the clouds and the visibility are known, this approach can be extended by dynamically adapting the sky model to these real values at runtime. This leads to a virtual model which corresponds to the real (world) conditions represented by the camera image. However the required values for height of clouds and visibility are often difficult to obtain directly in real time and thus require a quite complex weather model, in order to provide a correct estimation. As in the first approach, the spatial extent of the 3D sky model matches the extent of the displayed data. In the third approach - as described in the first approach - the sky plane is modeled with a fixed extent and at a fixed height, but the spatial extent of the displayed data is larger than the extent of the 3D sky plane. This leads to an extended view, where more data is displayed on a fixed area, but the real camera image and visualization do not (longer) correspond, because of additional distortions of the areal data.

In order to be able to interpret the displayed 3D data correctly it is necessary to have a scale showing spatial and / or temporal distances. If the scale is modeled as a 3D element, it has to rely on the same 3D model as the displayed data to obtain the same perspective distortions. Besides the camera image layer, the data layer containing the 3D view of the 2D geo-data and the scale layer, other layers could be created containing elements to augment the underlying image. These elements can consist of textual data containing additional information to the used sensor data (e.g. date / time of retrieval, direction of view) or graphical elements such as compass rose for better spatial orientation.

To generate the resulting overlay image, the layer stack is evaluated in a bottom up order by combining just two layers at one time. This implies that the combination process starts with layer 1 and 2 and goes on combining the result of this combination with the following layer 3. This combination process is continued until the layer on top of the layer stack has been reached. The combination of two layers is based on a composite operation taking into account both source and alpha values of the images and combining them with the following function:

$$a \cdot A + (1 - a) \cdot (b \cdot B) \quad (1)$$

where A, B refer to the source colors and a, b to the alpha values of the images to be combined. Calculating alpha values for each layer is one of the essential parts during the visualization task. This concept makes use of a transparency mask and has the advantage of being very flexible, because every transparency mask is individually created for each layer. With a transparency mask, particular regions could be defined where geo-data should be displayed with a particular alpha value or where no data should overlay the underlying image. For the AR weather cam geo-data should mainly be displayed in the sky part of the camera image and the corresponding transparency mask has therefore to differ between sky and no-sky regions. Ideally, this mask should be created dynamically by analyzing the current camera image, especially if there are a lot of possible camera positions. If there are only a limited number of camera positions, static transparency masks should be preferred which can be created during a pre-processing step. The advantage of using static masks instead of dynamically generated ones is the independency from weather and light conditions, which can easily complicate the process of automatic sky detection.

4.3 Implementation

The camera we use is a webcam combined with a user controllable pan-tilt device purchased by Mobotix AG (`http://www.mobotix.com`). The camera is currently mounted on the roof of an 8-story building at the University of Münster. Communication between the camera, external web services and the framework are based on the HTTP protocol. The framework was developed as an application and written in Java. To generate and handle the 3D scene models of the cameras environment, the Java3D library was used. As pointed out in the previous section, in order to visualize 2D data in a 3D environment, it is necessary to construct a 3D model based on the given data and some additional assumptions. Our approach to visualize geo–referenced 2D data (here mainly weather radar data, see next section for a detailed description) is based on the construction of a 3D sky model, where the weather radar data is mapped on. For this purpose a sky plane is constructed according to the spatial extent of the areal data and with the height of 3000 m as an average height of clouds. The dimension of the requested data is chosen according to the theoretic visibility of approximately 50 km.

4.4 Sensors and Actuators

As the framework should be able to integrate different sensor data from the internet, it is reasonable to use standardized interfaces to support the exchangeability of sensor data. This implies, that the collection of sensor data e.g. weather data such as wind direction or temperature is separated from the offering of sensor data over the internet. Therefore web services specified by the Open Geospatial Consortium (OGC) are used to support interoperability between the framework and external web services. Furthermore using web services relying on standardized interfaces facilitates the exchangeability of the services and which leads to a smoother integration of additional data into the framework. The communication structure between the sensors and actuators, the web services and the framework are shown in figure 3. As a Web Coverage Service (WCS) provides access to geospatial data as coverage (raster) data with its original semantics [5], in our case it is used to fetch weather radar data offered in an ASCII grid format. The radar data is divided into 6 classes showing the quantity of precipitation per hour. The spatial resolution of the data is 2 x 2 km and the data is actualized every 15 minutes. In order to get access to observation-respectively measurement outcomes like weather phenomenon data (e.g. temperature, wind direction, wind speed) the Sensor Observation Service (SOS) is used which acts like a wrapper hiding different communication protocols and data formats behind its standardized interface [2]. Sensor data is encoded on request as XML according to the Observation & Measurements specification

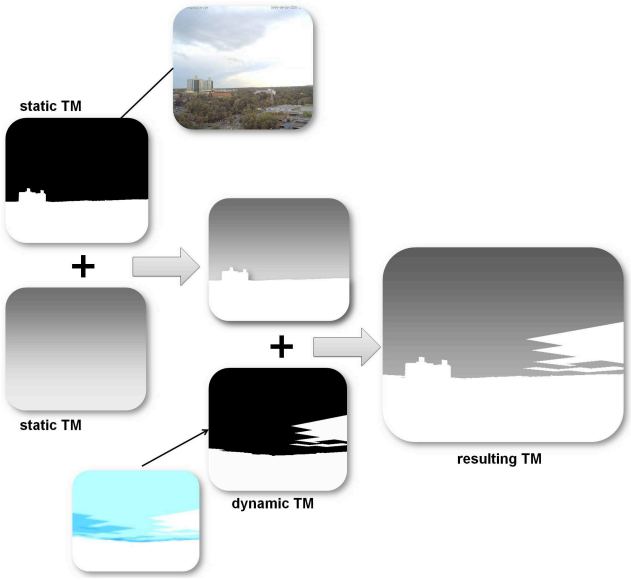


Figure 5: Creation of the transparency masks for geo-data representing precipitation around the area of the camera position.

being a part of the Geography Markup Language (GML). The sensor data we are using is updated every 10 minutes. The pan-tilt device of the camera as actuator is steered by the Sensor Planning Service (SPS) which provides a standardized interface to manage different stages of observation procedures, i.e. planning, scheduling, tasking, collection, archiving and distribution [2]. Although the pan device provides continuous access to camera positions between the 0° and 355°, we have decided to limit the AR weather cam to 16 pre-calibrated positions. The spacing between these positions is 22.5° and every position could be accessed exactly as they are stored in the pan-tilt device.

4.5 Example

The first step of the image geo-mashup process consists of retrieving a value for the wind direction from a SOS which is used to pan the real camera with a SPS. In order to make a contemporary weather forecast, the camera image should show the (short term) future weather by pointing to the wind direction. After the real camera has been rotated successfully, the new orientation parameter (roll, pitch and yaw angle) are stored in the virtual camera object.

In the next step, image layers with their corresponding transparency masks are created and combined with each other in a bottom up order. The bottom layer 1 consists of the current camera image and an opaque transparency mask. Creating layer 2 starts with retrieving weather radar data as raster data from a WCS. This geo-data has a weak geo-reference, since it provides for each data item the position in longitude and latitude. This weak referenced data is integrated under the assumption of an average height (as explained in section 4.2) in the strong referenced (6 degrees of freedom) image data. At this point a 2D image is created which is used as a texture and mapped onto the sky plane in the 3D sky model. The sky model is created according to the first approach mentioned in section 4.2 and a snap shot is taken from the 3D scene in consideration of

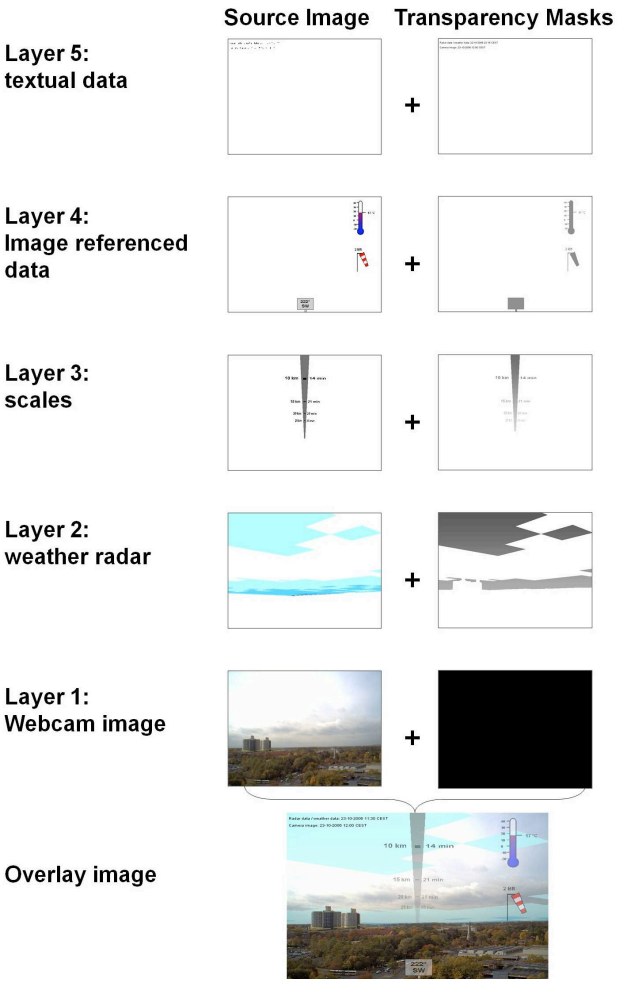


Figure 6: The composition of different image layers into the resulting image geo-mashup.

the real world camera parameter (position and orientation). The corresponding transparency mask of this layer is generated by three separate masks containing a static mask for the given camera orientation defining the alpha values for sky and no-sky regions, a smooth-mask with graded alpha values to account for the reduced visibility in the distance (i.e. close to the line of horizon) and a dynamically generated mask from the scene snap-shot separating background and data regions in the image. The separate transparency masks which are combined and the resulting mask are shown in figure 5. The third layer contains a spatial and temporal scale constructed as a 3D model corresponding to the sky model. The spatial scale shows distances of the rain clouds to the camera position being measured as distances on the sky plane. The temporal scale calculated from the distances and the wind direction provides an approximately forecast when the rain may reach the camera position. Before calculating the values of the temporal scale a value for the wind speed has to be retrieved from a Sensor Observation Service. Making a scene snap shot and creating the corresponding transparency masks is done as described above

The fourth layer displays discrete data with weak geo-reference (longitude/latitude or orientation) e.g. current

temperature or wind speed data associated with the geographic position of the camera to help users to make an easy interpretation of the image geo-mashup. For this purpose the discrete data values are not only presented as textual data but are enhanced by dynamic graphical representations in order to provide more efficient analyses. These symbolic representations should be placed on the underlying image where they do not mask other important information or overlap with other graphical representations. Therefore regions have to be defined either statically or dynamically according to the underlying image where to place these additional elements. Possible positions could be at the image's border area, e.g. the left or right hand side. For our purpose of displaying temperature data as a (dynamic) thermometer we have chosen a static position on the right hand side in the sky part of image. For the element that indicates the heading of the camera we have decided to place it the lower center part of the image. This position for headings can often be founding in advanced binoculars with an integrated compass. Figure 6 shows all different image layers and their combination according to the corresponding transparency masks in the final image geo-mashup result.

5. DISCUSSION AND FUTURE WORK

It is obvious that the interpretation of the resulting image depends on the additional data overlaying the camera image. In our case the resulting image provides a local rain probability forecast (for the camera's position) based on the weather radar data displayed in the sky part of the image and the camera's orientation pointing to the weather direction (where the rain is coming from). To be able to assess the weather situation (and to answer the question: *"Will it rain in 30 minutes?"*) spatial and temporal hints displayed as a scale bars are given to the user. The quality of this local and short term weather forecast depends of course on the geospatial and temporal resolution (refresh period) of used sensor data. The quality and update rates we are currently using for the AR weather cam provide satisfying results. A video of our prototypical service can be accessed at: `http://ifgi.uni-muenster.de/~j_scho09/ARCam.mov`. So far, we have received very encouraging feedback, but of course we have not made any formal evaluation and it is well known that information on the weather is always very popular.

While designing the image geo-mashup service, simplified assumptions have been made. One assumption concerns the estimated height of the rain clouds and another the estimated distance to the theoretic horizon which are both used to construct a 3D sky model to display the radar data. Besides these assumptions we have assumed that the wind direction and wind speed at the ground correspond to the direction and speed of the rain clouds in order to simplify the calculations. We know that this can lead to wrong camera headings, if lower winds differ from winds in higher regions. A possibility that we are currently exploring is to analyze the optical flow over a longer temporal sequences of weather radar data, which should highly correlate with the direction and speed of clouds.

Another interesting question regards the evaluation of the spatial properties of the image geo-mashup, i.e. to control how precise (in terms of spatial resolution) the weather data has been integrated into the image. The correctness of the overlay image can be determined by visually identifying corresponding points in the 3D and 2D view and comparing the displayed distances (3D scale view) to the measured distance (2D view). Under assumption that the reference of the geo-data is correct, we could confirm in such a way that the visualisation of the radar data in 3D space is correct. We are currently working on several additional details that we plan to integrate into the AR weather cam framework. Instead of using a plane for the sky model, we will test a dome-based (i.e. a half sphere) sky model, which might produce visually more pleasant results. We are also looking for more accurate geo- services (i.e. with higher geospatial and temporal resolution) that help us to improve the quality of the visualisation. It would also possible to integrate data from additional geo-services (e.g. wind speed, humidity and barometric pressure), which could be helpful in other usage scenarios of the AR weather cam. Another line of research regards the automatic detection of the horizon line. This could be achieved by analyzing longer temporal sequences of camera images and looking for the border between the rapidly changing sky and the rather stable ground. Additional geographic information, such as 3D representation of prominent landmarks and corresponding labels would further enhance the user's spatial orientation. Until now the camera is completely under control of the system. A straight forward extension would be to allow users to control the movement of the camera and to decide freely on the perspective of the weather cam. The handling would then be close to [6] and [22] combined with the advantage of our proposed short term weather forecast. An important aspect that we have not covered in our work is that of interoperability of geo-services. Currently we assume that the designer of the application decides which services to combine and how to do this. Depending on the degree standardization this can still be a very tedious task. It would be interesting to explore different ways to achieve an automatic goal-driven service composition. To further underline the mashup character of our approach we are currently implementing a web service that will allow owners of arbitrary web cams, to have their images augmented with weather data around the sites of their cameras. By just indicating the position an orientation of the cam, plus providing an image mask with information on sky and no-sky regions, web cam owners will be able to easily create their own image geo-mashup. In addition to that we think of augmenting Google's street view images with this real time weather data to adapt the image to the current weather situation. We are also exploring the possibilities to use the sky-part of an image to visualize other data than weather data. For example, one could think of visualizing geo–referenced traffic information, smog concentration or network coverage by using the *low information* areas of outdoor images: the sky.

6. CONCLUSIONS

In this paper we have presented the general concept of geo-image meshups, which combine geo- referenced data obtained through standardized web service with concepts from Augmented Reality to enhance a perspective image with additional geo information. By explaining in detail our AR weather cam system we have underlined our idea to use multiple layers for each type of geo-information. We have proposed to divide the process of image geo-mashups into three sub-steps; service selection, layouting and registration and have given instances of solutions to these sub-steps in the

context of the AR weather cam system. As the standardization of geo-services continues, we believe that producing complex and intelligent image geo-mashups will be much easier than today. Given the falling prices of camera equipment and the expected distribution of camera hardware in the near future, we are convinced that image geo- mashups will be a powerful means to integrate various spatial and temporal data sources into one single image.

7. ACKNOWLEDGMENTS

This work has been partially supported by a grant of the Department of Geosciences at the University of Münster.

8. REFERENCES

[1] D. Butler. The Web-Wide World. *Nature*, 439(16):776–778, 2006.

[2] I. Simonis. Sensor Webs: A Roadmap. *In Proc. of the 1st Goettinger GI and Remote Sensing Days, Goettingen, Germany*, 2004.

[3] R. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments(1054-7460)*, 6(4):355–385, 1997.

[4] R. Lerner. At the Forge: Creating Mashups. *Linux Journal*, 2006(147), 2006.

[5] J. Evans. Web Coverage Service (WCS), Version 1.0. 0. *Open Geospatial Consortium, Wayland, MA, USA*, 2003.

[6] J. Mower. Implementing an Augmented Scene Delivery System. *In Proc. of ICCS Part III*, pages 174–183, 2002.

[7] P. Milgram and F. Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, 77(12):1321–1329, 1994.

[8] I. Siio, T. Masui, and K. Fukuchi. Real-world Interaction using the FieldMouse. *In Proc. of UIST '99*, pages 113–119, 1999.

[9] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster. A Touring Machine: Prototyping 3D mobile Augmented Reality Systems for Exploring the Urban Environment. *Personal Technologies*, 1(4):208–217, 1997.

[10] S. Goose, H. Wanning, and G. Schneider. Mobile Reality: A PDA-Based Multimodal Framework Synchronizing a Hybrid Tracking Solution with 3D Graphics and Location-Sensitive Speech Interaction. *In Proc. of Ubicomp '02*, 2002.

[11] R. Bane and T. Höllerer. Interactive Tools for Virtual X-Ray Vision in Mobile Augmented Reality. *In Proc. of ISMAR '04*, pages 231–239, 2004.

[12] R. Malaka and A. Zipf. DEEP MAP-Challenging IT research in the Framework of a Tourist Information System. *Information and Communication Technologies in Tourism*, 7:15–27, 2000.

[13] E. Bier, M. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and Magic Lenses: The See-through Interface. *In Proc. of GRAPHITE '93*, pages 73–80, 1993.

[14] B. Hecht, M. Rohs, J. Schöning, and A. Krüger. WikEye–Using Magic Lenses to Explore Georeferenced Wikipedia Content. *In Proc. of PERMID '07*, 2007.

[15] J. Schöning, A. Krüger, and H. Müller. Interaction of Mobile Camera Devices with Physical Maps. *In Adjunct Proc. of Pervasive '06*, 2006.

[16] M. Rohs, J. Schöning, A. Krüger, and B. Hecht. Towards Real-Time Markerless Tracking of Magic Lenses on Paper Maps. *In Adjunct Proc. of PERVASIVE '07*, 2007.

[17] J. Schöning, B. Hecht, M. Raubal, A. Krüger, M. Marsh, and M. Rohs. Improving Interaction with Virtual Globes through Spatial Thinking: Helping users Ask "Why?". In *IUI '08: Proc. of the 13th annual ACM conference on Intelligent User Interfaces*. ACM, 2008.

[18] G. King, W. Piekarski, and B. Thomas. ARVino – Outdoor Augmented Reality Visualisation of Viticulture GIS Data. *In Proc. of ISMAR '05*, pages 52–55, 2005.

[19] H. Schnädelbach, B. Koleva, M. Flintham, M. Fraser, S. Izadi, P. Chandler, M. Foster, S. Benford, C. Greenhalgh, and T. Rodden. The Augurscope: A Mixed Reality Interface for Outdoors. *CHI '02*, pages 9–16, 2002.

[20] N. R. Hedley, M. Billinghurst, L. Postner, R. May, and H. Kato. Explorations in the Use of Augmented Reality for Geographic Visualization. *Presence: Teleoper. Virtual Environ.*, 11(2):119–133, 2002.

[21] G. Schall, E. Mendez, S. Junghanns, and D. Schmalstieg. Urban 3D Models: What's underneath? Handheld Augmented Reality for Subsurface Infrastructure Visualization. *In Proc. of Ubicomp '07*, 2007.

[22] V. P. C. Brenner and N. Ripperda. The Geoscope – A Mixed – Reality System for Planning and Public Participation. *In Proc. of UDMS '06*, 2006.

[23] M. Naaman, S. Harada, Q. Wang, H. Garcia-Molina, and A. Paepcke. Context data in geo-referenced digital photo collections. *In Proc. of Multimedia '04*, pages 196–203, 2004.

[24] R. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *Radiometry*, 1992.

Posters sessions

How Coherent Environments Support Remote Gestures

Naomi Yamashita
NTT Communication Science Labs.
naomi@cslab.kecl.ntt.co.jp

Keiji Hirata
NTT Communication Science Labs.
hirata@brl.ntt.co.jp

Toshihiro Takada
NTT Communication Science Labs.
takada@brl.ntt.co.jp

Yasunori Harada
NTT Communication Science Labs.
hara@brl.ntt.co.jp

ABSTRACT
Previous studies have demonstrated the importance of providing users with a coherent environment across distant sites. To date, it remains unclear how such an environment affects people's gestures and their comprehension. In this study, we investigate how a coherent environment across distant sites affects people's hand gestures when collaborating on physical tasks. We present video-mediated technology that provides distant users with a coherent environment in which they can freely gesture toward remote objects by the unmediated representations of hands. Using this system, we examine the values of a coherent environment by comparing remote collaboration on physical tasks in a fractured setting versus a coherent setting. The results indicate that a coherent environment facilitates gesturing toward remote objects and their use improves task performance. The results further suggest that a coherent environment improves the sense of co-presence across distant sites and enables quick recovery from misunderstandings.

Categories and Subject Descriptors
H.4.3 Information systems applications: Communications applications – Computer conferencing, teleconferencing, and videoconferencing

Keywords
Computer-supported collaborative work, collaborative physical task, video-mediated communication, coherent environment, remote gesture

1. INTRODUCTION
Recent research on distance work has significantly demonstrated the importance of providing people with a coherent environment [3, 5, 2]. When the positional relationships between distant sites become fractured, as is often the case with conventional video systems, people tend to have difficulties in making sense of others' speech and gestures with the surrounding environment [2].

The problem becomes particularly serious in distance work where gestures play a significant role. Collaborative physical tasks [1] fall into such works, in which one or more individuals (workers) work with a concrete object under the guidance of a remote individual (helper).

AVI'08, May 28–30, 2008, Napoli, Italy.

Given that gesturing is so crucial to collaborative physical tasks, a variety of video systems are being developed to facilitate remote gesturing (e.g., DOVE [1], Agora [7]). They typically facilitate remote gesturing by introducing a coherent space (i.e., shared visual space) in which the relationships between helper's gestures and the remote objects are maintained.

While previous studies have indicated that the introduction of a coherent space improves task performance [5], researchers have so far focused exclusively on how the introduction of a coherent space affects the worker's understanding of the helper's gestures [5, 1].

Yet no one has investigated the influence of the introduction of coherent space on helper's gestures. In other words, we still lack an understanding of how coherence affects gesture usage in collaborative physical tasks. For example, does a coherent environment equally facilitate all types of gestures or only certain types? If the latter case is true, what types of gestures are facilitated, and are those gestures understood efficiently in relation to the surrounding environment? Furthermore, does a coherent environment enhance the collaborators' sense of co-presence? Answering such questions will provide guidance for designers of video-mediated technologies.

2. CURRENT STUDY

2.1 Re-classification of Gestures
Previous studies suggest that people use several types of gestures during collaborative physical tasks [1]. The classification of such gestures differs between systems [8], but all differentiate between pointing and representational gestures.

Pointing gestures are used to refer to objects and locations. Representational gestures are used to represent the shapes of objects and the nature of the actions to be done with the objects [8]. Representational gestures are further classified into three types that play a critical role in collaborative physical tasks [1]: iconic, spatial, and kinetic. Iconic representations form hand shapes to show what a particular object looks like; spatial gestures describe the distance between two objects by typically placing two fingers or hands a certain distance apart; kinetic gestures describe how actions should be performed on an object.

While researchers have mainly focused on the role of each gesture, we are more interested in the mediation of gesture across distant sites. To this end, we re-classify representational gestures into two types based on whether the gesture involves interaction with objects at a remote site: *remote-oriented representational gestures*, which involve interaction with remote sites, and *locally-closed representational gestures*, which do not involve interaction with remote sites.

We expect that gestures involving interaction with remote objects (i.e., both remote-oriented representational and pointing gestures) are only understood effectively in a coherent environment where the relationship between gesture and object is maintained. Those gestures will not be understood correctly in a fractured setting, in which positional relationships between distant sites are not preserved.

2.2 Hypotheses

If distant collaborators are provided with a coherent environment, we expect that collaborators will be able to understand each others' gestures better in relationship with the surrounding objects. This leads to several hypotheses regarding the performance of helper-worker pairs in the mentoring physical tasks explored in this study.

H1 (Gesture usage): Collaborators in a coherent environment will make greater use of remote-oriented representational and pointing gestures than a fractured environment. Conversely, collaborators in a fractured environment will make greater use of locally-closed representational gestures than a coherent environment.

H2 (Effects of Gestures): Higher use of pointing and remotely-oriented representational gestures is correlated with faster performance in a coherent environment, but not in a fractured setting.

When collaborators frequently gesture toward a remote site, we expect that collaborators will feel co-present with their distant collaborators and objects and tend to often use local deixis.

H3 (Sense of co-presence): Collaborators in a coherent environment will frequently use local deixis and achieve a greater sense of co-presence than in a fractured environment.

2.3 t-Room System

In this study, we investigate the value of the coherent environment by comparing mentoring collaborative physical work using the t-Room system [4].

Figure 1 shows the hardware design of the system. A single t-Room consists of six modules called Monoliths arranged octagonally and a worktable at the center embedded with LCD displays.

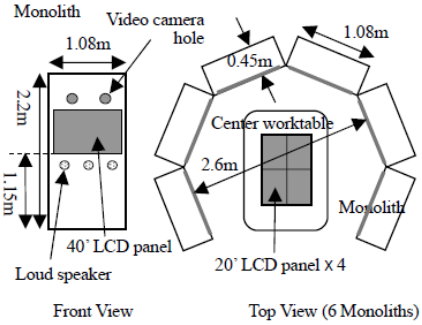


Figure 1 Hardware Design of t-Room

Users in the t-Room are surrounded by six 40-inch LCD panels (resolution of 1280 by 768), six HDV cameras, and 18 loudspeakers. An HDV camera is mounted inside each Monolith to capture the views inside the room, especially the heads and upper bodies of users. A polarized film is placed over each camera to eliminate infinite video feedback. LCD panels are positioned at the height of user heads and upper bodies, showing both local users' self-reflection images and remote users' images (Figure 3). The self-reflection images are intended so that the users can check how their own figures are projected at the distant site. An HDV camera is also hung from the ceiling to capture the scene at the worktable. In this way, collaborators can share the same views projected on the wall and table screens; collaborators are aware of exactly what the others can see of the work space.

2.4 Experimental Design

We installed two identical t-Rooms in the cities of Atsugi and Kyoto, which are approximately 400 km apart. A commercially available 100 Mbps optical fiber line connects the two rooms. The network delay for video and audio data transmission between Atsugi and Kyoto is around 0.7−0.8 and 0.4-0.5 seconds, respectively.

In the experiment, a helper and a worker performed a repair task on a personal computer (DELL OptiPlex 170L) in each of two media conditions: (a) *fractured setting*: a video system that fractures the relationships between gesture and the target object in a distant space; a handy camera that captures a partial view of the worker's task space, and a scene camera capturing the helper's upper body (see Figure 2). (b) *coherent setting*: a video system that provides collaborators with a coherent environment. Cameras and displays are setup so that the relationship between action and environment is maintained across distant sites.

The study included ten participants. The workers consisted of nine part-time employees who had never deconstructed a PC or used a video-mediated communication system before the experiment. We recruited a male helper who is a PC repair expert and had worked as an instructor at a PC technical college to provide guidance from the Atsugi t-Room to all nine workers in the Kyoto t-Room. Prior to the experiment, the helper practiced giving instructions with two extra participants, so that he could offer steady instructions throughout the experiment.

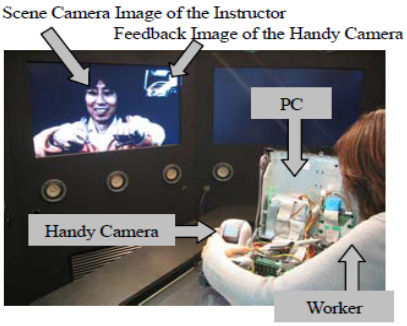


Figure 2 Media Condition (a): Fractured Setting

Figure 3 Media Condition (b): Coherent Setting

Table 1 Characteristics of Each Media Condition

	Condition (a)	Condition (b)
View	Narrow	Wide
Detailed Image	Yes	No
Camera Movement	Yes	No
Coherence	Fractured	Coherent

Table 1 summarizes the differences between the two media conditions focused on in our study. In condition (a), we setup a general video setting suitable for mentoring collaborative physical tasks; a worker can move the camera and control what the helper sees; he can zoom or/and focus on parts of the object to which he wants to draw the helper's attention. However, the camera view in the condition is relatively narrow and fractures the environment across sites. In condition (b), collaborators are provided with a wide view of each other's spaces, although they cannot control the camera views. The collaborators are also provided with a coherent environment, although the helper's gestures (particularly pointing gestures toward a remote object) are sometimes occluded by the actual object.

2.5 Procedure

The following was the experiment's procedure:

Procedure (1): Workers were given explanations how the system worked. The helper and a worker also engaged in a short-term pre-study task to become familiar with the t-Room environment and to grasp how to deal with a real object.

Procedure (2): Workers were given an overview of their roles in the experiment: to replace a broken PC. Then, the helper and a worker engaged in three tasks: exchanging a power supply unit, a hard disk drive, and a DVD unit, each in different system settings: fractured setting, coherent setting, and another setting, which is over the scope of this paper. Trials, tasks, and media conditions were counterbalanced. The pairs were instructed to complete the task as quickly as possible. They were allowed to freely communicate, but the helper was instructed to avoid giving workers information unrelated to their current task.

Procedure (3): Following the tasks, workers and the helper were interviewed about the ease of understanding each other's utterances, the usefulness of specific technological features, and their preference of technology.

3. RESULTS

Since the experiment was initially designed to compare three media conditions [9] (i.e. fractured vs. coherent vs. coherent with partially fractured space), results were analyzed in a trial by task by media condition repeated measures ANOVA.

Pairs completed the tasks in an average of 12.4 and 11.9 minutes under fractured and coherent settings, respectively. The differences in task completion times were not significant. Furthermore, all workers correctly exchanged the PC units in both conditions. However, two of nine workers misunderstood the helper's instruction and attached the PC cord to a different place during the fractured setting.

3.1 Effects of Gestures

3.1.1 Gesture Usage

The helper frequently gestured when instructing the workers; he gestured once every 11.8 seconds in the fractured setting and once every 9.4 seconds in the coherent setting. Analysis on the frequency of gesture indicated a significant main effect for media condition ($F[2,18]=6.29$, $p=<.01$). Post-hoc tests indicated that the helper gestured more frequently in the coherent than the fractured setting ($p<.05$).

To investigate how the helper's gestures differed between conditions, we classified them into three categories: Pointing, Remotely-oriented representational, and Locally-closed representational. Two independent coders classified gesture samples until they reached 90% agreement. They then each coded half of the videos. Table 2 shows the proportion of gestures in each of the three categories across each media condition.

Table 2 Proportion of Helper's Gestures in Each Category

Environment	*Pointing*	*Remotely -oriented*	*Locally -closed*
(a) Fractured	13%	19%	68%
(b) Regular t-Room	44%	31%	25%

Analysis on the proportion of each gesture usage indicated that the usage of gestures differed significantly across media conditions (pointing gestures: $F[2,18]=111.41$, $p<.001$; remotely-oriented gestures: $F[2,18]=23.18$, $p<.001$; locally-closed representational gestures: $F[2,18]=255.37$, $p<.001$). As predicted by *H1*, post-hoc tests indicated that the helper made greater use of pointing and remotely-oriented gestures in the coherent than in the fractured setting ($p<.001$).

3.1.2 Effects of Gestures on Completion Time

Although the helper frequently gestured toward the PC unit in the worker's site, not all his gestures could be seen by the worker; some were out of camera site. Approximately half of the helper's gestures were unable to see from the worker's site in both fractured and coherent settings (gestures were not deemed "viewable" when part of the view was missing). Regardless of many cameras used in the coherent setting, many of the helper's gestures were off the camera site, since the helper frequently gestured toward the object on the central table, which was slightly lower than the shooting area (i.e., side wall screens).

To examine *H2*, we first calculated the rate of viewable gestures per second and then examined the relation between the viewable gestures and task performance (Table 3).

As shown in Table 3, the rate of viewable remotely-oriented gestures were significantly correlated with faster performance times in the coherent setting, but not in the fractured setting. The rate of viewable pointing gestures slightly correlated with the task performance in the coherent setting. A higher rate of viewable locally-closed gestures was slightly correlated with faster performance in the fractured setting, but not in the coherent setting.

Table 3 Correlation between Viewable Gestures and Completion Time

Environment	*Pointing*	*Remote-oriented*	*Locally-closed*
(a) Fractured	r= -.12 (p= .77)	r= .41 (p= .28)	r= -.59 $^{+}$ (p=.09)
(b) Coherent	r= -.58 $^{+}$ (p= .09)	r= -.80 ** (p< .01)	r= -.29 (p=.45)

+ significant at 10% level; * significant at 5% level; ** significant at 1% level

3.2 Sense of Co-presence

Previous studies have shown that people feel co-present when they gesture a lot. We have seen in Section 3.1 that the helper gestured significantly more in the coherent setting than in the fractured setting.

To examine *H3,* we further calculated the number of local deixis in each utterance and compared the values across media conditions (Figure 4). Typically, people use local deixis (e.g., *here, this, these*) more often when they feel present in a remote environment and co-located with a set of distant objects [4].

Analysis on the numbers of local deixis per utterance indicated significant main effects for media condition ($F[2,18]=18.37$, $p<.001$), but no main task effect. Post-hoc tests indicated that the use of local deixis was significantly higher in the coherent setting than the fractured setting ($p<.001$).

Consistent with the quantitative results, several participants remarked in the post-experimental interviews that they felt more co-present with their remote collaborator in the coherent setting than the fractured setting.

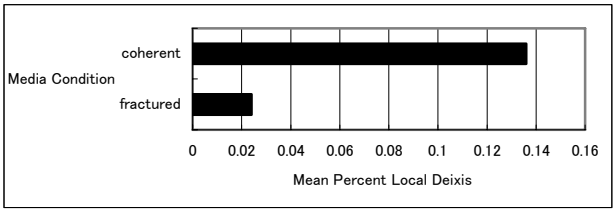


Figure 4 Proportion of use of Local Deixis per Utterance

3.2.1 Overcoming Misunderstandings

In the coherent setting, we found interesting cases where the helper instructed the worker as if they were in the same room, relying on the practices and resources of co-located collaboration; when the workers had trouble identifying a PC component, the helper sometimes walked around the table and directed the workers to look at the PC from his standing position as shown in the following excerpt.

Helper: Can you pull the loop like this? [Gestures how to pull the loop].
Worker: Yes. [Tries to pull out a different component].
Helper: Umm. Excuse me.
Worker: Yes?
Helper: Can you come over here? [Walks around the table] …stand over here?
Worker: Ok? [Walks around the table, and stands very close to the helper].
Helper: This orange cable. . . See it? Bend it down a little bit.
Worker: [Bends it down as told].
Helper: See the orange thing… looks like a wire? Something round.
Worker: Oh, I got it. This?
Helper: Yes. Pull it up.

Such a scene only makes sense when the participants in the rooms can move freely inside the rooms, while maintaining the spatial relationships between the two sites.

4. Conclusions

Our results demonstrate the value of providing distant collaborators with a coherent environment for collaborative physical tasks. First, a coherent environment improved the collaborators' sense of co-presence and enabled them to rely on the practices and the resources of co-located collaboration. For example, collaborators walked around the table to view an object from the same angle and quickly resolved misunderstandings.

Second, the environment facilitated collaborators' use of remotely-oriented gestures (i.e., representational gestures involving interaction with remote sites). Using such gestures in the environment facilitated grounding in the task procedure and was highly correlated with faster performance.

Regardless of such benefits of the coherent environment, the workers did not complete the tasks significantly faster in the coherent setting than in the fractured setting. Perhaps the visibility of the helper's remotely-oriented gestures was low (13% of the total gestures) so no effects on overall task performance times were visible.

5. ACKNOWLEDGEMENTS

We thank Yoshinari Shirai, Shigemi Aoyagi, and Junji Yamato for their assistance. We also thank Hideaki Kuzuoka and several anonymous reviewers for their valuable comments.

6. REFERENCES

1. Fussell, S. R., Setlock, L., Yang, J., Ou, J., Mauer, E., and Kramer, A. D. I. Gestures over video streams to support remote collaboration on physical tasks. *Journal of Human-Computer Interaction*, 19, (2004), 273-309.
2. Heath, C. and Luff, P. Disembodied Conduct: Communication Through Video in a Multi-media Office Environment. *Proceedings of CHI'91,* ACM Press, (1991), 99-103.
3. Heath, C., Luff, P., Kuzuoka, H., and Yamazaki, K. Creating Coherent Environments for Collaboration. *Proceedings of ECSCW'01*, Kluwer Academic Publishers (2001), 119-128.
4. Hirata, K., Harada, Y., Takada, T., Aoyagi, S., Shirai, Y., Yamashita, N., and Yamato, J. The t-Room: Toward the Future Phone, *NTT Technical Review*, 4, 12, (2006), 26-33.
5. Kirk, D., Crabtree, A. & Rodden, T. Ways of the Hands. *Proceedings of ECSCW'05*, Kluwer (2005).
6. Kramer, A., Oh, L., and Fussell, S. Using Linguistic Features to Measure Presence in Computer-Mediated Communication. *Proceedings of CHI'06*, ACM Press (2006), 913-916.
7. Kuzuoka, H., Yamashita, J., Yamazaki, K., Yamazaki, A. Agora: A Remote Collaboration System that Enables Mutual Monitoring. In CHI'99 Extended Abstracts, (1999), 190-191.
8. McNeill, D. *Hand and mind: What gestures reveal about thought.* Chicago: University of Chicago Press (1992).
9. Yamashita, N., Hirata, K., Takada, T., Harada, Y., Shirai, Y., and Aoyagi, S. Effects of Room-sized Sharing on Remote Collaboration on Physical Tasks. *IPSJ Journal*, Digital Courier, Vol. 3, (2007).

SLMeeting: Supporting Collaborative Work in Second Life

Andrea De Lucia, Rita Francese, Ignazio Passero, Genoveffa Tortora
Dipartimento di Matematica e Informatica
University of Salerno
84084 – Fisciano (SA) – Italy
Tel:+39 (0) 89 963376

adelucia@unisa.it, francese@unisa.it, ipassero@unisa.it, tortora@unisa.it

ABSTRACT

Second Life is a virtual world which is often used for the synchronous meeting of teams. However, supporting distributed meeting goes beyond supporting user activities during the meeting itself, because it is also necessary to facilitate their coordination, arrangement and set up.

In this paper we investigate how teams can work together more effectively in Second Life. We also propose a system, named SLMeeting, which enhances the communication facilities of Second Life to support the management of collaborative activities, organized as conferences or Job meetings and later replayed, queried, analyzed and visualized. The meeting organization and management functionalities are performed by ad-hoc developed Second Life objects and by the communication between these objects and a supporting web site. As a result, the functionalities offered by Second Life are enriched with the capabilities of organizing meetings and recoding all the information concerning the event.

Keywords

Collaborative work, groupware, CSCW, multimedia meetings, Second Life, Collaborative Virtual Environment, 3D interfaces

1. INTRODUCTION

Many companies manage projects that involve people from different teams and other companies around the world. Meetings are the only mechanism enabling the effective resolution of issues and the building of consensus [1]. The drawback of meetings is their cost in resources and the difficulty in their management.

In the last two decades several research efforts have been devoted to support multimedia meetings, e.g. [3], [6], [8], [15]. In particular, 3D interfaces are a very popular base for groupware. Indeed, the metaphor of meeting rooms is adopted by several tools, such as [2], [11], [14], [16], [18]. These tools represent human participants with avatars. Generally they do not offer particular features for the meeting control and set-up.

The goals is to exploit the opportunity offered by the technological revolution [17] to support distance meeting, obtaining a simple, low-cost setup that is affordable for everyday use, avoiding solutions that use expensive tracking systems, or expensive proprietary videoconferencing software.

Second Life (SL) is effective for communications [4]. Indeed, worldwide organizations such as IBM, NASA, or Intel Conference Center [7], exploit the communication features of SL to support synchronous communication. SL is a virtual world where users are able to actually "see" the person they are talking to. This has a great effect on the conversation, even if person's avatars looks nothing like their owners. People sit around a virtual campfire or in a virtual coffee shop to talk, providing a sense of community and realism that is not available in a chat room. Second Life (SL) allows creating virtual meeting rooms where avatars can meet each other and discuss [4]. Communication support is provided by the action awareness, information sharing (projecting a slide or streaming a video), and the vocal and textual chats. Most aspects of synchronous communication considered important for Computer Support Cooperative Learning (CSCW) are covered. However, also in this environment there is no implementation of an explicit meeting support.

In this paper we investigate how Second Life can enable teams to work together more effectively across distance. At this aim we propose a system, named SLMeeting, which enhances the communication facilities of Second Life by supporting the management of collaborative activities which can be organized as conference or Job meetings and later replayed, queried, analyzed and visualized. The meeting organization and management functionalities are performed by ad-hoc objects created using the development framework offered by Second Life. The counterpart Web 1.0 of the meeting management is a web site, which communicates with the SL objects and automatically records the meeting minute and all the information concerning the event. In this way knowledge management processes, such as the development of shared understanding and organizational memory as well as knowledge, are also supported.

2. THE PROPOSED APPROACH

In this section we describe how we have enhanced Second Life to be not only a means for social interaction support, but also (beyond that) a tool for CSCW. As previously affirmed, Second Life enhances communication. The ability to teleport a

AVI'08, May 28-30, 2008 - Napoli, Italy.

representation of yourself to a meeting anywhere in a virtual world could, on the face of it, solve many meeting problems [4]. Head movements, body postures, facial expressions, emotions and conversational gestures also enrich the avatar communicability.

The participants of a meeting should be able to express their ideas and make decisions. They should dispose of the needed material, such as documents, meeting minutes, presentations. Additional features supporting coordination and awareness services are required to coordinate and to solve possible conflicts between collaborative entities involved in the session [5]. To this aim we have created a meeting room in Second Life where annotations of a meeting are automatically generated and the meeting control is managed using ad-hoc interactive, metaphorical objects. To support workplace awareness, conversations and decisions are automatically saved for later references. Ad-hoc developed SLMeeting Objects in Second Life send data to the SLMeeting server outside SL using HTTP requests to PHP pages. The server then accesses the database and provides the required information back to the objects which handle it by LSL scripts.

2.1 Meeting Set Up

The proposed approach enables to organize the meeting with the support of the SLMeeting web site. A wizard helps the organizer to create a meeting. In particular, it is necessary to create the event, identify the participants and their roles.

All the selected participants are invited to adhere to the meeting through the Web site. Participants have to communicate their adhesion and their SL identities to enable access control to the meeting area and to schedule and control their interventions. To reach the meeting, participants can use a link on the Web site to directly teleport themselves to the Second Life meeting room, or access directly from SL. Figure 1 shows the SLMeeting setting. Let us note that we selected an outdoor setting, because after the construction of a typical meeting room as a close environment we noticed that the potentiality of flying, and the avatar and camera movements were strongly bounded by such a working environment.

2.2 The different roles and their tasks

We analyzed how a meeting is managed in order to recognize the various participant roles. Once we recognized the various roles we ensured that each user interacts only with the UI established for his/her role. Thus, in order to facilitate the meeting, we assign the following roles to selected participants, or to SLMeeting objects.

The *facilitator* organizes the meeting and guides its execution. During the meeting preparation, he/she submit the meeting agenda and the support material useful to the participants. The meeting agenda consists of a list of discussion points and for each of them a reference speaker is assigned by the facilitator. The facilitator assigns a fixed time to each discussion point and/or user intervention. When the speaker ends the talk, the facilitator starts the discussion. SLMeeting automatically schedules the interventions considering the booking list, unless the facilitator modifies the order.

The *scribe object* automatically records all the messages of the speakers on the Web Server. These messages are shown to all the participants on a blackboard.

During each talk, the *participants* can book their intervention by the handrising mechanism. The intervention list is available on another blackboard.

The *speaker* is the participant who is actually talking. SLMeeting highlights the speaker avatar by a searchlight. Only the floodlit avatar records its writing in the Meeting Chat bar.

The *timekeeper* role is automatically performed by an SLMeeting object.

2.3 SLMeeting Interaction

As shown in Figure 1, participants are seated around a table, a well accepted metaphor inside a meeting setting. The application issues were generally concerned with the affordances of objects and the lack of help inside the meeting setting. A user manual is available on the web site of the system.

The interaction between the avatars and SLMeeting is performed by using the following objects whose meaning is easily understandable:

- The *facilitator command* bar, shown in Figure 1 (a). The meeting is coordinated by the facilitator pressing the buttons exposed by this interface. In particular, the *start* button begins the meeting, The *interrupt* button enables the facilitator to immediately speak: the searchlight illuminates him/her immediately and the chat bar registers only his/her interventions. The *vote* button starts a voting session. The *hurry up* button sets the color of the searchlight to red. To properly close the meeting, the *end* meeting button should be pressed. The remaining buttons are the same as the ones appearing in the participant gesture bar.
- *Agenda*, the blackboard labeled (b) in Figure 1. A panel shows the title of the meeting and lists items and the corresponding speakers. The moderator clicks on this object and selects the next item action. Then the item is highlighted and the assigned speaker can start his/her talk.
- The *Meeting Chat board*, the blackboard labeled (c) in Figure 1. The text typed in the chat by the floodlit participants is saved in the Web site. Let us note that vocal chat is supported by Second Life, however, using this type of communication does not enable to record the intervention on the Web site, unless introducing speech recognition features.
- The *booking list*, shown in an homonym panel depicted in Figure 1 (d) and better detailed in Figure 2. The current item in agenda is shown, together with the intervention list. The facilitator has the permission of going to the next discussion point using the more internal arrows. Whether the facilitator needs to change the item ordering, he/she selects the item and modify the schedule acting on the more external arrows. Once a participant has spoken, he/she is erased from the list.

- The *participant gesture bar* offers the features of *applause*, *yes*, *no*, and *hand rising*. When the hand rise button is pressed, the participant is added to the booking list. Pressing the cancel button it is possible to erase the booking.
- The *Voting participant palette*. When a voting is required, an apposite palette appears with three buttons: favorable, contrary and abstained.
- The *searchlight.* Once the moderator has selected a speaker from the booking list he/she is floodlit. Only the floodlit avatar records its writing in the Meeting Chat bar.
- *Timekeeping clock*. As shown in Figures 3 and 4, the center of the table hosts an analogical chronometer aiming at signaling the remaining time for the current talk. While the time flows, the green portion of the clock becomes gradually red. When the dish is entirely red, the time is finished.
- The *slide shower*. The slides to be shown during the meeting have to be pre-loaded on this object in image format. The speaker can change the slide by clicking on this component. Only the speaker and the facilitator have this permission.

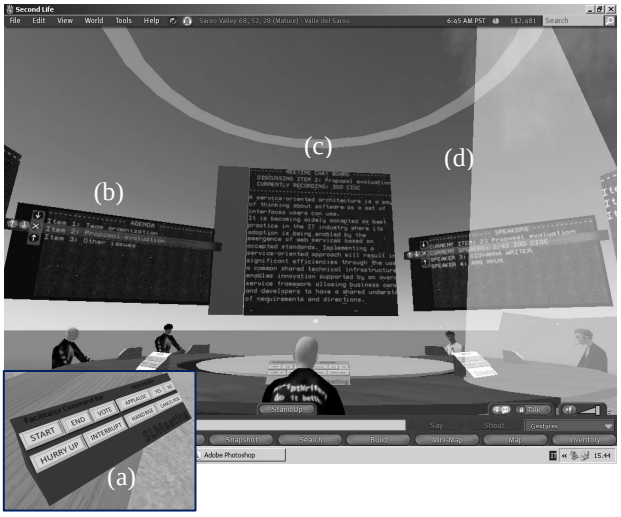


Figure 1. The facilitator feedback boards

Concerning the timekeeping, the systems provides a warning when the time is finished, but leaves the control of the evolution of the discussion to the facilitator. In particular, the facilitator may send an “Harry up” signal to the speaker, or definitely interrupt him/her. In the former case, the searchlight becomes red, in the latter the facilitator acts on the interrupt button. Successively, he/she introduces the next speaker and selects him/her from the booking list. The searchlight is directed on it and the timekeeping clock is reset.

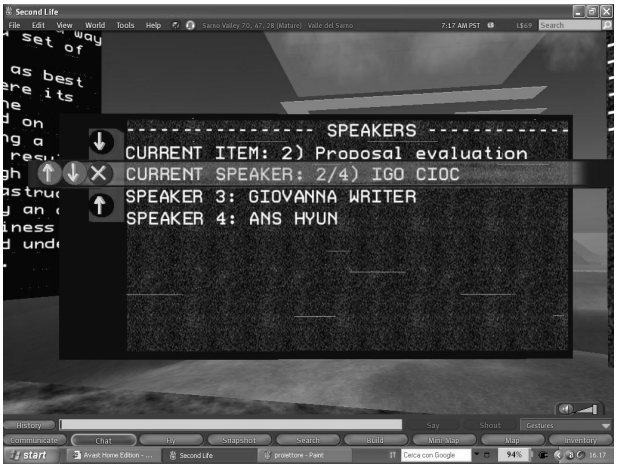


Figure 2. The booking list blackboard

After the end of the meeting, each participant can access the SLMeeting web site and consult the chat of the meeting. In addition, the minute taker creates a minute of the discussion, which can be obtained examining and modifying the recorded meeting chat. This report can be shared with the meeting participants and with other members who where not present at the meeting.

Let us note that the adoption of gesture bars solves the problem to conversation disruptions due to the searching for the appropriate gesture or animation. It is also important to point out that avatars frequently interact with the SLMeeting objects. Because SLScript, the programming language offered by SL, prescribes one second delay after each http request/response there is the need of adopting buffering strategies to forward messages toward the SLMeetng server. To this aim we use the object table to collect the inputs from the user bars and periodically send them towards the Web server.

3. CONCLUSION

In this paper we have presented SLMeetings, a system aiming at enhancing the support offered by Second Life to the management and the control of meetings. To this aim several SL objects have been created, adopting real life metaphors. Each role of the meeting has particular permissions on the SLMeeting objects. Conversations and decisions are automatically saved for later reference to overcome the poor asynchronous communication offered by SL.

We performed a preliminary usability analysis and the results are encouraging [10]. In our experience we tried to simulate a real working environment, although we only used master students of the software engineering course of the Computer Science program at University of Salerno. The participants were fifteen. The students have been grouped in three teams; each team was required to develop an application. The meetings referred to the application analysis and design.

There was no abandonment and (as shown from the questionnaire results) the tool usage was clear. To provide data on user

satisfaction, we asked the participants to fill a questionnaire on their impression on using SLMeeting. As a result, the system provides an useful support and its usage is pleasant. Also the results concerning awareness, social awareness and communication are satisfying. The avatar movement did not create particular problems and also novices are able to easily use the system.

Last-year master students have a very good analysis, development and programming experience, and they are not far from junior industry analysts. In addition, subjects are students enrolled in an advanced course of software engineering, thus they have both knowledge of software development and project management. Unfortunately, in a real working environment there are other pressures and practitioners might have a much lower tolerance for interaction difficulties and can be less expert in the usage of 3D environments. Thus, probably the results cannot be completely generalized to the industrial context and then the experience should be replicated with practitioners. Another aspect to consider is the age of the participants, which ranges from 21-26 years. This kind of users is very practical in playing 3D games and enjoys in finding such a kind of environment on the working place. Different results could be reached with aged users. Thus, we are further investigating how people perceive and interpret meeting situations and how they react on them in a virtual meeting room. To this aim we need to compare the effectiveness of SLMeeting with other meeting modalities, such as audio meeting, face to face meeting or with the support provided by commercial meeting tools.

In the future we also plan to increase the asynchronous communication features offered by SLMeeting. In particular, we are interested in structuring asynchronous communication, such as the versioning of objects that are being co-developed by groups or threaded discussions ordered by topic or time, which are features actually not supported at all.

REFERENCES

[1] *Bruegge B., Dutoit A. H.,* Object-Oriented Software Engineering: Using UML, Patterns and Java, 2nd Edition *Publisher: Prentice Hall*, Upper Saddle River, NJ, 2003.

[2] Carlsson, C., Hagsand, O., "DIVE - a Multi-User Virtual Reality System", In *Proceedings of IEEE Virtual Reality Annual International Symposium* (VRAIS'93), Seattle, WA, September1 993, pp. 394-400.

[3] Cook, P., et al. Project Nick: Meetings augmentation and analysis. *ACM Transactions on Information Systems* (TOIS), 5(2): 132 – 146, April 1987.

[4] Edwards, C., Another World. *IEEE Engineering & Technology*, December 2006.

[5] Henrique João L. Domingos, J. Legatheaux Martins, Nuno M. Preguiça, Coordination Support for Scalable Collaborative Work, in the *Proceedings of Ninth International Workshop on Database and Expert Systems Applications*, 1998. Page(s):554 – 558

[6] Gottesdiener, E., Facilitated Workshops in Software Development Projects, in the *Proc. of the Int. Conference on Application of Software Measurement (ASM 2001)*, San Diego, CA, USA, February 2001.

[7] Intel Conference Center ttp://slurl.com/secondlife/Intel%20Software%20Network/153/122/90

[8] Isaacs, E., Morris, T., Rodriguez, T. K., A Forum for Supporting Interactive Presentations to Distributed Audiences. In the *Proc. of the Conference on Computer-Supported Cooperative Work (CSCW 1994)*, 1994.

[9] Jacovi, M., Soroka, V., Gail Gilboa-Freedman, Sigalit Ur, Elad Shahar, Natalia Marmasse, The Chasms of CSCW:A Citation Graph Analysis of the CSCW Conference. In the ACM Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, 2006.

[10] Lockner, M. and Winroth, U., Usability and Social Interaction in a Collaborative Virtual Environment. http://www.eurescom.de/~public-webspace/P800-series/P807/results/Usability/R7/D2-T3-Usability-R7-SocialInteraction.pdf

[11] Light, J., and Miller, J., D. Miramar: a 3D workplace. In *the IEEE Proc. IPCC*, 2002., 271-282.

[12] Lineage II, http://www.lineage2.com, retrieved October 30th, 2006.

[13] Ling, C., Gen-Cai, C., Chen-Guang, Y., Chuen, C., Using Collaborative Knowledge Base to Realize Adaptive Message Filtering in Collaborative Virtual Environment , in the *Proceedings of ICCT*, 2003.

[14] Nijholt, A., Zwiers, J., Peciva, J., The Distributed Virtual Meeting Room Exercise, In the *Proceedings of the Workshop on Multimodal multiparty meeting processing (ICMI 2005)*, Trento, Italy.

[15] Nunamaker, J.F. et al. Electronic meeting systems to support group work, *Communications of the ACM*, 34(7):40-61, July 1991.

[16] Rosenman, M., Merrick, K., Maher, M., L., and Marchant D., Designworld: A Multidisciplinary Collaborative Design Environment Using Agents In *a Virtual World. Design Computing And Cognition*, Springer Netherlands (2006).

[17] Tapiador, A., Fumero, A., Salvachu'a, J., Aguirre, S. A Web Collaboration Architecture, in the *Proceedings of the 2nd IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing* (CollaborateCom 2006), Atlanta, Georgia, USA

[18] Yankelovich, N. et al. Meeting Central: Making Distributed Meetings More Effective. In the *Proc. of ACM Conference on Computer Supported Cooperative Work (CSCW 2004)*, November 6-10, 2004, Chicago, Illinois, USA.

Balancing Physical and Digital Properties in Mixed Objects

Céline Coutrix and Laurence Nigay

Grenoble Informatics Laboratory (LIG), University of Grenoble 1, BP 53, 38041 Grenoble Cedex 9, France
33 4 76 51 44 40
{Celine.Coutrix, Laurence.Nigay}@imag.fr

ABSTRACT

Mixed interactive systems seek to smoothly merge physical and digital worlds. In this paper we focus on mixed objects that take part in the interaction. Based on our Mixed Interaction Model, we introduce a new characterization space of the physical and digital properties of a mixed object from an intrinsic viewpoint without taking into account the context of use of the object. The resulting enriched Mixed Interaction Model aims at balancing physical and digital properties in the design process of mixed objects. The model extends and generalizes previous studies on the design of mixed systems and covers existing approaches of mixed systems including tangible user interfaces, augmented reality and augmented virtuality. A mixed system called ORBIS that we developed is used to illustrate the discussion: we highlight how the model informs the design alternatives of ORBIS.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**] Theory and methods, User-centered design. D.2.2 [**Design Tools and Techniques**] User interfaces

General Terms

Design, Human Factors.

Keywords

Mixed Systems, Mixed Objects, Augmented Reality, Tangible User Interfaces, Design Space.

1. INTRODUCTION

Mixed interactive systems seek to smoothly merge physical and digital worlds. Examples include tangible user interfaces, augmented reality and augmented virtuality. The design of such mixed systems gives rise to further design challenges due to the new roles that physical objects can play in an interactive system. The design challenge lies in the fluid and harmonious fusion of the physical and digital worlds. Addressing this challenge, in [7], we introduced the Mixed Interaction Model: Our contribution is a new way of thinking of interaction design with mixed systems in terms of mixed objects, putting on equal footing physical and digital properties of an object since combining physical and digital worlds is the essence of mixed systems. In mixed systems, a mixed object is involved in the interaction. As identified in our ASUR (Adapter, System, User, Real object) design notation [8] for mixed systems, an object is either a tool used by the user to perform her/his task or the object that is the focus of the task (i.e., task object).

In this paper, we focus on the physical and digital properties of a mixed object in the light of our mixed interaction model. We present a new characterization space of the physical and digital properties of a mixed object from an intrinsic viewpoint. Intrinsic characteristics of a mixed object are independent of its context of use. Intrinsic properties can then be applied to an object that plays the role of a tool or of a task object in the interaction. By characterizing mixed objects, we enrich our model by providing a better and unified understanding of the design possibilities.

The paper is organized as follows: We first present the main features of ORBIS, a mixed system that we designed and developed. ORBIS is used to illustrate our intrinsic characterization space. We then recall the key elements of our model before presenting the intrinsic characterization scheme of a mixed object. We illustrate it by considering a mixed object in ORBIS. We finally consider related studies and show how our characterization scheme unifies existing approaches.

2. ILLUSTRATIVE EXAMPLE: ORBIS

ORBIS is a system providing new ways to enjoy personal pictures, music and videos in a family house. As part of a multidisciplinary project involving HCI researchers, computer scientists and a product designer, we designed and developed the functional prototype of Figure 1. The list of personal media is to be imported beforehand in the system. In the first version of the system, we only consider pictures. Pictures are embedded in a silicone object (Figure 1-a), displayed as a slideshow through a mini screen and are always correctly displayed according to the orientation of the silicone shape (Figure 1-b), thanks to embedded accelerometers. This mixed object is called “List of pictures”.

-a- -b-

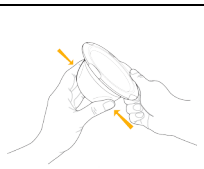
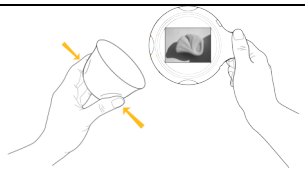
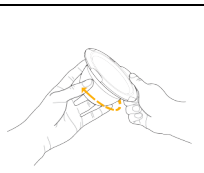
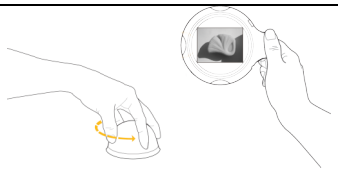
Figure 1: -a- ORBIS prototype. -b- Rotating the mixed object.

ORBIS then allows the user to perform tasks including play/pause the presentation, shuffle or navigate the list of pictures (Table 1) by interacting with the mixed object. For example, to play/pause

AVI'08, May 28-30, 2008, Napoli, Italy.

the presentation, the user presses a tool. This action is sensed by a balloon fixed to an atmospheric pressure sensor. To navigate the pictures, the user rotates the tool where a potentiometer is embedded. We considered different solutions, for example a design with only one mixed object (Table 1, left column) that plays the role of both task object and tool, and another one with two distinct objects (Table 1, right column). Table 1 shows different design solutions for interacting with the ORBIS mixed object "List of pictures". These are examples to show how the mixed object can be used in ORBIS. Nevertheless in the rest of the paper, we focus on the mixed object "List of pictures" from an intrinsic point of view without considering its context of use.

Table 1: Interacting with the mixed object "List of Pictures" in ORBIS: different design solutions.

Play/Pause		
Navigate		

3. MODELING OF A MIXED OBJECT

The key concept of the Mixed Interaction Model is a mixed object. The Mixed Interaction Model enables us to model both mixed objects and interaction with them. We recall here the main principles of the model for defining a mixed object only, since we focus on intrinsic characteristics of an object without considering the interaction with it.

3.1 Definition

Objects existing in both the physical and digital worlds are depicted in the literature as mixed objects [4], augmented objects or physical-digital objects, but there is no precise definition of such objects. In the Mixed Interaction Model, a mixed object is defined by its physical and digital properties as well as the link between these two sets of properties. The link between the physical and the digital parts of an object is defined by *linking modalities*. We base the definition of a *linking modality* on that of an *interaction modality* [17]: Given that d is a physical device that acquires or delivers information, and l is an interaction language that defines a set of well-formed expressions that convey meaning, an *interaction modality* [17] is a pair (d,l), such as *(camera, computer vision)* or *(microphone, pseudo natural language)*. We reuse these two levels of abstraction, device and language. But as opposed to *interaction modalities* used by the user to interact with mixed environments, the modalities that define the link between physical and digital properties of an object are called *linking modalities*. There are two types of linking modalities that compose a mixed object: An input linking modality (d_i,l_i) is responsible for (1) acquiring a subset of *physical properties*, using a device d_i (input device), (2) interpreting these acquired physical data in terms of *digital properties*, using a language l_i (input language). An output linking modality is in charge of (1) generating data based on the set of *digital properties*, using a language l_o (output language), (2) translating these generated physical data into perceivable *physical properties* thanks to a device d_o (output device).

As an example of a mixed object, we consider the list of pictures in ORBIS presented in Figure 1 and modeled in Figure 2. Two accelerometers each acquire 1D acceleration from physical properties. The resulting data are combined: for the composition of linking modalities at both device and language levels, we reuse the CARE properties [17]. The input linking language then translates the resulting combined data into the digital property `top`, which can have four possible values corresponding to each side of a picture. Figure 1 illustrates this process by showing how the changes of physical properties (rotation of the mixed object) impact on the digital properties of the object (orientation of the displayed picture) thanks to the linking modalities. The output linking language translates the digital properties of the object (Figure 2) in order to present the list of pictures as a slideshow. Finally the device of the output linking modality (i.e., the mini screen in Figure 1 and 2) makes the slideshow perceivable by the user.

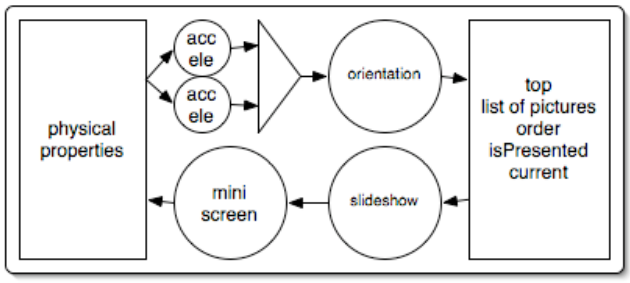


Figure 2: Mixed object "List of pictures" in ORBIS.

3.2 Intrinsic Characteristics

The intrinsic characterization space is based on two orthogonal axes that describe the physical and digital properties of a mixed object.

3.2.1 Sensed/Generated Physical Properties

We consider physical properties independently of the linking modalities. Without specifying the linking modalities, a physical property can be sensed or not by an input linking modality, and generated or not by an output linking modality, as shown in Figure 3.

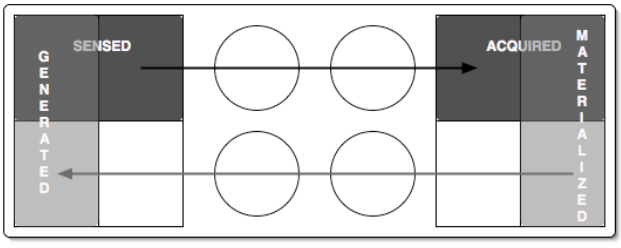


Figure 3: Characterization of the physical and digital properties of a mixed object.

In order to take into account the user in the design process, we relate the perceived affordance [12] of physical properties, cultural constraints and predictability [1] to the sensed physical properties. Affordance [12] is defined as the physical properties the user can act on. Cultural constraints are conventions shared by users from a same cultural group. For example, if a ball has the

appearance of a soccer ball, this suggests to the users to hit the ball with their feet. Such actions, called expected actions in [3], should then correspond to sensed physical properties to ensure partial predictability [1]. The complete predictability will then be ensured by designing the proper input linking modality.

To fully illustrate the Sensed/Generated physical properties of Figure 3, we consider the example of the NAVRNA system, a system that we have designed and developed for the manipulation of ARN molecules [2]. Biologists move blue tokens around a table instrumented with camera and projector (Figure 4). The physical position of a token is sensed by the video camera. Biologists explore (move, turn, resize) molecules shown as a graph projected on the table.

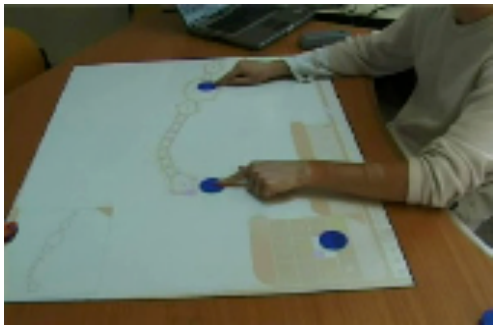


Figure 4: NAVRNA.

Physical properties taken into account in the NAVRNA tool, i.e. blue token, include the physical position and the color of the tokens. Instead of having a non-generated color, we can envision generating the color of the token as a feedback of the sensed physical position, as in [14]. Nevertheless in NAVRNA the color is sensed by the computer vision linking modality. We could then change the linking modality and consider infrared as in [11][14]. In that case, the color of a token, which was initially a Sensed/Non Generated physical property, is now a Non Sensed/ Generated physical property. Considering the second physical property, the position of the tokens, we may decide that when the user moves a molecule, all tools (and therefore tokens) move accordingly: The physical property of a token, its position, which was Sensed/Non Generated, is now Sensed/Generated, as in [14][15]. Identifying such a physical property during the design phase leads the designer to decide the protocol for modifying this shared resource (i.e., the physical property). For example in [14], a mode is used: the object is either in sensing or generating mode. As shown with the NAVRNA example, the two characteristics Sensed/Generated of a physical property allow the designer to systematically explore the design space independently of the linking modalities and therefore the technological considerations.

3.2.2 Acquired/Materialized Digital Properties

In a symmetric way, digital properties can be acquired or not, and materialized or not. In order to take into account the user in the design process at the digital properties level, we may relate the materialized characteristic to the observability property [1]. By considering the same example, NAVRNA, designers may have a top-down approach, starting from the digital side. The digital property is `[x,y]`. It is an acquired digital property as explained above. For enhancing the observability of the state of the object, the property can be materialized for example by projecting a color on top of the token as in [14]. The digital property is then Acquired/Materialized.

4. INTRINSIC DESIGN OF A MIXED OBJECT: ORBIS EXAMPLE

Physical and digital properties of a mixed object are characterized by two orthogonal design axes, respectively Sensed/Generated and Acquired/Materialized as schematized in Figure 3. The characterization scheme does not constrain the order of design activity. On the one hand, the design approach can be bottom-up, starting from a physical object with a set of physical properties and then defining its generated physical properties as well as its sensed physical properties, before deciding the linking modalities. On the other hand, the approach can be top-down starting by a set of digital properties and defining the acquired and materialized digital properties, as in the ORBIS example. Going back and forth, considering alternatively the physical properties and the digital properties in the light of our characterization scheme defines a smooth combination of bottom-up and top-down design approach of a mixed object. We illustrate this point by considering the design of the "list of pictures" object in ORBIS.

In the context of the design of ORBIS, the list of pictures is originally a digital object. As we wanted it to be more anchored in the physical world, we designed it as a mixed object. The first obvious digital property is the digital `list of pictures (Image 0, …, Image n)`. We identify further digital properties attached to it: The `order` of the pictures, initially arranged `(0, …, n)`, the boolean digital property `isPresented`, initially `false`, and `current`, initially `0`. Digital properties can be acquired and/or materialized. In this case of purely digital pictures (non-acquired), we decided to materialize these digital properties by choosing the *(mini-screen, slideshow)* modality. Based on this digital part of the object, we explore alternatives for linking devices and languages (i.e., linking modalities) in order to augment this object with a physical part. Physical properties can be sensed/generated or not by linking modalities. The design choice of physical properties neither sensed nor generated were driven by aesthetic and portability requirements, such as the silicone shape around the screen (Figure 1). We also consider a physical property to be sensed, such as the top of the silicone shape, since we want the picture to be always correctly displayed according to the orientation of the silicone shape (Figure 1). Thus we need to define an input linking modality, linking the physical to the digital `top` of pictures. The non-generated physical property i.e. the top of the silicone shape is sensed by an input linking modality, such as *(accelerometers, orientation)*. The input linking modality being defined, a new digital property is identified, having four values corresponding to the four possible sides of a picture. This new digital property is acquired thanks to the input linking modality, as opposed to the other digital properties that are not acquired. Figure 2 shows the corresponding design, with an input linking modality based on accelerometers as well as the acquired digital property, `top`.

5. RELATED WORK

The Sensed/Generated and Acquired/Materialized characteristics of the physical and digital properties generalize the *Input & Output* axis presented in [9], the characterization of physical properties in MCRit [16] and the sensed movements in [3].

- First, the *Input & Output* axis [9] characterizes the system inputs and outputs without considering the two levels of a linking modality, device and language, as well as the two types of properties physical and digital. These levels of

abstraction are also presented in [5] and [10]. For example, we refine "Light (photoelectric cell)" from [9] into: the sensed physical luminosity, the input linking modality *(photoelectric cell, language-filter)*, and resulting digital properties. Such a refinement helps explore the design possibilities by systematically considering the design choices at each level of abstraction.

- Second, MCRit [16] splits the output of the system between tangible and intangible representation. Our model extends this definition by considering both inputs and outputs. Moreover since our framework is not dedicated to tangible UI only, we consider tangible and non-tangible mixed objects. For example, an object superimposed on the physical world through semi transparent glasses is mixed, but not tangible.
- Finally in the framework for designing sensing-based interaction [3], sensed movements can be related to the sensed properties of a mixed object: the sensed movements/properties that are measured by a computer. Our model extends this notion by also considering the generated physical properties. Moreover our model not only considers the physical properties but proposes a symmetric analysis of the digital properties.

6. CONCLUSION

Based on our Mixed Interaction Model, we introduce a new characterization space of the physical and digital properties of a mixed object from an intrinsic viewpoint. Our intrinsic characterization scheme unifies existing design spaces while extending them. According to [13], it proves the usefulness of our model that facilitates interconnection between existing approaches. Moreover the model has been used to analyze existing mixed systems. We currently do not find examples of design solutions in the literature that our model left out. This proves that the model could be used to design a wide and relevant range of mixed objects since reverse engineering was possible. This demonstrates the soundness of the underlying concepts of the model. More importantly the modeling of existing systems enables us to describe in detail the systems and to make a fine distinction between them. As a benchmark, we chose similar interfaces like NAVRNA [2], IRPhicon [11] and the Actuated Workbench [14]. Differences between them are not obvious: in all of them the user interacts by moving an object on a surface. Applying our model and its intrinsic characteristics, we were able to make a fine distinction between these interfaces, where other taxonomies only partially capture these differences. This shows that our model provides a useful framework for better understanding existing mixed systems.

Going further than describing and classifying existing mixed systems, in order to assess if the model is useful for design, we use another form of empirical evaluation: we applied the model in real design situations. Although we presented here only one example, the model has been used to design new mixed systems such as ORBIS, RAZZLE [7] or Snap2Play [6] with real end users of the model, i.e. the designer and the software engineer, not the authors of the model. As on-going work, we are currently further evaluating the model by considering three groups of designers in the context of a mixed system for museum exhibits: one group working with this model, another with the ASUR model [8], and a third group without any model.

7. REFERENCES

[1] Abowd, G., Coutaz, J., Nigay, L., 1992. Structuring the Space of Interactive System Properties. In proceedings of EHCI'92, North Holland Publ., 113-130.

[2] Bailly, G., Nigay, L., Auber, D., 2006. NAVRNA: Viusalization-Exploration-Edition of RNA. In proceedings of AVI'06, ACM Press, NY, 504-507.

[3] Benford, S., et al., 2005. Expected, Sensed, and Desired: A Framework for Designing Sensing-Based Interaction. ACM TOCHI, 12, 1 (March 2005), 3-30.

[4] Binder, T., et al., 2004. Supporting Configurability in a Mixed Media Environment for Design Students. Springer Personal and Ubiquitous Computing, 8, 5 (Sept. 2004), 310 - 325.

[5] Buxton, W., 1983. Lexical and pragmatic considerations of input structures. ACM SIGGRAPH Computer Graphics, 17,1 (Jan. 1983), 31-37.

[6] Chin, T., You, Y., Coutrix, C., Lim, J., Chevallet, J.-P., Nigay, L., 2008. Snap2Play: A Mixed-Reality Game based on Scene Identification. In proceedings of ACM IEEE MMM'08, LNCS, Springer, 220-229.

[7] Coutrix, C., Nigay, L., 2006. Mixed Reality: A Model of Mixed Interaction. In proceedings of AVI'06, ACM Press, NY, 43-50.

[8] Dubois, E., Nigay, L., Troccaz, J., 2001. Consistency in Augmented Reality Systems. In proceedings of EHCI'01, LNCS, Springer, 117-130.

[9] Fitzmaurice, G., Ishii, H., Buxton, W., 1995. Bricks: Laying the foundations for Graspable User Interfaces. In proceedings of CHI'95, ACM Press, NY, 442-449.

[10] Mackinlay,J., Card, S., Robertson, G., 1990. A Semantic Analysis of the Design Space of Input Devices. Lawrence Erlbaum HCI, 5, 2&3 (1990), 145-190.

[11] Moore, D., Want, R., Harrison, B., Gujar, A., Fishkin, K., 1999. Implementing Phicons: Combining Computer Vision with InfraRed Technology for Interactive Physical Icons. In proceedings of UIST'99, ACM Press, NY, 67-68.

[12] Norman, D., 1999. Affordance, Conventions and Design. ACM Interactions, 6 ,3 (May-June 1999), 38-43.

[13] Olsen, D., 2007. Evaluating user interface systems research. In proceedings of UIST'07, ACM Press, NY, 251-258.

[14] Pangaro, G., Maynes-Aminzade, D., Ishii, H., 2002. The Actuated Workbench: Computer-Controlled Actuation in Tabletop Tangible Interfaces. In proceedings of UIST'02, ACM Press, NY, 181-190.

[15] Patten, J., Ishii, H., 2007. Mechanical Constraints as Computational Constraints in Tabletop Tangible Interfaces. In proceedings of CHI'07, ACM Press, NY, 809-818.

[16] Ullmer, B., Ishii, H., Jacob, R., 2005. Token+constraint systems for tangible interaction with digital information. ACM TOCHI, 12, 1 (March 2005), 81-118.

[17] Vernier, F., Nigay, L., 2000. A Framework for the Combination and Characterization of Output Modalities. In proceedings of DSVIS'00, LNCS, Springer, 32-48.

A FLEXIBLE, DECLARATIVE PRESENTATION FRAMEWORK FOR DOMAIN-SPECIFIC MODELING

Tamás Mészáros Gergely Mezei Tihamér Levendovszky
Budapest University of Technology and Economics
1111 Budapest, Goldman György tér 3., Hungary
+36-1-463-2870
mesztam@aut.bme.hu gmezei@aut.bme.hu tihamer@aut.bme.hu

ABSTRACT
Domain-Specific Modeling has gained increasing popularity in software modeling. Domain-Specific Modeling Languages can simplify the design and the implementation of systems in various domains. Consequent domain specific visualization helps to understand the models for domain specialists. However, the efficiency of domain-specific modeling is often determined by the limited capabilities – i.e. the lack of interactive design elements, low customization facilities – of the editor applications.

This paper introduces the Presentation Framework of Visual Modeling and Transformation System, the framework provides a flexible environment for model visualization and provides a declarative solution for appearance description as well.

Categories and Subject Descriptors
D.2.2 [**Software Engineering**]: Design Tools and Techniques;

General Terms
Design and Languages.

Keywords
Domain-specific modeling, software modeling, metamodeling, modeling framework, model visualization

1. INTRODUCTION
It has turned out that general purpose modeling languages are often hard and inflexible to use in software and hardware modeling. Generality is the greatest advantage of these languages and their greatest drawback at the same time. Applying general modeling languages to model domain-specific problems can result in inflexible models that completely lack the specialties of the target domain. A possible solution to solve this problem is metamodeling. Metamodels are the models of languages, they define language elements, the attributes of the elements and the possible relations between them. However, metamodeling is not meant to describe the visual representation, or the editing behavior of modeling items. In order to support domain-specific modeling by metamodeling, we require an editing framework that allows constructing metamodels, supporting instantiation of these metamodels and offers an easy-to-use way to define customized visualization (editing behavior) as well.

Recent solutions of model visualization environments (Section 2) were mainly building on procedural coding. Tools and visual languages built for designing model element appearances may ease this process, however, today's applications provide only the most basic features (e.g. basic primitives and subtitles). Our aim was to provide a framework which simplifies this process by using declarative solutions. The increased interactivity of languages makes easier the editing of the model as well. Visual Modeling and Transformation System (VMTS) [22] is an n-layer metamodeling and model transformation environment. N-layer means in this context that there is no limitation for the number of modeling layers. VMTS has successfully been used in various domain-specific modeling areas including but not limited to resource editors for mobile phones, or communication network modeling.

In this paper, we introduce the VMTS Presentation Framework, a flexible and highly configurable environment for graphical model visualization and editing. Although the Presentation Framework is not a new component in VMTS [2], we have completely redesigned and optimized the architecture and rewritten the implementation. The new Presentation Framework supports defining the metamodels in a visual way, while the domain-specific appearance can be defined by declarative constructs. Although the presented approach has been implemented in our metamodeling system, the solutions presented in the paper can be used in any other (meta)modeling framework.

2. RELATED WORK
Eclipse [7] is a highly generic modeling environment. Eclipse Graphical Editing Framework (GEF) [7] is an open source infrastructure for creating and using graphical editors based on Eclipse. In contrast to VMTS Presentation Framework, the underlying architecture of GEF is the Model-View-Controller [6] pattern. The visualization mechanism of elements relies on the Java2D package, which builds on bitmap operations. The updating of an element must be initiated manually. GEF provides a flexible framework for building graphical diagram editors, however, even the basic features (e.g. event propagation, change notification, repainting and serialization) should be implemented manually without tool support.

Graphical Modeling Framework[7] (GMF) is also an Eclipse project, and it utilizes GEF. Compared to GEF, GMF uses Eclipse Modeling Framework[7] (EMF) models as the underlying model

AVI'08, May 28–30, 2008, Naples, Italy.

object. EMF facilitates the serialization of models in various formats and the change notification mechanism is also a built-in feature. The editor environments in GMF are generated based on models describing the appearance and the mapping. The creation of models and generation is supported by the GMF Dashboard wizard. However, this tool provides very basic features only in defining visual models. Complex visualization requires to be defined using Java In VMTS, we do not need to generate the editor, because the environment can be extended at *runtime* based on the plugin-mapping and the metamodel. Compared to GMF, we do not need procedural coding in most cases either because of the declarative interface definition.

TIGER[10] (Transformation-Based Generation of Modeling Environments) utilizes the Eclipse Modeling Framework, and Eclipse Graphical Editing Framework is used as the target visual environment. The appearance of visual language elements can be defined in a formal way by modeling, but the number of applicable primitives is strongly restricted. Language elements can also contain dynamic content (text). The visual language description models are transformed into source code using graph transformation provided by AGG[1]. The resulting source code can be compiled into an Eclipse plugin, providing the same advantages and limitations of the Eclipse and GEF platform.

DiaGen [6] (Diagram Editor Generator) is a framework for generating graphical diagram editors. It uses hypergraph grammars to specify visual languages. Editors generated with DiaGen are capable of running online structural and syntactic analysis on the edited models. DiaMeta [8] is the successor of DiaGen. It employs the Eclipse Modeling Framework for MOF[11] to define visual languages. The generated editors have the same properties as those generated with DiaGen. Compared to DiaMeta, VMTS Presentation Framework is more flexible especially in defining dynamic behavior of model elements. Complex visualization of elements needs manual procedural coding, as it is poorly supported by the tool.

Generic Modeling Environment (GME)[9] is a general purpose metamodeling environment for creating modeling a program synthesis tools. Concrete syntax in GME can be defined by implementing COM components. Notifications about changes in the model have to be handled manually. GME does not support automatic layout of elements, furthermore, connections in GME cannot be customized.

Microsoft Visio [12] is a modeling tool for creating various types of diagrams. The visual elements can be extended with the help of external templates. Visio is not based on metamodeling and its models are not meant to be processed algorithmically. Visio is mainly a drawing tool, it had a great influence on our work though, by introducing an easy-to-use but efficient user interface, which is usually a neglected feature of other modeling tools.

3. THE PRESENTATION FRAMEWORK

3.1 Basics

When designing the VMTS Presentation Framework, one of the most important design goals was to provide an easy-to-use and flexible framework to define visual languages. Our solution provides high customizability and the possibility to create interactive elements in a simple way. The VMTS Presentation Framework is based on Windows Presentation Foundation (WPF) [15]. In our approach, every language element is represented as a Windows control with a special, customized appearance. Consequently, we can rely on the event handling mechanism of the control management system provided by the operating system. Since controls in WPF are highly flexible and customizable, using Windows controls as the basis of model elements does not limit the visualization or event handling capabilities of our framework. The update and repainting mechanism is also inherited from Windows controls achieving a remarkable performance gain due to the optimized performance of WPF. We have placed much effort in facilitating visual design in a declarative way. Language items can be defined with the help of the powerful XAML [15] language which is an XML based user interface descriptor language. By using XAML, we can describe not only the static appearance but the dynamic behavior of an element as well. Using the data binding mechanism of WPF together with generated *Attribute Wrapper* objects (see section 3.5 for details) object properties can be visualized without a single line of imperative program code.

3.2 Architecture

One of the most important user requests was to support multiple views of models and to provide extensibility so that different models can be visualized in a customizable way. Different views are required to be able to focus simultaneously on different aspects of models, while customization is needed to support domain-specific languages. The Model-View architecture is an appropriate choice to satisfy this requirement. For each model element, VMTS associates exactly one *Model* and an arbitrary number of Views. To facilitate the customization of the appearance, we have designed a flexible plugin system. Each plugin is attached to a metamodel with the help of the unique identifier of the (meta) model. Metamodels can have more than one plugins attached, the user can select the plugin to use at run-time. With the help of plugins, we can customize not only the *Views* of a model, but the behavior of *Model* (e.g. persistence) or the user interface of the as well.

The main components of the plugin-based system are the *BaseModel* and *BaseView* classes (following the Model-View architecture). In VMTS, models are represented as directed, attributed graphs consisting of nodes and edges. Following this distinction, we have two types of model elements: shapes and lines with own *Model* and *View* implementations. An additional model element is the *Diagram* itself, which represents the entire model. The visual appearance of model elements can be defined in two ways: (i) creating a *ShapeView* or *LineView* descendant class implementation, where we can describe the appearance and the behavior as well. (ii) using XAML code. The XAML descriptor language is capable of describing customized appearance based on simple graphical constructs (e.g. rectangle, lines) and hierarchical control structures (e.g. tree control). Furthermore, it is also appropriate to describe the animation of the elements based on event triggers. Recall that language elements are visualized with the help of customized Windows (WPF) controls. Using *Control Templates* in WPF is an appropriate choice to override the appearance of controls with the help of XAML. *Control Templates* for different model elements can be placed in the same XAML file, which has to be copied next to the main executable. Editing or replacing this file at run-time will also affect the visualized diagrams in VMTS, *without recompiling the sources*.

XAML is easy to generate and to edit due to the hierarchic structure, and there already exist external tools for this purpose (e.g. Expression Blend from Microsoft)

3.3 Object Hierarchy

Model elements have references on their container and on their contained items. Therefore, the containment hierarchy forms a double-linked object tree. This tree (referred to as Model-based hierarchy) reflects the hierarchy of the model elements on the model repository level. However, a similar hierarchy is maintained between *View* objects, where the containment relation is based on visual containment. This hierarchy is referred to as the View-based hierarchy. The two hierarchies are not necessarily the same: (i) It is possible to construct a *DiagramView* that does not contain all model elements in order to handle complex models. (ii) The container item of an object can be different in the Model-based and in the *View*-based hierarchies. The reason is that in our approach, we can visualize a sub-level in the model hierarchy without building the complete containment hierarchy back to the root. Note that this feature is useful for example in UML Class diagrams, where Packages can be created and each Package can contain a set of classes. The same class elements can be visualized as children of packages and in another view they can be shown without showing the complete containment hierarchy.

In order to handle this issue, we distinguish primary and secondary views. Modifications performed on primary views affect the original (model repository-level) model hierarchy as well. Primary views always reflect the modeling hierarchy. At most one primary view can exist for a model at the same time, any other view is secondary and it is used for visualization purposes only. A secondary view cannot influence the model-based hierarchy, but it does not have to strictly reflect it either.

3.4 Containment Visualization

In the VMTS Presentation Framework, we use *Workspaces* to represent a container canvas for contained model elements. Each *Workspace* has its own transformation coordinate system including customized position information, rotation and zoom.

A traditional *Workspace* consists of two layers: a *Root Canvas* and a *Root Panel*. The *Root Panel* can perform automatic layout transformations on the contained controls (e.g. docking, stretching, table layout). The *Root Canvas* is necessary to represent a common control parent for both the *Root Panel* and the elements which do not require automatic layout. Automatic layouting is not required, for example, by lines or selection controls. When connecting two shapes in a model with a line, the framework traverses the visual containment trees starting at the target shape, and the first common container object is selected to be the visual container of the line. However, when connecting two shapes in different *Workspaces*, but contained by the same shape (e.g. the Composite State element of UML State Chart diagrams), it is not possible to find a common *Root Canvas* in the visual hierarchies to contain the line (inside the common logical container shape). For this purpose, we have invented *Workspace* hierarchies. *Workspace* hierarchies can be at most two levels deep. The unique root element within a shape is called *Main Workspace*, and the child *Workspaces* are called *Sub-Workspaces*. The Main Workspace is intended to contain line objects which can interconnect shapes in two different *Sub-Workspaces*. The maximum number of the possible *Workspace*-levels does not mean any practical limitation, because the *Main Workspace* can always be used to contain all elements which have to be shown over all *Sub-Workspaces*.

Recall that, VMTS can also visualize parts of the model hierarchy, which parts do not necessarily contain the *DiagramModel*. For this purpose we facilitate to use a model element as the root container view. The appearance of an element may differ when placing it inside another element (or directly on the *DiagramView*) or using it directly as the root container in the window. The appearances can be distinguished in the mapping between the model element and the visualization.

3.5 Visualizing Model Attributes

Recall that in VMTS, models are represented as directed, attributed graphs. Model elements correspond to graph nodes, while relations between elements are defined as edges. Unlike several modeling environments, in VMTS edges can also have attributes. In order to support n-layer metamodeling and to support customizable domain-specific attributes, we have defined a highly flexible formal attribute structure [5] based on attributes and attribute types. The attributes of a metamodel element determine the possible attribute structure of the associated model level elements, similarly to UML Class diagram – Object diagram, where classes define the possible attributes of their instantiations, the objects. In other words this means that the attributes of metamodel elements define a schema for the model level attributes. The attributes of model level elements are forced to follow this schema automatically.

In the current version of VMTS, attributes are represented by classes inherited from a common *AttributeBase* class. Attributes containing other attributes are supported by using the Composite design pattern [6]. We have implemented the *Attribute Panel* control for Presentation Framework that allows editing Attributes and validates them against the schema. If a model element is selected, the *Attribute Panel* obtains the metamodel of the element and builds a schema (the *AttributeTemplate*) from the metamodel attributes. Then, *Attribute Panel* loads the attribute configuration of the model element and validates the attributes according to the schema. Besides editing and schema-based validation, it is also important to offer an easy-to-use way to visualize the attribute values of model elements. WPF provides a data binding mechanism that can bind property values of an object (e.g name property of a model element) to a property of a UI element (e.g. *Text* property of a *TextBox* control). To utilize the built-in data binding and change notification, we generate a wrapper class (*AttributeWrapper*) for each metamodel element at run-time using Intermediate Language [13]. The *AttributeWrapper* facilitates to reach model attributes by navigating on its generated properties which have the same name as the attributes have. As data binding is bi-directional, we can also edit model properties with the bound controls on the language element. Attribute wrappers and the data shown in Attribute Panel are automatically synchronized.

3.6 Persistence

Persisting visual information is essential to reconstruct the different views of a model properly. We have designed a powerful externalization framework that (i) provides a transparent mechanism to store data without any assumption of its low-level

representation, and (ii) simplifies the attribute-externalization process with the help of declarative language constructs. The persistence engine consists of two main components: a general object store (*ExternalizerStore* class) and a data formatter (*Externalizer*). The data formatter component is used to perform bi-directional conversion between the original data format and its low level representation (e.g. binary stream). The general object store is capable of maintaining tagged value lists, where values are identified by unique string tags and can be of any type. Tags can also identify an additional object store, thus we can build object trees in an arbitrary depth. There was a need to provide a space-efficient but general solution to persist data. For this purpose, we do not store type information in the serialized data, it contains tag-value pairs only. Consequently, the proper data types are recovered right before accessing the data. As a result, the objects have to be encoded using the *Externalizer* before placing it in the store. Plugin developers can directly use the store to access the objects, because the actual *Externalizer* is applied by the store implicitly. The fact that the store is filled with data already encoded, does not mean any performance issue, since encoding/decoding a tag-value pair is usually applied only before saving the model to, or loading the model from disk. The *Externalizer* provides a flexible extension mechanism. By using this mechanism, custom converters can be applied in addition to the built-in ones. Moreover, classes implementing the *IExternalizable* interface can have a direct influence on their serialization. The member attributes and properties of an object implementing the *IExternalizable* interface can be selected to be persisted by marking them with the *Externalizable* attribute. By implementing the *Externalize* and *Internalize* methods we can also extend the pure declarative notation.

In VMTS, we have provided a reference implementation for the approach. This implementation contains an XML *Externalizer* which converts tags into XML tags and values into the bodies of the XML tags. The implementation provides the conversion of basic types and of numerous complex types (e.g. *Point*, *Vector*, *Color*) that are usually needed to describe visual appearance.

4. CONCLUSIONS

By the increasing complexity of software systems, the model-based software development became essential. The high level of abstraction and the visual representation of models can definitely help in solving complex engineering problems. The increasing popularity of Domain-Specific Modeling Languages has attracted the focus on creating modeling frameworks capable of defining, visualizing and editing domain-specific models in a flexible, yet user-friendly way.

The design time architectural decisions and the key features of the Presentation Framework were presented in this paper including the applied architectural patterns, element and containment visualization. We have also introduced a flexible attribute editing solution and the applied serialization mechanism. We have facilitated creating domain-specific plugins using purely declarative syntax, while imperative constructs are also available. In addition to static appearance we can also define dynamic behavior including various interactive features. Visualizing and editing models using different plugins is possible as well as creating sub-views of arbitrary branches of the model level containment tree. The persistence layer provides a declarative way to externalize data. It keeps modeling objects and final data format completely independent allowing us to use various model factory formats.

Future work includes the definition of a visual language for appearance description, thus we could generate the VMTS plugin projects, the mapping and the XAML code automatically. We are also working on a discrete simulation environment for modeling dynamic model behavior and user interface animation.

5. ACKNOWLEDGMENTS

The found of “Mobile Innovation Centre” has supported, in part, the activities described in this paper. This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

6. REFERENCES

[1] Taentzer G: AGG: A Graph Transformation Environment for Modeling and Validation of , In J. Pfaltz, M. Nagl, and B. Boehlen (eds.), Application of Graph Transformations with Industrial Relevance (AGTIVE'03), volume 3062. Springer LNCS, 2004.

[2] G. Mezei, T. Levendovszky, H. Charaf, A Presentation Framework for Metamodeling Environments. *WISME*, Montego Bay, Jamaica, October, 2005.

[3] J. de Lara, H. Vangheluwe, ”AToM3 as a Meta-Case Environment”, International Conference on Enterprise Information Systems, 2002

[4] G. Mezei, T. Levendovszky, and H. Charaf, Attribute Algebra for N-layer Metamodeling. *Proc. of WSEAS Int.Conf. on Applied Informatics and Communication*, Athens, Greece, 2007. 144-150

[5] M. Minas. *Concepts and Realization of a Diagram Editor Generator Based on Hypergraph Tranformation.* Science of Computer Programming, 44(2):157-180, 2002

[6] E. Gamma, R. Helm, R. Johnson, J. Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley Professional Computing Series)

[7] Eclipse home page : http://www.eclipse.org

[8] Mark Minas. *Generating Visual Editors Based on Fujaba/MOFLON and DiaMeta.* Proc. 4th Fujaba Days, pp. 35-42, 2006. Technical Report tr-ri-06-275 (University Paderborn).

[9] GME homepage: http://www.isis.vanderbilt.edu/projects/gme

[10] Ehrig, K., Ermel, C., Hansgen, S., Taentzer, G.: Generation of Visual Editors as Eclipse Plug-Ins http://www.tfs.cs.tu-berlin.de/~tigerprj

[11] Meta Object Facility (MOF) http://www.omg.org/mof

[12] Microsoft Visio homepage : http://office.microsoft.com/visio

[13] K. Burton, .NET Common Language Runtime Unleashed, Sams Press, 2002

[14] Visual Modeling and Transformation System http://vmts.aut.bme.hu

[15] Windows Presentation Foundation (WPF) http://msdn2.microsoft.com

Advanced Visual Systems Supporting Unwitting EUD

Maria Francesca Costabile*, Piero Mussio°, Loredana Parasiliti Provenza°, Antonio Piccinno*

*Dipartimento di Informatica, Università di Bari, ITALY
°DICO, Università di Milano, ITALY
{costabile, piccinno}@di.uniba.it, {mussio, parasiliti}@dico.unimi.it

ABSTRACT
The ever increasing use of interactive software systems and the evolution of the World Wide Web into the so-called Web 2.0 determines the rise of new roles for users, who evolve from information consumers to information producers. The distinction between users and designers becomes fuzzy. Users are increasingly involved in the design and development of the tools they use, thus users and developers are not anymore two mutually exclusive groups of people. In this paper types of users that are between pure end users and software developers are analyzed. Some users take a very active role in shaping software tools to their needs, but they do it without being aware of programming, they are *unwitting* programmers who need appropriate development techniques and environments. A meta-design participatory approach for supporting unwitting end-user development through advanced visual systems is briefly discussed.

Categories and Subject Descriptors
H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces – *Asynchronous interaction, Organizational design, Theory and models*

General Terms
Design, Experimentation, Human Factors, Theory.

Keywords
End user, end-user development, user classification.

1. INTRODUCTION
Users and developers have been traditionally considered two distinct communities: users are the "owners" of the problems and developers are those who implement software systems for supporting users to solve the problems. Nowadays, with the widespread use of software systems and the evolution of the World Wide Web toward the so called Web 2.0, more and more people do not only use software but also get involved in designing and developing software. Such users deploy various pre-programming and programming activities, ranging from simple parameter customization to variation and assembling of components, creating simulations and games. In this paper, we analyze different types of users that are between pure end users and professional software developers. A particular type of end users refers to those that are very active in shaping software tools to their needs without being aware of programming. Similarly to what it is done in [7] for children, we call them *unwitting* programmers and we will explain why in Section 2. We then discuss how the meta-design participatory approach described in [3] [4] is refined for designing advanced visual systems that can support the activities of unwitting programmers.

AVI'08, 28-30 May , 2008, Napoli, Italy

2. UNWITTING PROGRAMMERS
More and more end users are required to perform various activities that push them to possibly modify and/or create software artefacts in various ways: from simple customization to changing software functionalities. End-User Development (EUD) [6] [11] and End Users Software Engineering [9] are definitions that refer to these activities. Such end users are neither pure end users, nor software professionals: some of them may have certain software development skills but surely they are not interested in software per se. They might develop software to solve specific problems they own [3]. According to the activities they perform with software systems, the following types of users are distinguished:

1. End users who perform simple customization activities, such as changing colours, selecting or creating toolbars to be visualized, selecting items to be shown in a toolbar, etc., in order to adapt the software environment to their habits.
2. End users who write macros to automate some repetitive operations. These possibilities are permitted in spreadsheets.
3. End users who develop web applications, i.e., people with modest levels of experience in Web development, or in computing activities; they have possibly taken a course in HTML and create Web sites. They might be experts in a certain domain, but have poor knowledge in computer science. This type of users has been studied in [8], where an exploratory study of the application and consequences of informal design planning by such users is reported.
4. Developers using domain-specific languages, i.e., professionals in a particular domain (not in computer science) using a specific language to write programs for solving problems of their own domain. Examples are mathematicians, physicists and engineers using MatLab™, but also biologists working with systems like Biok (Biology Interactive Object Kit) [5], which extends formula languages available in spreadsheets and supports tailorability and extensions by end users through an integrated programming environment.
5. Data-intensive researchers who intensively manage stored data and create computer programs, so that they can be considered almost professional software developers, even if they do not have any software engineering background. They use general-purpose

programming languages, such as C and C++, for developing software for their own research goals. They may have attended courses on particular programming languages, but they have no formal training in software engineering. Consequently, they do not pay attention to software qualities such as security, code readability, maintenance, as software engineers usually do. Such end users are called "professional end user developers" in [10], intending with this term people who work in highly technical, knowledge-rich domains and develop software in order to further their professional goals. Examples of these end users are space scientists, research scientists, financial mathematicians.

End users and the activities they usually perform or are willing to perform with computers were previously analyzed in [1] [2] [12] (also mentioned in the introductory chapter of [6]). Two classes of end-user activities were identified, depending on whether their activities imply creating or modifying a software artefact (Class 2) or not (Class 1). More specifically, Class 1 includes activities that allow users to choose among alternative behaviours (or presentations or interaction mechanisms) already available in the application by setting some parameters; such activities are usually called parameterisation or customization or personalization. Class 2 includes all activities that imply some programming in any programming paradigm, thus creating or modifying a software artefact. End users belonging to Class 1 actually correspond to "end users that perform customization" (type 1 in the above classification), while users belonging to Class 2 include the following three categories: end users who write macros (type 2 in the above classification), Web contents developers (type 3), developers using domain-specific languages (type 4).

Fischer and Ye also analyse different types of end users, proposing a spectrum of software-related activities [12]. At the right end of the spectrum, there are the *software professionals*, i.e., software engineers that develop software systems that are used by people other than themselves. At the opposite side (left side) are the *pure end users* that passively use software systems to accomplish their daily task. End users in the middle of the spectrum are "people who have certain development skills but are not interested in software per se. They do not develop software for other people; rather they are developing software to solve specific problems that they own" [12].

We have experience of working with various end users, primarily people who are experts in a specific discipline, such as geology, mechanical engineering, medicine, etc., who are not expert at all in computer science, nor willing to be, and use computer systems for their daily work activities [4]. They do not have any development skill and do not want to be constrained by formalisms unfamiliar to their culture. They want software environments that are easily accessible and usable, that they can "tailor" to their needs, tasks and habits. They do not know what programming means, even if they use software applications that allow them to create or modify software artefacts, but they do this without being aware of programming. In this sense, they are similar to the children analyzed in [7], defined *unwitting end user programmers*. Indeed, children playing with a computer game are required reasonably sophisticated programming, but this is embedded in an intrinsically motivated activity that is perceived as something easy and fun to perform; in their opinion, it cannot be programming, which is generally perceived very difficult.

Petre and Blackwell observe that programming is not the children's goal; playing, constructing and deconstructing is their goal. They also say: "Children have always appropriated, reconfigured and customized their toys. Papert was motivated by such behaviour in the philosophy of 'constructionism' that motivate the Logo programming language" [7]. Children have used Lego, Meccano, and similar toys for doing construction. Computer-based authoring tools allow children to construct interactive simulations, animations, and games, in a manner that places a lot of emphasis on construction; thus they program by construction not by algorithm development, their intention is to play and enjoy, they are not aware of programming. Similarly, adult end users want to manipulate and tailor objects in their software environments in order to create new configurations and designs. They want to do this as part of their own activities that they are highly motivated to perform, not being aware that they are programming.

Going back to the end user classification discussed above, it is evident that programming in an unwitting way is a characteristics of several categories of those end users. Moreover, from the considerations derived in [7], useful indications for the design of software systems for adult unwitted programmers can be extracted. Specifically, children learn and do programming by composition while they tinker or play. They learn by trying things out. Similarly to many adults, when they use a new environment, they do not read tutorials, but go straight to the example or ask friends. Children enjoy a lot communicating with friends and possibly performing collaborative activities, often conducted remotely by exchanging artefacts online. The Software Shaping Workshop (SSW) design methodology we have developed, based on a meta-design approach, includes features that comply with these indications [4]. We will show with a case study how it permits the design of advanced visual systems that support unwitting end-user development.

3. A META-DESIGN APPROACH

The aim of the *Software Shaping Workshop* (SSW) methodology presented in [1] [3] [4] is to create software environments that support end users in performing their activities of interest, but also allow them to tailor these environments to better adapt them to their needs, and even to create or modify software artifacts. The latter are defined as activities of End-User Development (EUD) [6] [11]. To permit EUD activities, we have to consider a two-phase process, the first phase being designing the design environment (meta-design phase), the second one being designing the applications using the design environment. In this way, we distinguish between the design (or "shaping") work that is done by different types of end users (second phase), and the process of creating suitable environments and tools that can be applied by end users for their own design/shaping purposes.

Traditionally, the life cycle of interactive software system distinguishes between design time and use time. At *design time*, system developers (with or without user participation) create environments and tools for the *world as imagined* by them to anticipate users' needs and objectives. At *use time*, users use the system in the *world as experienced* [12]. Existing design frameworks are based on the assumption that major design activities end at a certain point after which the system enters use time. Our approach bridges these two stages into a unique

"design-in-use" continuum that permits the creation of *open and continuously evolvable systems*, which can be collaboratively extended and redesigned at use time *by users and user communities*. Moreover, meta-designed software systems not only provide the technical means for users to customize and extend the systems but also provide social and technical mechanisms to facilitate user participation and collaboration during the design activities.

Participatory Design is an approach originated in Scandinavian, which consists of the participation of end users in the design process, not only as an experimental subject but as active members of the design team. In this way, end users are not passive participant whose involvement is entirely governed be the designer. The rationale is that users are expert in the work context and a design can be affective if these experts are allowed to actively contribute to the design. Moreover, the adoption of a new system is liable to change the work context and organizational processes and it can only be accepted if these changes are acceptable to the user. The traditional Scandinavian approach stresses the involvement of end users in the design process, but it does not indicate when this involvement stops. The SSW methodology adopts a Participatory Design approach that it is not over with the release of the software, but continues throughout the whole software life cycle. Users are provided with software environments to perform design activities even at run-time.

The basic idea of this methodology is that an interactive system must be designed as a network of different software environments (also called Software Shaping Workshops), each one specific for a community of users of that system. Each workshop allows its users to interact through a visual language familiar to their culture and skills, since it reflects and empowers the users' traditional notations and system of signs. Such workshops enable domain experts, as well as HCI experts, to create and modify software artefacts by direct manipulation of visual elements.

We briefly refer to a case study to better explain our approach. The case arises from the collaboration with CIDD ("Consorzio Italiano Distribuzione Dolciaria"), a consortium of about thirty small and medium Italian companies and about twenty big partner companies, operating in the confectionery field. The consortium's main business is the confectionery distribution in the whole Italy. It provides its associated companies with a number of services, such as price lists, discounts, order management, etc. We collaborated with CIDD in order to create its Web portal of services. In CIDD, there is a chairman who is the person formally responsible for the CIDD activities according to the consortium statute. The main role with reference to the portal is played by the sales manager, who manages all the consortium activities and defines the services for the associated companies. Such companies purchase products from the partner companies and, through the portal, communicate with their customers.

The sales manager wants to tailor the software environments to be used by the associate companies, maintaining a consistent style of documentation and interaction. On its side, each company wants to define the environments to be used by their customers. This is a typical case in which the meta-design approach of the SSW methodology can be successful. The Web portal is then developed as a network of different workshops, each one specific for a community of users, where users can perform the EUD activities they require. The sales manager will work in a meta-environment

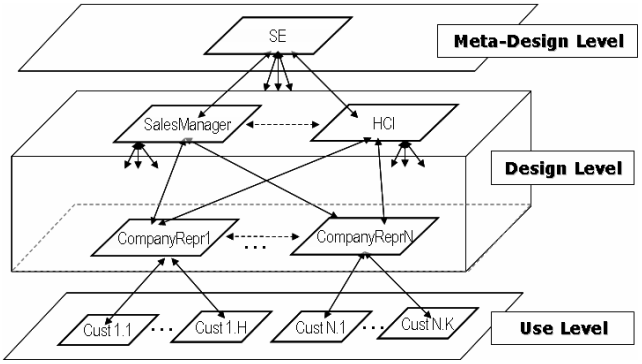


Figure 1. The SSW network in the case of CIDD.

through which he can design the environments to be used by the associated companies, and the companies' representatives will use their environments to tailor the tools to be used by their customers.

During the field study we carried out for requirements analysis, we identified five different types of users based on the roles they have in the consortium:

– *power users*: they are able to see, modify and delete portal contents, define access rules to the portal and even design workshops for the CIDD companies; the role of the power user is played by the sales manager and his secretary, who works on his behalf;
– *associated companies*: their representatives can access contracts, catalogues, promotions, competitions, place orders and design/tailor the workshops for their customers;
– *registered guests*: they are the customers of the associated companies and can see some contents according to the access rules defined by the power user;
– *Unregistered guests*: any user who can see the portal home page when browsing on the web.

The portal allows its users to access and exchange information cooperating through the Web according to the different access rules in force. It is evident that there is a variety of users with different roles and accesses. They are experts in a specific discipline, but not in computer science. They need to use the portal for performing their work tasks, but they are not and do not want to become computer scientists. When the power users, or the representatives of associated companies, modify and update the CIDD portal, they actually program, but they are not aware of this, also because they do not use conventional programming that would be too unfamiliar to their culture and skills, but they compose new software artefacts by construction, similarly to the children's program construction described in [7]. CIDD users work with the available tools through direct manipulation techniques, so that they do not perceive they are creating or modifying software artefacts, they are simply carrying out their work activities. In other words, they are unwitting end-user developers.

The variety of roles and accesses requires different workshops for accessing and managing the different information. The identified SSW network in the case study of CIDD is represented in Figure 1. At the meta-design level (the top level) there is a SSW ("SE" in the figure) used by the software engineers to shape the tools to be

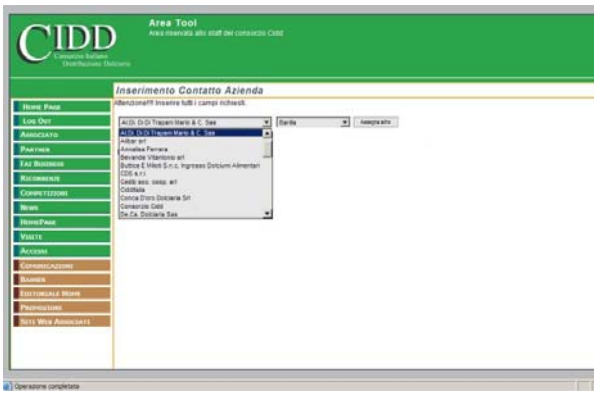


Figure 2. A screen shot of the SalesManager workshop in which is generating services for associated companies

used at the other levels and to participate in the design, implementation, and validation activities of all the SSWs in the network. At the design level, there is a workshop for HCI experts, a workshop for the sales manager and a number of workshops for representatives of associated companies ("AssocRep1", ..., "AssocRepN" in Figure 1). The latter are used by representatives of associate companies to create and modify the application workshops devoted to their customers ("Cust1.1",..., "Cust1.H",.., "CustN.1",..., "CustN.K"). At the use level, there are workshops used by company customers ("Cust1.1".. "Cust1.H", .., "CustN.1",.. "CustN.K") for their consortium activities. On the whole, both meta-design and design levels include all the SSWs that support the design team in their activities of participatory design.

In the *SalesManager* workshop, the sales manager finds tools that allow him to design the system workshops for associate company representatives ("AssocRep1",..., "AssocRepN" in Figure 1). This workshop is shown in Figure 2. Let us suppose that the sales manager wants to design the system workshop to be used by the representatives of an associated company, providing it with some services. He designs this workshop by direct manipulation. Specifically, he selects the company from a drop-down list available in the central area of his workshop (the list is shown in the central area in Figure 2); he also selects a service he wants to provide from another drop-down list (available on the right of the previous one, not open in Figure 2) and clicks on the "Assign" button (the latter on the right in Figure 2) to actually define that service for that company workshop. He does this for all services he wants to provide. The result of this activity is the workshop for representatives of the associated company, presenting the assigned services, that lack of space prevents us to show.

4. CONCLUSIONS

This paper has discussed the actors involved in system development, ranging from pure end users to software professionals. We propose a framework for classifying the different actors, focusing primarily on those that are very active in shaping software tools, without being aware of programming. Such unwitting programmers require new software environments, new tools, new programming approaches able to comply with their needs. The meta-design approach briefly outlined in the paper is able to provide software environments to support unwitting programmers.

5. ACKNOWLEDGMENTS

This work was supported by the Italian MIUR and by EU and Regione Puglia under grant DIPIS. We thank the CIDD consortium and Nicola Claudio Cellamare for their collaboration related to development of the CIDD portal.

6. REFERENCES

[1] Costabile, M. F., Fogli, D., Fresta, G., Mussio, P., and Piccinno, A. 2003. Building environments for end-user development and tailoring. Proc. HCC 2003 (Auckland, New Zealand October 28-31, 2003). 31-38.

[2] Costabile, M.F., Fogli, D., Letondal, C., Mussio, P., and Piccinno, A. 2003. Domain-Expert Users and their Needs of Software Development. Proc. HCII 2003 (Crete, Greece, June 22-27, 2003). Lawrence Erlbaum Associates, 532-536.

[3] Costabile, M.F., Fogli, D., Mussio, P., and Piccinno, A. 2006 End-User Development: the Software Shaping Workshop Approach, in Lieberman, H., Paternò, F., and Wulf, V. (Eds), End User Development. Springer, Dordrecht, The Netherlands, 183-205.

[4] Costabile, M.F., Fogli, D., Mussio, P., and Piccinno, A. 2007. Visual Interactive Systems for End-User Development: a Model-based Design Methodology. IEEE Trans. on SMC - Part A: Systems and Humans 37, 6, 1029 - 1046.

[5] Letondal, C. 2006. Participatory Programming: Developing Programmable Bioinformatics Tools for End-Users, in Lieberman, H., Paternò, F., and Wulf, V. (Eds), End User Development. Springer, Dordrecht, The Netherlands, 207-242.

[6] Lieberman, H., Paternò, F., and Wulf, V. (Eds) 2006. End User Development. Springer, Dordrecht, The Netherlands.

[7] Petre, M. and Blackwell, A. F. 2007. Children as Unwitting End-User Programmers. Proc. VL/HCC 2007 (Coeur d'Alène, USA, Sep. 23-27, 2007). 239-242.

[8] Rosson, M. B., Sinha, H., Bhattacharya, M., and Zhao, D. 2007. Design Planning in End-User Web Development. Proc. VL/HCC 2007 (Coeur d'Alène, USA, Sep. 23-27, 2007). 189-196.

[9] Scaffidi, C., Shaw, M., and Myers, B. (Eds) 2005. Proc. 1st workshop on End-user software engineering. (St. Louis, Missouri, May 21 - 21, 2005), 1-5.

[10] Segal, J. 2007. Some Problems of Professional End User Developers. Proc. VL/HCC 2007 (Coeur d'Alène, Idaho, USA, Sep. 23-27, 2007). 111-118.

[11] Sutcliffe, A., Mehandjiev, M. (Eds.) Special Issue on End-User Development. CACM, 47, 9, 31-66.

[12] Ye, Y. and Fischer, G. 2007. Designing for Participation in Socio-Technical Software Systems. Proc. HCII 2007 (Beijing, China, Jul. 22-27, 2007). LNCS,Springer, 312-321.

VCode and VData: Illustrating a new Framework for Supporting the Video Annotation Workflow

Joey Hagedorn, Joshua Hailpern, Karrie G. Karahalios
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, IL 61801 USA
{hagedorn, jhailpe2, kkarahal}@uiuc.edu

ABSTRACT

Digital tools for annotation of video have the promise to provide immense value to researchers in disciplines ranging from psychology to ethnography to computer science. With traditional methods for annotation being cumbersome, time-consuming, and frustrating, technological solutions are situated to aid in video annotation by increasing reliability, repeatability, and workflow optimizations. Three notable limitations of existing video annotation tools are lack of support for the annotation workflow, poor representation of data on a timeline, and poor interaction techniques with video, data, and annotations. This paper details a set of design requirements intended to enhance video annotation. Our framework is grounded in existing literature, interviews with experienced coders, and ongoing discussions with researchers in multiple disciplines. Our model is demonstrated in a new system called VCode and VData. The benefit of our system is that is directly addresses the workflow and needs of both researchers and video coders.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Evaluation/methodology Video
; H.5.2 [**User Interfaces**]: Graphical user interfaces

General Terms

Design, Human Factors, Measurement

Keywords

Graphical user interfaces (GUI), Annotation, Video

1. INTRODUCTION

Human behavior does not naturally lend itself to being quantifiable. Yet time and again, researchers in disciplines ranging from psychology to ethnography to computer science, are forced to analyze as if it was quantified. Those in human centered domains can now rely on video annotation to provide them with measures on which to draw conclusions. Unlike transcription, which is akin to what a court stenographer does, annotation is the marking of movements, sounds, and other such events (with or without additional metadata such as rankings). The emergence of technology as a tool to aid in video annotation has raised the possibility of increasing reliability, repeatability, and workflow optimizations [6]. Three notable limitations of existing video annotation tools are lack of support for the annotation workflow, poor representation of data on a timeline, and poor interaction techniques with video, data, and annotations. This paper details a set of requirements to guide the design of video annotation tools. Our model is the direct result of an analysis of existing tools, current practices by researchers, and workflow difficulties experienced by real-world video coders. By understanding what data researchers are looking to gather, and the shortcomings of existing techniques and technology utilized by coders, we believe that we have created a framework for video annotation that can reach across disciplines. Our model is demonstrated through the design and construction of our new system VCode and VData (Figure 1); two fully functional, open-source tools which bridge the video annotation workflow.

The primary contribution of this paper is the set of design requirements for facilitating a system conducive to video annotation. Specifically, we demonstrate how a system could be designed and built to meet these requirements through a

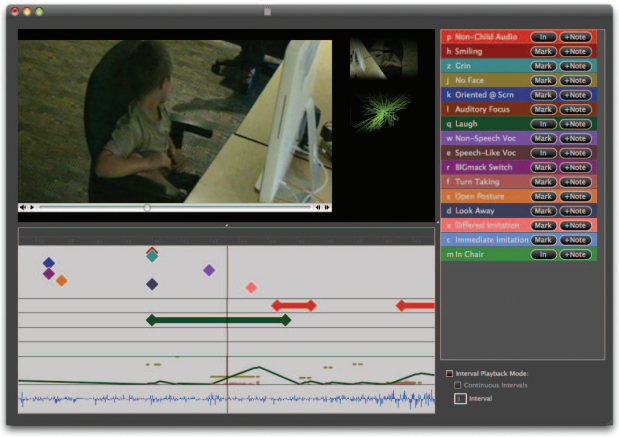


Figure 1: The VCode application graphically represents the behaviors being coded as marks on a timeline. It is easy to see correlation between the marks on the timeline and sensor data displayed below.

AVI '08 May 28-30, 2008, Naples, Italy

set of carefully designed interfaces and graphical representations of data.

2. RELATED WORK

2.1 Video Coding In Practice

The analysis of human behavior is a study that dates back hundreds of years. This has ranged from anthropological ethnographies [10] to psychological evaluations. As technology has developed, the use of video and creation of annotation techniques have aided researchers by providing a referable document that can be used as evidence to back up claims and observations made [21, 16, 20]. These techniques involve detailed event logging on paper, specifying features such as durations, ratings/levels, and time-stamps [15].

To ensure a reliable set of data from annotation, researchers perform agreement calculations between coders [7]. This agreement is utilized throughout the data gathering process (by testing some small percentage of data segments to ensure consistency throughout), but also during training of coders (to decide when they fully understand what events they are looking for). There are many techniques for calculating agreement including Cohens Kappa [11], Cochran's Q-test, and Point-By-Point Agreement. Regardless, the management of data with traditional means is considered "cumbersome" [20].

2.2 Video Coding Tools

Digital annotation tools have demonstrated significant benefits from simple copy/paste and undo to increased quality of coding by the facilitation of multiple passes on video and graphical representations [4, 6]. A timeline is commonly utilized in these tools and is familiar without extensive training [1]. Existing research also indicates that presenting coders with secondary or sensor data on a timeline helps them outperform coders without sensor data [6]. Increased accuracy, quality, and speed not only enhance the data collected, but also allow for more annotation to be conducted in the same amount of time. In addition to the computational benefits of digital annotation tools, they also provide a controllable mechanism for different forms of reliable video playback. [4].

One critical limitation of existing tools is poor representation of data on a timeline and utilization of screen real-estate. For example, the VACA solution, while utilizing minimal screen real-estate by condensing all annotations to one large easy to read track, presents a problem with overlapping and simultaneous events [6]. The VisSTA solution takes the contrary approach by showing many small vertically tiled tracks. Though this allows for a good comparative view, reading individual annotations & holistic interpretation is difficult due to scrolling [4]. These and other existing solutions have not successfully dealt with this problem [14, 3, 13, 12, 1, 2, 22, 18].

Another limitation of current annotation tools is poor interaction techniques with video and data. Though robust functionality is provided for playback, controls can be cumbersome & overly complex, (e.g. [4]). Too many windows resulting in an over-saturation of information, imprecise video interaction & annotation or rigid, inSSexible marking interfaces (e.g. [4, 14, 12, 18, 3]). Each of these are common stumbling blocks which could result in unreliable data.

One last limitation is lack of support for the full annotation workflow; 1) collect video 2) create segments to code 3) train coders/demonstrate reliability 4) gather data 5) perform regular checks on reliability & discuss discrepancies 6) perform data analysis. Many tools support small portions of this workflow (i.e. simply facilitating segmentation, annotation, or reliability [22, 6, 12]), but with each break in the process researchers can become delayed. Without export/import data reentry is required. Technology is situated to optimize this process.

Researchers have also explored dialogue transcription [17, 13, 12], tagging [2, 12], scene based automatic annotation [9, 19], automatic event logging [5], and object of focus identification [8]. Our work contrasts these other foci by demonstrating techniques for supporting human based annotation of events that occur in video.

3. INTERVIEWS & COLLABORATION

To gain a deeper understanding of methods, analysis processes, bottlenecks, and types of data needed for effective video annotation software, we maintained an active dialogue with researchers (in Special Education, Speech and Hearing Sciences, and Computer Science) who use video annotation, conducted informal 40 minute interviews with two experienced video coders, and refined functionality through dialog with current users of VCode and VData. Existing tools for video annotation may address a subset of the below described requirements, however, our system more fully satisfies all of them.

R1 *Facilitate Coding Workflow:* The coding workflow consists of; (1) establishing video clips and coding guidelines, (2) intense training of coders and checks for reliability, (3) annotation of videos, (4) weekly reliability checks on annotated videos, (5) repeat 3 and 4 ad infinitum, (6) analyze data in statistical packages. Tools targeting video annotation should attempt to optimize the transition between steps in this workflow.

R2 *Video, Annotations, and Coding Guidelines should be presented in a synchronized manner:* Interviewees described their coding process centering around analog video on a TV-VCR device, annotating in a Microsoft Excel file, and referencing lengthy code guidelines. Due to the visual separation between annotations, source material, and video, coders had great difficulty during reviews.

R3 *Capture Appropriate Data:* Researchers and existing literature indicate that there are different types of data that are collected through the annotation process: counting events/occurences, determining duration of events, assigning levels, values, or ranking to events, performing phonetic transcription, and general commenting [14]. Effective interfaces must provide methods for capturing these conceptually different data types while preserving each of their unique nuances.

R4 *Additional data should be displayed to coders:* Effective annotation tools should allow researchers to provide additional data to coders to aid in their assessment of video; for example, a volume histogram of the current video, sensor/log data collected in tandem to the video capture, or annotations made automatically or

from another source. Displaying additional datapoints has shown to increase the accuracy of coded events [6]. Further, annotation software should facilitate the management of multiple video streams to get the most accurate "view" on the session, and thus produce the most accurate data [4].

R5 *Allow multiple forms of playback:* Researchers mentioned that continuous playback is not always the preferred method of analyzing a video. Often multiple modes of playback are utilized; continuous or standard playback, continuous interval playback (play for N seconds, then stop), and skip interval playback (jump N seconds, then stop). This allows the video to be divided in to smaller segments for annotation of events that are more difficult to pinpoint (i.e. when a smile starts or ends) [4]. Though conceptually simple, manipulations of video using a standard VCR was described as "annoying" and "a mess" due to hand eye coordination and repeatability issues.

R6 *Agreement calculations should be easy and manipulatable:* Regardless of agreement technique used, researchers expressed a frustration in attempts to calculate inter-observer reliability. Specifically, existing solutions were limited to importing data into a statistical software package for calculation or calculating them by hand. Video annotation tools should provide quick & easy reliability calculations for individual variables, as well as overall.

R7 *Provide functionality for visual, graphical and contextual review of annotations:* In interviews, coders lamented the process of ensuring reliability on a weekly basis; as it consisted of searching through printouts of a spreadsheet for discrepancies. Specifically, by lacking context in this spreadsheet coders found it difficult to recognize what a given coding mark referred to due to the lack of synchronization with video. By providing a visual, graphical way to review annotations (in the context of the video) coders would be better able to justify the decisions, determine the correct solution, and save time identifying the errors.

4. VCODE AND VDATA

VCode and VData are a suite of applications which create a set of effective interfaces for the coding workflow following the above design requirements. Our system has three main components: VCode (annotation), VCode Admin Window (configuration) and VData (examination of data, coder agreement and training). The interaction with VCode and VData is demonstrated in Figures 2-4 in which two coders are marking a video of a child in an experiment, and checking the agreement between their annotations. The reader should note our solution is only one possible implementation of the design requirements, and that these requirements could be applied to improving existing video annotation software.

4.1 VCode

The VCode application (Figure 1) is designed to provide researchers with an effective way to obtain reliable data from an observational research video. By allowing researchers to present multiple video streams in addition to other sensor data (e.g. log data, annotations from other software, or signals recorded by a computer/monitoring device) the coder can make the best annotation decision possible.

Video: To facilitate multiple video streams VCode presents one main video at full size, and a dock with other streams playing in real time. When a docked stream is clicked on, it repositions itself into the main video window, while the video which was the previous focus, scales down to the dock, thus equating visual importance with relative size and visual weight.

Events: When annotating a video, two different classes of coding events emerge: ranged and momentary. A ranged event is one which extends over a period of time (marking action start and duration). Momentary marks have no duration, and thus represent one specific moment in time. Comments can be attached to any mark, allowing additional observations, levels/ranking, or phonetic transcription (through onscreen phonetic keyboard). Any mark with a comment has a inverted outlines to signify that it has a comment attached. Figure 1 shows a ranged event representing the length of time which a child is making a sound, with additional momentary marks at the start noting other features of the child's state of being).

Timeline: The timeline is the heart of VCode. It is modeled after the moving timeline one might find in a video editing application (e.g. iMovie, Final Cut Pro, etc.). Events, graphically represented by diamonds, appear in a spatial linear fashion to sync with the video. Once an event has been placed on the timeline, it can be graphically manipulated by dragging, clicking, and double-clicking. The standard solution for dealing with large numbers of tracks or variables is to provide a vertical scroll bar or overlay tracks. Rather than limiting the amount of information on screen by scrolling, tracks representing momentary events are "stacked," such that they vertically overlap. This optimizes usage of the screen while still providing enough area for track isolation and selection, even under dense data conditions. Ranged event tracks are unable to benefit from this stacking optimization because of the more complicated interaction for manipulation and thus are vertically tiled. Researchers can present video volume, sensor data, software log data (from Eclipse or Photoshop for example), and even other annotations to the coders. This additional information is presented graphically to the users by bar, line, or scatter plot. This secondary data can allow coders to annotate data captured by other sources than the video streams, as well as provide additional context to their code. For example, should a coder be instructed to mark when a certain noise occurs, he can line the mark up with an audio peek, rather than estimate it and be concerned with reaction time.

Interaction: Annotations can be inserted into the timeline via UI buttons or keyboard hot keys. To optimize the typically complex transport controls we isolated the key activities that coders need execute and provided controls limited to play/pause buttons, coarse and fine grained playhead positioning, and step controls. The three modes of playback outlined in R5 are available.

4.2 VCode Administration Window

To ensure consistent configuration between coders and sessions, all administrative features are consolidated in a single window. The expected workflow is such that a researcher would setup a single coding document with all the variables to be used on all the videos. This template would then be

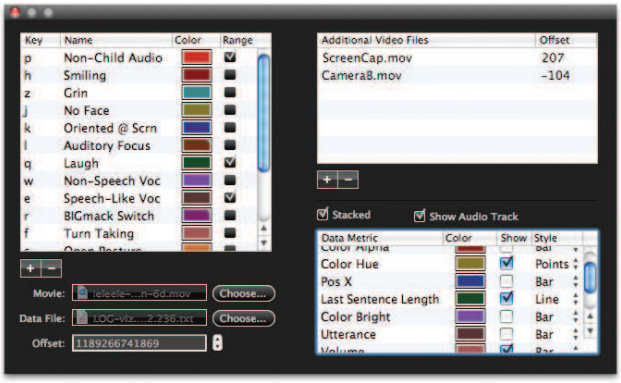


Figure 2: The code is specified in the Administration Window along with the different video angles, screen capture, and log data.

duplicated (with media and log files inserted for each trial). The main task the Administration Window (Figure 2) is to facilitate is the creation of tracks, used to code data. Researchers can add, remove, and reorder tracks which appear in a list format. The name, color and hot key of each tack can be set through this list presentation. Tracks can be enabled as ranged events through a check box in this interface. The Administration Window is also where a researcher specifies videos and data file to be coded, as well as secondary data for contextual annotation. These elements are specified and synchronized through a drag and drop interface, all of which is hidden from the coder to prevent configuration corruption.

4.3 VData

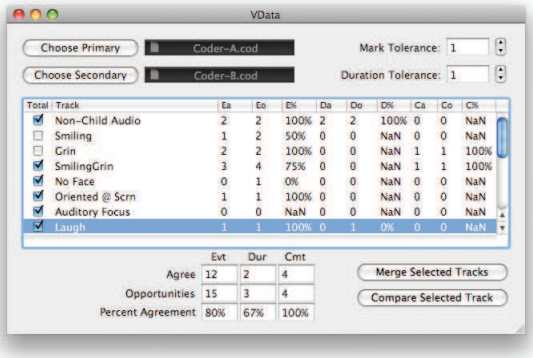


Figure 3: Later, analysis is performed on independent codings of the same video. A track with low agreement can be reconciled by viewing the results of two coders side-by-side in VCode, thanks to the capabilities of the VData analysis tool.

Critical aspects of the video coding workflow (training, reliability, and accuracy) revolve around demonstrating agreement between coders. VData (Figure 3) is a separate executable application specifically targeted to aid researchers in training and agreement analysis of coded data produced in VCode.

Multi Coder Analysis: By loading two VCode files into VData, tracks are automatically loaded into the main data table which presents opportunities, agreements, and percentage agreement. For each event (momentary or ranged) an opportunity is said to occur when the primary coder makes a mark. If the secondary coder also makes a mark within a specified short interval, the marks are said to agree. A percentage is calculated from $\frac{agreements}{opportunities}$ for easy interpretation. A tolerance variable is also present to (1) accommodate for variability in the mark placement by the coders, and (2) recognition that there is no quantization of marks beyond the granularity of the millisecond timescale, a property of the system. VData also provides agreement for ranged events and annotations in a similar fashion. It is not uncommon for multiple tracks or variables to be measuring slight variations on a theme (e.g. smiling vs. large smile vs. grin), thus VData implements a track-merging feature which allows opportunities on two distinct tracks to be treated indistinguishably. For a holistic view, researchers can select tracks to be added into a total agreement calculation. In other words, if analysis determines that a single track is not reliable or it is determined that a given track will not be used in the future, it can be easily excluded from the total agreement calcuation.

Conflict Resolution & Exporting: We have optimized coder training and reliability analysis by providing a graphical mechanism to directly compare annotations of two coders. VData can create a VCode session containing specific tracks of two individual coders for side-by-side comparison. The visual, side by side, representation of the data makes it easy to recognize systematic errors in context and detect differences between two coders markings. This reduces the time necessary to locate discrepancies and discuss the reasons why they might have occurred. It is necessary to keep records of these agreement analyses performed with VData by text export. Maintaining export at each stage of the process provides additional transparency and maintains traceability of results that come out of the system.

4.4 Implementation

Our system was implemented in Objective-C using the Cocoa Framework for Mac OS X 10.5. VCode supports all video formats and codecs supported by QuickTime to enable wide compatibility with available video files.

5. MEETING REQUIREMENTS

To ensure Video, Annotations, and Coding Guidelines are presented in a synchronized manner, VCode provides a unified interface containing the target video, a timeline with graphically represented annotations (ranged event, momentary event, or comment depending on data metaphor), additional tracks of signal data (to increase accuracy), and a list of coding guidelines which place marks and stand as a visual reminder. Three forms of video playback (continous/standard, interval playback, skip interval playback) are available via check boxes on the main VCode window to allow easily switching between modes of playback.

VData provides a dynamic interface for real time calculation of multiple agreement values to facilitate easy and dynamic agreement calculations. Through the transparent calculation process, researchers can see both the raw data, and the percentages side by side for easy judgements about the reliability of data collected. Upon request a visual, graphical and contextual review of annotations for both agreement review and training is supported.

Finally, the Coding Workflow is encouraged through VCode's template model in conjunction with the separate VCode Adminstration Window for easy set up and configuration. Training, data collection, and inter-coder agreement are enabled through a tight collaboration between annotation environment and agreement analysis. By consistently providing data export, researchers can be assured that any information annotated by coders can be easily extracted and exported into the statistical analysis tool of their choice.

6. INITIAL REACTION

To evaluate our system in a cursory fashion, we conducted an informal series of interviews with several coders that used our system during the course of an independent study. Analysis using VData showed inter-observer agreement was good and provided valuable coded data for the study. In general, comments from the coders were positive, especially when comparing the VCode system to non-computerized methods. One coder wrote: "The software was easy to use in general, and cut down on coding time." Several features of VCode stood out in their comments; color coding of tracks provided direct linkage between events on the timeline and the description panel, the correlation between files was clear to see during review, sensor data helped anticipate events and accurately code them. It was also noted that the sensor data provided reassurance that what they had noticed in the video was actually correct.

In addition to these positive marks we uncovered several shortcomings of the interface. The seemingly low-resolution bar-graph of volume data left coders unsure where precisely to make their mark. Because the elements of this graph are relatively wide, it appears especially coarse in comparison with the precision with which one may place a mark on the timeline. A spectrogram was suggested as an alternate visualization of the audio data that could help understand sound and video.

Overall, results from these interviews were very encouraging and suggest a more formal study to determine if performance improves in the same way that coders stated that they felt as the tool lowered the amount of time necessary for coding.

7. CONCLUSION AND FUTURE WORK

Video annotation tools can be valuable to researchers by enhancing the annotation process through increased reliability, repeatability, and workflow optimizations. However, many existing solutions do not fully address all the needs of researchers and coders; effective representation of data on a timeline, efficient and robust interaction techniques with video and data, and support for the full video annotation workflow. Our research has provided many contributions in addressing these weak points.

We create a set of design requirements based on existing literature and annotation techniques, interviews with experienced coders, and discussions with researchers in multiple disciplines. Based on these investigations, we implemented a system, VCode and VData, that largely satisfies the requirements we outlined. These systems were then used in ongoing research, and coders were interviewed concurrent with and after using the software, and their reactions were solicited. Our model demonstrates how video annotation software, for many disciplines, can be enhanced to meet the needs of both researchers and coders.

From the reaction of the coders, as well as our own assessment of VCode and VData, we have many directions of possible future work. One avenue is creating a database or networked system in order to facilitate remote access to content, and management of coding objects and assignments for individual coders. It is foreseeable that the system could be extended to a tool to prepare coding files; assist in dividing up raw footage, syncing data to video enmasse, and other automation hooks. This could leverage some of the other existing work in automatic video segmentation. Lastly, we hope to address some of the concerns of our coders, including creating a richer set of data visualizations.

8. ACKNOWLEDGMENTS

We would like to thank NSF for their support of this work (NSF-0643502).

9. REFERENCES

[1] *Annotations as multiple perspectives of video content*, Juan-les-Pins, France, 2002. ACM.

[2] *The Family Video Archive: an annotation and browsing environment for home movies*, Berkeley, CA, 2003. ACM.

[3] *Fluid interaction techniques for the control and annotation of digital video*, Vancouver, Canada, 2003. ACM.

[4] *A Coding Tool for Multimodal Analysis of Meeting Video*, Montreal, Quebec, Canada, 2004. IEEE.

[5] *Creating Multi-Modal, User-Centric Records of Meetings with the Carnegie Mellon Meeting Recorder Architecture*, Montreal, Quebec, Canada, 2004. IEEE.

[6] *Work-in-progress: VACA: a tool for qualitative video analysis*, Montreal, Quebec, Canada, 2006. ACM.

[7] K. J. Berry and P. W. Mielke. A generalization of cohen's kappa agreement measure to interval measurement and multiple raters, 1988.

[8] M. Bertini, A. Del Bimbo, R. Cucchiara, and A. Prati. Applications ii: Semantic video adaptation based on automatic annotation of sport videos. In *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval MIR*, 2004.

[9] S.-C. Chen, M.-L. Shyu, W. Liao, and C. Zhang. Scene change detection by audio and video clues. `www.cs.fiu.edu/~chens/PDF/ICME02_Video.pdf`.

[10] J. Clifford and G. Marcus, editors. *Writing Culture*. University of California Press, Berkeley, California, 1986.

[11] A. J. Conger. Kappa reliabilities for continuous behaviors and events, 1985.

[12] S. B. Group. Studiocode business group - supplier of studiocode and stream video analysis and distribution software. `http://www.studiocodegroup.com`, 2007.

[13] A. Johnson. About vprism video data analysis software. `http://www.camse.org/andy/VP/vprism.htm`, 2007.

[14] M. Kipp. Anvil - the video annotation research tool. `http://www.anvil-software.de`, 2007.

[15] B. J. Leadholm and J. F. Miller. *Language Sample Analysis: The Wisconsin Guide*. Wisconsin Department of Public Instruction, Madison, Wisconsin, 1995.

[16] L. Lee. *Developmental Sentence Analysis*. Northwestern University Press, Evanston, IL, 1974.

[17] S. S. LLC. Salt software. `http://www.saltsoftware.com`, 2007.

[18] Noldus. The observer. `http://www.noldus.com/site/doc200401012`, 2007.

[19] D. Ponceleon and S. Srinivasan. Automatic discovery of salient segments in imperfect speech transcripts. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 490–497, New York, NY, USA, 2001. ACM.

[20] K. S. Retherford. *Guide to Analysis of Language Transcripts*. Thinking Publications, Eau Claire, Wisconsin, 1993.

[21] K. L. Rosenblum, C. Zeanah, S. McDonough, and M. Muzik. Video-taped coding of working model of the child interviews: a viable and useful alternative to verbatim transcripts?, 2004.

[22] S. Soft. Annotation. `http://www.saysosoft.com/`, 2006.

An Investigation of Dynamic Landmarking Functions

Philip Quinn, Andy Cockburn
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
philip.quinn@canterbury.ac.nz
andy@cosc.canterbury.ac.nz

Indratmo, Carl Gutwin
Department of Computer Science
University of Saskatchewan
Saskatoon, Canada
j.indratmo@usask.ca
gutwin@cs.usask.ca

ABSTRACT

It is easy for users to lose awareness of their location and orientation when navigating large information spaces. Providing landmarks is one common technique that helps users remain oriented, alleviating the mental workload and reducing the number of redundant interactions. But how many landmarks should be displayed? We conducted an empirical evaluation of several relationships between the number of potential landmarked items in the display and the number of landmarks rendered at any one time, with results strongly favouring a logarithmic relationship.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Theory and methods

Keywords

Landmarks, navigation, navigational aids, information scent, visual search, visual clutter.

1. INTRODUCTION

User interface landmarks are visual identifiers or cues that assist the user in locating and orienting themselves in a data space [4]. They can refer to a specific point in the information space, or be representative of a larger group. For example, Google Maps, shown in Figure 1, is a widely used system that makes extensive use of landmarking. It allows access to millions of data items (streets, roads, shops, and so on) and provides simple zooming, overview+detail, and panning tools for navigation.

Although it is understood that landmarks assist in navigation [1, 12], the level of landmarking required to most effectively support navigation is not known—too many landmarks will impede navigation through excessive clutter, but too few will provide insufficient information scent [9].

In this paper, we review the requirements for a landmark function and propose several preliminary relationships. We conducted an experiment to evaluate these relationships in a continuous visualisation of several thousand alphabetically organised items. Results from the evaluation show strong potential for a logarithmic relationship between the number of items presented to the user and the number of landmarks shown.

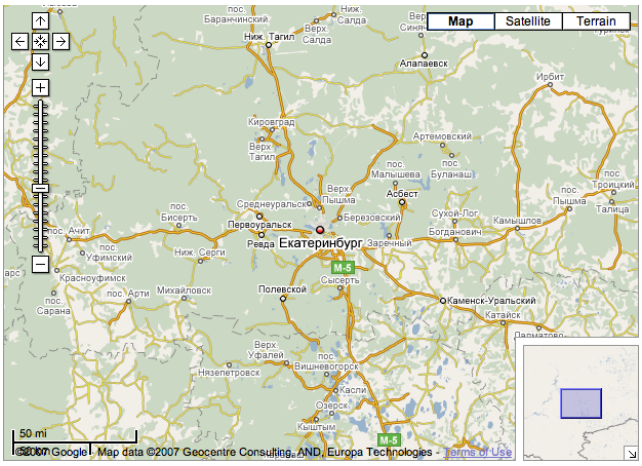


Figure 1: The Google Maps interface; illustrating the geographic visualisation and navigation tools (zoom/pan controls and manipulatable overview inset).

2. BACKGROUND

Several theoretical and empirical studies establish our understanding of information navigation. Pirolli and Card [10] used information foraging theory, including 'scent' (or the cues found at one location to indicate what resides at another) to develop an understanding of the strategies employed by users to seek, gather, and consume information. Furnas [2] used the term "residue" for similar concepts, recommending that information residue should be provided about every data item from every other item—resulting in a completely navigable information space. Woods [12] also introduced a related concept in the theory of *visual momentum*—the ability for users to integrate information across displays of a large information space. He also described *perceptual landmarks* which "providing an easily [discernible] feature which anchors the transition, and which provides a relative frame of reference to establish relationships." [12, p. 236]

In evaluating landmarking systems, Allen et al. [1] found that landmarks were essential for the development of spatial memory. Hornof [5] found that participants could search landmarked screen layouts much more efficiently than those without, but he cautioned against over-proliferation of landmarks, recommending that no more than 30% of a screen should be used for information display (the remaining 70% should be blank). For efficient performance, users

should be able to perform a parallel, pre-attentive scan of landmarks, rather than relying on serial visual hunting [11].

3. EVALUATION

The goal of this research is to explore the comparative effectiveness of functional relationships ($f(n)$) that can be easily implemented in systems to automate the relationship between the number of currently visible items (n) and the number of landmarks presented. The four functions investigated represent a broad coverage of potential behaviours: from relatively abundant landmarks, providing extensive cues at the cost of visual clutter, through to none.

1. **square-root** $f(n) = \sqrt{n}$, chosen due to the high density of landmarks that will be produced;
2. **logarithmic** $f(n) = \log_2 n$, chosen for the slowing rate of increase with large numbers of items;
3. **seven** $f(n) = 7$, based on Miller's [8] research on human cognitive capacity for storing 'chunks' of information.
4. **zero** $f(n) = 0$.

An interesting characteristic of the square-root and logarithmic relationships is that the number of landmarks decreases with the number of items visible. We believe that it is essential for the presence of landmarks to subside once the data items reach a resolution where the user can conduct an efficient visual search over them. At such a point, landmarks would be redundant distractors.

Tasks in the experiment involved locating and selecting a word from a grid of 3,264 alphabetically arranged words (see Figure 2). Landmarks were used to denote the location of word ranges within the grid. Although alphabetically ordered, the grid of words did not start at the letter 'a', intentionally increasing the difficulty of anticipating word location without attending to the semantics of the display (such as viewing landmarks or zooming in to see the underlying words). Words were used rather than other information spaces such as maps to reduce the impact of participants having widely varied knowledge of the information space. For instance, the landmark 'Pakistan' is only of use to a user searching for 'Kabul' if the user knows that Kabul is in Afghanistan, neighbouring Pakistan.

3.1 Apparatus and Participants

The experiment ran on an Intel Pentium D 3 GHz computer running Fedora Core 6, equipped with 1GB of RAM, and an NVIDIA GeForce 6200 connected to a 17″ LCD display at 1280×1024 resolution. All input was through a Microsoft Wheel Mouse Optical, with a 1:2 control-display gain ratio. Python/OpenGL software controlled the participants' exposure to conditions; the software ran full-screen, and logged all actions to microsecond granularity.

The fourteen volunteer participants (two female) were all University students, and their participation lasted twenty minutes.

3.2 Task and Stimuli

The evaluation environment (Figure 2) emulated an interface where participants had to zoom and pan to locate a target. As the level of zoom changed, the landmarks changed to reflect the current set of items shown to the user.

The dataset and targets were all seven letter words, arranged alphabetically (arranged top-to-bottom, left-to-right) in a 4×4 grid of cells (with the exception of training conditions, shown in a 2×2 grid). Each cell consisted of 6 columns of 34 words each; each word was rendered in a 180×32 unit area. Cells were arranged from left-to-right, top-to-bottom in the grid.

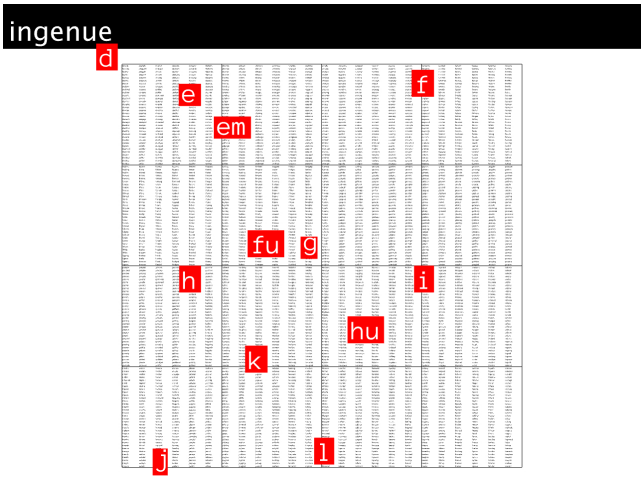


Figure 2: The evaluation interface prompting the user to select the word "ingenue" in the logarithmic condition.

Zooming was controlled by the mouse wheel. Each zoom action was performed around the position of the mouse cursor and altered the current width and height of the viewport by a factor of $\sqrt{2}$—chosen after examining the zooming properties of several other document navigation interfaces. At any zoom level, participants could pan by holding down the left mouse button and dragging. There was a 1:1 mapping ratio between cursor motion and pan distance.

Selection was performed with a single left-click within a word's area, and could be performed at any zoom level.

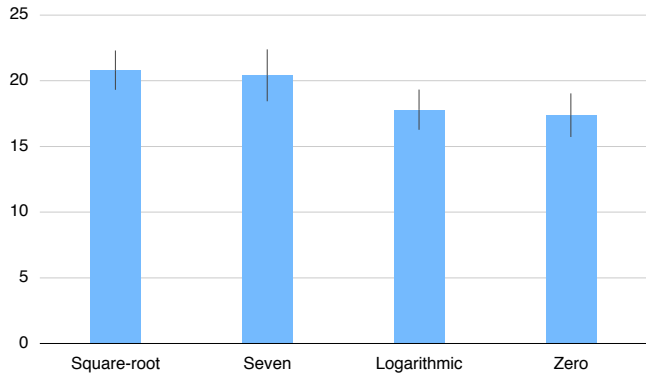
Landmarks were used to denote a range of words, rather than a specific word in the set—for example, words beginning with '*b*', or words beginning with '*bea*'. They were displayed at a fixed size, regardless of zoom; and were always displayed to the left of the first item in the range. Landmarks were chosen based on the content of visible items—at any particular zoom level, a list of possible landmarks was generated (all landmarks for word ranges that began in the list of visible words); landmarks were then selected from the list in the following order:

1. Landmarks that had previously been rendered at the current or more distant zoom levels.
2. The possible landmarks were then filtered by length into groups. Landmarks were then randomly selected from each group (in ascending order); when a group was exhausted, selections continued randomly from the next group.

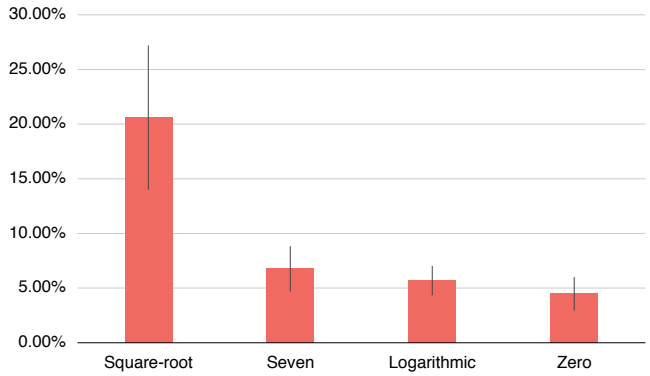
This continued until the maximum number of landmarks prescribed by the current condition had been selected.This algorithm was designed to promote the selection of general landmarks and to support smooth transitions between zoom levels.

3.3 Procedure

Each condition consisted of seventeen random selection tasks and conditions were counter-balanced with an incomplete Latin square. Each selection task began at an initial zoom level such that every item was visible on-screen (as shown in Figure 2). The user then had to zoom/pan until they could locate the target item. Upon correct selection, the zoom was reset to the initial level and the next selection was prompted. An incorrect selection caused the background of the selected word's area to momentarily turn red, but timing continued until the correct selection was made.



(a) Mean selection time (seconds) per condition (+/- SE)



(b) Mean error rate per condition (+/- SE)

Figure 3: Target selection times and error rates.

All tasks were completed with one condition before beginning with the next, with a voluntary rest period between each condition. At the conclusion of the experiment, participants were asked to complete a subjective evaluation form.

3.3.1 Words

Words were selected from a list of 18,553 seven-letter words[1]. The word list was alphabetically divided into four, 4,638 word blocks from which 3,264 words were randomly selected for each condition. This division was to ensure the elimination of learning effects as to the spatial locations of word ranges across conditions. Training conditions had 816 randomly selected five-letter words.

4. RESULTS

The primary dependant variable is task completion time: the elapsed time from revealing the stimulus word until its correct selection. The experiment was run as an analysis of variance (ANOVA) for the within-subjects factor *landmark function* with four levels: zero, seven, logarithmic, and square-root.

A summary of the empirical results are shown in Figure 3. There is a significant main effect for the factor *landmark function* ($F_{3,39} = 9.86$, $p < 0.001$). The zero landmark condition allowed for the fastest mean selection time of 17.37 seconds (sd. 6.21), followed by logarithmic (17.79s, sd. 5.72), constant (20.41s, sd. 7.41), and square-root (20.80s, sd. 5.58). A post-hoc Tukey test gives an Honest Significant Difference (HSD) of 2.47s ($\alpha = 0.05$). This reveals a number of pair-wise significant differences; notably, a significant difference between the logarithmic and 'seven' conditions.

Error rates were highest in the square-root condition with a mean of 20.59% (sd. 24.71%), followed by 'seven' (6.75%, sd. 7.75%), logarithmic (5.67%, sd. 5.01%), and zero (4.48%, sd. 5.67%).

Subjective responses Participants were asked to rank the interfaces in order of preference for the particular landmarking style (ranks 1 to 4, best to worst); interfaces were described to them as "no landmarks" (zero), "few landmarks" (seven), "some landmarks" (logarithmic), and "many landmarks" (square-root)—with a small screenshot representing each one. The logarithmic condition had the best ranking with a median of 1, followed by constant, zero, and square-root (with medians of 2, 3, and 4 respectively). The poor ranking of the square-root condition was backed by negative comments about the clutter and visibility issues caused by the large number of landmarks.

Other comments by participants noted the lack of useful landmarks in the 'seven' condition and the need to perform two visual searches in the square-root condition—one to search for a landmark, and another to search for the target.

Panning and Zooming Actions To further characterise how the landmarking conditions influenced the users' performance, we analysed their panning and zooming actions, finding that participants performed significantly more panning operations in the zero condition than any other condition. The analysis also showed that selections in the square-root condition were performed at a significantly closer zoom level than any other condition. In contrast, participants panned the least in the square-root condition. High standard deviations in the results suggest a wide variance in the navigation activities of different participants.

5. DISCUSSION

The *logarithmic* landmarking function performed significantly better than both the square-root and 'seven' conditions. The logarithmic condition was further supported by the subjective results.

We believe that the strong performance observed for the zero landmark condition was due to a combination of the predictable organisation of the items and the lack of distractors in the condition. This is not an issue for our hypothesis, as the target application of our model are interfaces that necessitate the use of landmarks due to the unpredictable nature of their organisation. The lack of significant difference between the logarithmic and zero landmark conditions show that the use of logarithmic landmarks did not induce additional visual search tasks upon the user.

The navigation data reveals that performance was not dictated solely by the number of landmarks visible, but that it was a contributing factor. In the case of the 'seven' condition, the sparse landmarks were often unhelpful and acted as distractors to the task. In the square-root condition, the abundance of landmarks induced their own visual search tasks that needed to be completed before searching for the target item.

We believe that the logarithmic condition was able to find the appropriate level of landmarking to allow for useful landmarks (further supported by the comments from several participants)—preventing them from being distractors to the task. However, further study is required to assess the factors that caused the 'seven' condition to perform significantly worse than the logarithmic condition.

5.1 Experimental Concerns

All experiments raise concerns of generalisability and validity, described below to aid further work and replication.

Organisation of Items Items were organised alphabetically in a grid, but other configurations could have been used. The grid

[1]Retrieved from `http://www.math.toronto.edu/~jjchew/scrabble/lists/common-7.html`

configuration was chosen because it does not place a visual search bias in any one direction and gives the smallest number of zooming steps and shortest average path between any pair of items.

However, this configuration caused word ranges to wrap across rows of the grid, reducing the value of the landmark. This caused frustration for participants, as they had to zoom and pan over to the following row and begin their visual search task again.

Predictable Data The use of alphabetic data allowed participants to predict the locations of items in the absence of landmarks to guide them—we believe this was the reason for the performance observed in the zero landmarks condition. Techniques—such as randomising the order of grid cells—would have reduced this effect, but at the detriment of the landmark's descriptive power; landmarks would no longer indicate the start of a contiguous group of data, but only a partial set.

Obstruction Participants complained that landmarks sometimes obscured the underlying words. Landmarks were always of a fixed visual size, and long landmarks often obscured words at distant zoom levels. Continuing to zoom in would have eliminated this effect, but participants were often unwilling to zoom more than necessary in order to be able to read the items. Techniques such as using transparent landmarks, or landmarks that reacted to the cursor position may have reduced this frustration and error rate.

6. FUTURE WORK

In our evaluation, landmarks were chosen at random to avoid bias, but in production interfaces, this is not a viable option. Selection of landmarks that gave preference to the semantic importance of the data that each landmark is representative of, or the frequency/recency of items used within the group are possible strategies that would result in more valuable landmarking. We believe that the selection of such a strategy is dependant on the data being shown (although generic techniques such as cluster analysis [7] have been investigated), but we are interested in future evaluation and comparison of such strategies.

Development of a landmarking model is still in its infancy and further evaluation of its application is required in order to discover the contributing factors to user efficiency with landmarks and establish its strength and validity. Evaluation in scenarios where landmark location would not carry such high predictive power as that in our evaluation should be conducted, as should the influence of different landmarking strategies.

We believe there may also an interesting interaction between landmarking and spatial memory principles. As landmarks are promoted as spatially stable identifiers and can be rapidly searched, there is reason to believe they may improve development of spatial memory of the data being navigated.

We also believe there may be a connection to the Hick-Hyman Law [3, 6], which models the time taken for a user to make a decision amongst several choices as a function of the item's probability. Due to the low information content of landmarks (probability varies with the landmark selection strategy), landmarks would appear to have a short reaction time—supported by the Hick-Hyman Law and our evaluation results. Future work may investigate the link and possible model for human performance between our developing model for landmark visibility and the Hick-Hyman Law.

Irrespective of the Hick-Hyman Law, we are interested in future work that further establishes an accurate model that can be applied by designers of navigation interfaces for landmarking.

7. CONCLUSION

The evaluation of our landmarking relationships has strong implications for interfaces that use landmarks to aid user navigation. Interfaces that necessitate the use of landmarking by virtue of the nature of their data are the primary benefactors. For example, in mapping interfaces—such as Google Maps—the relationships between sibling items are less predictable than those in sets of alphabetic words. In such interfaces, the ability to identify these relationships is dependant on the user's knowledge of the data (although navigation tools, and features of navigation views can assist with this), and landmarks are required to provide rapid orientation. Absent landmarks require a comprehensive visual search of low-level items, and too many landmarks induce further serial hunting tasks due to the visual clutter.

The primary goal of landmarks are to provide navigation aids at a density that assists the user without becoming distractors and inducing additional visual search tasks. The empirical and subjective results from our evaluation show that a logarithmic relationship principle has best supported this goal.

Acknowledgements

This research was funded by a New Zealand Marsden grant.

8. REFERENCES

[1] G. L. Allen, A. W. Siegel, and R. R. Rosinski. The role of perceptual context in structuring spatial knowledge. *J. Experimental Psychology: Human Learning & Memory*, 4(6):617–630, November 1978.

[2] G. W. Furnas. Effective view navigation. In *CHI '97: Proceedings of ACM conference on Human factors in computing systems*, pages 367–374, New York, NY, 1997. ACM Press.

[3] W. E. Hick. On the rate of gain of information. *Quarterly J. Experimental Psychology*, 4:11–26, 1952.

[4] J. Hochberg and L. Gellman. The effect of landmark features on mental rotation times. *Memory and Cognition*, 5(1):23–26, 1977.

[5] A. J. Hornof. Visual search and mouse-pointing in labeled versus unlabeled two-dimensional visual hierarchies. *ACM Trans. Comput.-Hum. Interact.*, 8(3):171–197, 2001.

[6] R. Hyman. Stimulus information as a determinant of reaction time. *J. Experimental Psychology*, 45(3):188–196, 1953.

[7] S. Jul and G. W. Furnas. Critical zones in desert fog: aids to multiscale navigation. In *UIST '98: Proceedings of ACM symposium on User interface software and technology*, pages 97–106, New York, NY, 1998. ACM Press.

[8] G. A. Miller. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review*, 63:81–97, 1956.

[9] P. Pirolli. Computational models of information scent-following in a very large browsable text collection. In *CHI '97: Proceedings ACM conference on Human factors in computing systems*, pages 3–10, New York, NY, 1997. ACM Press.

[10] P. Pirolli and S. Card. Information foraging. *Psychological Review*, 106(4):643–675, October 1999.

[11] A. Treisman. Perceptual grouping and attention in visual search for features and for objects. *Journal of Experimental Psychology: Human Perception and Performance*, 8(2):194–214, April 1982.

[12] D. D. Woods. Visual momentum: a concept to improve the cognitive coupling of person and computer. *International Journal of Man-Machine Studies*, 21(3):229–244, 1984.

Eulr: a novel resource tagging facility integrated with Flickr

Rosario De Chiara
ISISLab - Dip. di Informatica
ed Applicazioni "R.M. Capocelli"
Università di Salerno - ITALY
dechiara@dia.unisa.it

Andrew Fish*
School of Computing, Mathematical
and Information Sciences,
University of Brighton - UK
Andrew.Fish@brighton.ac.uk

Salvatore Ruocco
ISISLab - Dip. di Informatica
ed Applicazioni "R.M. Capocelli"
Università di Salerno - ITALY
salvruo@gmail.com

ABSTRACT

We have developed a novel information storage and display structure called EulerView, which can be used for the systematic management of tagged resources. The storage model is a non hierarchical classification structure, based on Euler diagrams, which can be especially useful if overlapping categories are commonplace. Keeping the constraints on the display structure relaxed enables its use as a categorisation structure which provides the user with flexibility and facilitates quick user tagging of multiple resources. As one instance of the application of this theory, in the case when the resources are photos, we have developed the Eulr tool which we have integrated with Flickr. User feedback indicates that the Eulr representation is intuitive and that users would be keen to use Eulr again.

Keywords

Euler Diagrams, Tagging, Classification, Categorisation, Flickr, Metadata, EulerView.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Graphical user interfaces (GUI); H.5.3 [**Group and Organization Interfaces**]: Web-based interaction

1. INTRODUCTION AND BACKGROUND

Metadata refers to data about data, and typical usage involves data about resources such as documents, books, articles or photos. Metadata are specific to the particular purpose they are designed for (e.g. metadata that can be associated to books). They are used to facilitate the retrieval of those specific resources. The importance of Metadata is increasing, especially in web-related activities, because many web sites allow users to add metadata to resources such as photo, URLs and blog entries. In this particular context the operation of adding metadata has been dubbed *tagging*.

Tags are an effective way for authors to categorize their resources: they facilitate future retrieval of information using tags as keywords. Similarly, tags are useful for readers who are browsing for resources. Although an author may be the best person to assign tags to his/her resources, his/her resources may mean something different to other people, or they may just perceive the tags used in a different manner. For this reason web sites like Del.icio.us allow users to tag other people's resources. This *collaborative tagging* has been dubbed "Folksonomy" [6, 12], joining the term *folk* and the term *taxonomy*; "specifically it refers to subject indexing systems created within Internet communities".

Flickr tagging system.

Flickr is a popular photo-sharing site that exploits tags as a core element to the sharing, retrieval, navigation, and discovery of user-contributed images. Every Flickr user can upload his/her photos to be stored online and they can choose to allow these photos to be publicly viewable and therefore easily discoverable. Making photos accessible to the public, together with strong emphasis on the tagging facility [11] has facilitated the expansion of the site; currently it boasts more than $700,000$ registered users. A functionality of Flickr which is greatly appreciated is the support of social interactions: in addition to uploading photos, users can create networks of friends, join groups, send messages to other users, comment on photos, tag photos, choose their favourite photos, and so on.

Flickr offers support for the development of applications which are aimed at manipulating photos, tags, contacts and the whole set of resources available. It offers a well documented set of libraries available for many programming languages. This library has enabled the flourishment of a wide range of applications that implement different kinds of elaboration of Flickr resources. We are interested here in those applications that have a strong focus on tag management. The standard Flickr application [2] and other similar, but independently developed applications [4] allow users to select photos and associate tags to them. Tag management is an important issue because many social interactions within the Flickr community rely on tags. This is emphasized by the existence of many applications which enable the user to browse through photo collections using tags as a guideline [1, 3, 5].

We wish to facilitate systematic tag-management, which raises the users awareness of the relationships between categories, and so we review and discuss categorisation and classification structures.

Categorisation and Classification.

We recall the following distinction between categorization and classification [10]: "Categorization divides the world of experience into groups or categories whose members share some perceptible similarity within a given context", where category composition depends on the context and on the user of the organization; "classification involves the orderly and systematic assignment of each entity to one and only one class within a system of mutually ex-

*funded by UK EPSRC grant EP/E011160: Visualisation with Euler Diagrams.

AVI'08, May 28-30, 2008, Napoli, Italy

clusive and non-overlapping classes", where classification usually refers to the assignment of a resource (e.g. a document, URL or photo) to a single class, among classes, often hierarchically organized making clear the complete relationships amongst classes (see [7] for a typical biological example).

We wish to delve a little deeper into the characterisations of categorisation and classification. The process of categorisation is generally perceived as being less precise than classification: the placement of an item within a classification structure indicates precise global information about that item, whereas placement of an item within a categorisation structure may represent partial information about the item, which is to be interpreted locally, or within a given context.

We also wish to also broaden the notion of classification structures: since hierarchical classification structures are often not sufficient for user-classification needs (and this is even felt to be the case in a real office setting [14]), one can consider non-hierarchical classification structures, such as polyarchies [13] or EulerView [8]. One could stretch the imagination slightly and think of non hierarchical classifications as fitting into the classification definition given above if one thought of two overlapping classes as three classes: two non-overlapping classes together with another class for the intersection. On the other hand it also fits quite well with the notion of categorisation: items in the intersection of categories share some property. So, non-hierarchical classifications could be thought as living somewhere in the middle of this Categorisation-Classification spectrum, allowing the overlap of categories but with a formal underlying model.

In general, strict classification system interfaces are especially useful for storage and retrieval of information, but are often thought to be too restrictive for users, for many reasons, such as: they may be time consuming to use, they may get very large and the visualisations unmanageable very quickly, and changes in the classification structure over time may require the update of all existing resources that are already classified – which may be a difficult chore. On the other hand, categorisation systems such as free-form tagging are much easier and quicker to use, but they have their own drawbacks such as: the necessity to use different visualisation techniques, such as tag clouds [9] in order to browse through existing resources.

We advocate a "best of both worlds approach", by keeping an underlying (non-hierarchical) classification system, but presenting to the user interactive views of that data in the form of a simple categorisation structure which may be user-created and easily manipulated. This will bring the benefits of a classification structure, especially in terms of information retrieval, but with the flexibility associated to a categorisation structure.

In [8], EulerView was proposed as a means to enable the common user to easily capture the naturally non hierarchical organization of website bookmarking. User difficulties due to migration from the standard Treeview control were avoided by developing EulerView as an extension of Treeview. However, the hierarchy of a Treeview control is extended into a non hierarchical structure by using the power of the Euler diagram model; developing the EulerView component as such a $(1 + 1)$-dimensional visualization of Euler diagrams also assists the user in overcoming potential navigation difficulties (i.e. scrolling and/or zooming in 2 or 3 dimensional interfaces).

2. EULR APPLICATION

We wish to enhance the user experience of tagging resources by providing them with the facility to create and easily manipulate their own categorisation structure to update tags (and in particular tags for photos on Flickr). So we wish to consider common users' tasks, and what facilities would be useful to assist them in performing those tasks, whilst not adding too much computational overhead to the system.

Tagging tasks can be described as the process of adding, or altering, tags to resources, where this might involve adding multiple tags to multiple resources. Whilst free-form user tagging allows quick single resource tagging it doesn't allow multiple resource tagging, although there are applications that do facilitate this feature. However, they do not give an idea of the related categorisation structure, which would aid user awareness and facilitate quick tagging of multiple related resources.

We consider certain features that would be desirable with reference to some natural scenarios. Firstly, we note that the familiar use of a tree-structure where the parent-child vertex relationship represents category-subcategory relationship is a natural method for displaying a categorisation structure in a non-flat manner.

A feature desirable in categorisation structures is the facility of overlapping categories (called intersection categories). For example, a user may wish to place a photo in their "University Friends" Group but also in the overlapping category of "Work Colleagues". This is useful within a single natural categorisation of groups of people, but if the user wants to tag a collection of photos which are "University Friends, Work Colleagues, In Italy and At Conference" then they may wish to create the intersection across naturally different categorisation structures. This motivates our desire to display a non-hierarchical categorisation structure (and to use an underlying non-hierarchical classification structure).

Enabling the use of multiple user categorisations provides flexibility. For example, suppose that a user wants to tag a collection of photos taken on holiday abroad with a certain group of friends. Suppose that the user had two categorisation (or classification) systems, one with groups of friends according to different social circles (so these groups may overlap) and another according to places in the world they they have visited. Then, placement of the photos within these categorisation structures, enabling partial tagging (i.e. not requiring the explicit creation of intersection categories) would provide a quick method for utilising both categorisation structures together. This would also facilitate re-use of user-created categorisation structures. Therefore, a small investment in initial creation of categorisation structures will lead to continued benefits in the future. This motivates our desire to develop a user categorisation display structure which has a high degree of flexibility.

We have developed a visual component called EulerView which has an underlying model based on Euler diagrams. It has a novel, flexible interface to enable users to construct multiple integrated categorisation paths, which will facilitate tagging tasks as well as be useful in enhancing ease of future navigation (or classification) tasks. This methodology will extend to multiple different users using their own (multiple) categorisation (or classification) systems, but still enabling them to share their information effectively. To ground our ideas we have developed an application which is an instance of this EulerView theory. This integration of EulerView with Flickr has been dubbed *Eulr*; it facilitates user creation and management of tags attached to photos in Flickr.

The Eulr tool facilitates the development of a user categorisation structure, making it easier to assign sets of tags to collections of resources. The constraints we impose are very relaxed, allowing great flexibility, although the imposition of stronger constraints could be enforced, if desired. Eulr also allows the export, to Flickr, of the user-defined sets of tags for a collection of resources. Furthermore, we enable the import of multiple resources from Flickr into the Eulr display, thereby creating a categorisation structure which the user can use or manipulate. Thus we have addressed our major tagging

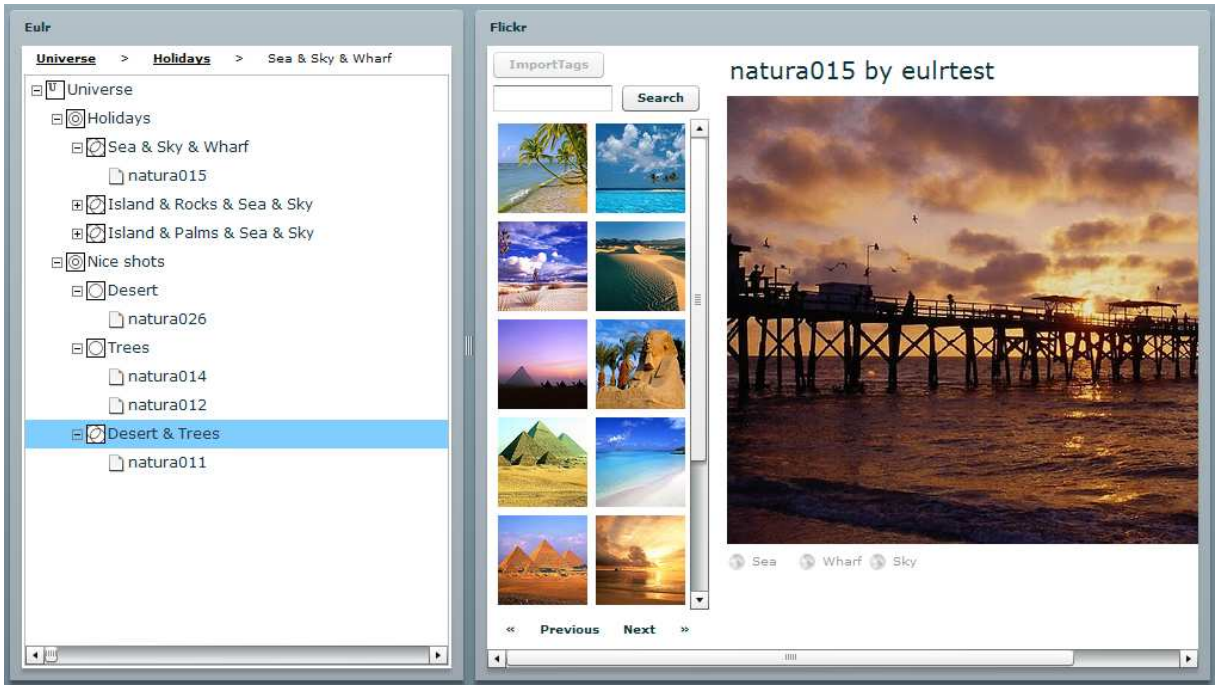


Figure 1: Eulr application user interface: on the left pane the EulerView control, on the right pane the Flickr photos.

requirements.

A major advance of Eulr over the existing Flickr applications is that Eulr enables the systematic management of tags. This brings many advantages such as enhancing the users' understanding of the categorisation structure (allowing user construction and manipulation of such a structure which displays relationships between resources); this can be a difficult comprehension task using "flat" tags. It has also been developed within a constraint-based framework, where the level of constraint could be varied, according the the application domain or user preferences, for instance. Tag management is an important issue because many social interactions within the Flickr community rely on tags.

2.1 Eulr user interface

Figure 1 shows the current user interface. The left pane shows the EulerView component used to categorize photos, while the right pane displays photos taken from a specified Flickr account. The Flickr pane presents the whole photo collection in thumbnails divided into pages in order to allow the user to browse through them rapidly. It also displays a larger photo of the currently selected photo, together with the associated tags for the photo that are stored on Flickr. Clicking on a thumbnail changes the currently selected photo and updates the display accordingly. A search facility is also available which enables the user to search for photos with a specified set of tags.

The EulerView control in the lefthand pane shows a user constructed categorisation structure. The basic idea is that we have a rooted tree for the categorisation structure, where the set of tags associated to a vertex are in fact the complete set of tags in the path back to the root node (called the Universe). Resources, which are in this case links to the photos, can be placed as shown thereby associating the corresponding set of tags to the resource. For example, the photo "natura012" has associated tags {Nice shots, Trees} within the EulerView control, whilst the photo "natura015" has associated tags {Holidays, Sea, Sky, Wharf}.

Notice that the tags associated to the photo in the Flickr pane for "natura015" are {Sea, Sky, Wharf}, which are not the same as those in the EulerView pane. This is because we chose not to enforce the automatic update of tags in Flickr whilst the user was manipulating the EulerView control, but instead to allow the user to decide when they wish to export the set of tags associated to each photo to Flickr. Once exported the updated set of tags become visible in the Flickr pane. For instance, upon exporting from the EulerView pane in Figure 1, the tag "Holidays" would also be added to the tags for the photo "natura015", and the other photos would have their tags updated. Automatic updates could instead be used if user-preference dictated it.

The icon associated to a category vertex consistently reflects the content of the branch from that vertex in the Eulr structure, as shown in Figure 2.

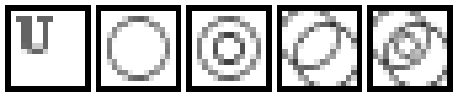


Figure 2: Icons for different categories (left to right): the Universe of Discourse category, a simple category, a category containing subcategories, an intersection category, an intersection category containing subcategories.

Now, the core functionality of the Eulr system is to be able to manage photos and the tags associated with them. Firstly, the user needs to create an EulerView categorisation structure, and then use that to help in photo tagging: drag a photo from the Flickr pane and drop it on the EulerView pane within a category is the basic means to associate the corresponding tags to that photo. Clicking on a resource in Eulr opens the link to that resource.

The Eulr categorization structure can be created and manipulated by a user, and we have very few constraints enforced in order to facilitate user flexibility. The universe, or root vertex, is always

present. A new category vertex can be added as child vertex of any existing category vertex. Any non-root category vertex v can be moved to become a subcategory of another vertex w; note that the branch from v, moves with it. Two category nodes, with distinct label sets A and B, can be intersected to create a new category vertex with label $A\&B$. One can relabel any single-label category vertex. When an intersection is created, the label of the intersection vertex is linked to the single-label category vertices involved in its label and consistently reflects the renaming of those labels. Any non-root category vertex v can be deleted and the branch from v is also deleted (including all of the resource links). Resources (which are links to photos stored in Flickr here) can be added, deleted and moved at will (so one can place their own photos and change their labels). Adding new categories or resources, together with renaming, is achieved by right click followed by selection via a drop down menu. Category movement is achieved via drag and drop, and category intersection is achieved using drag and drop with a modifier of holding down the control key.

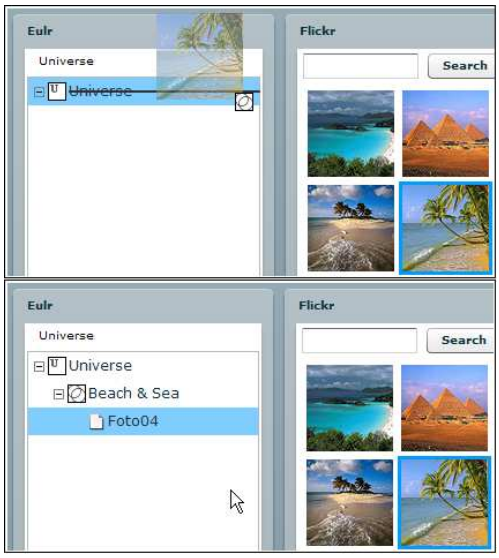


Figure 3: Drag and drop of a photo using the wizard.

In order to assist the user in creating their categorisation structure we developed a *wizard* function that allows a user to "migrate" the set of tags from photos to EulerView automatically generating their associated EulerView categorisation structure. Figure 3 shows the effect of category creation using the wizard. The top picture in Figure 3 shows the photo "Foto04" tagged with $\{Beach, Sea\}$ dragged, keeping the Control key pressed, onto the EulerView. During the drag operation a thumbnail of the photo tracks the mouse pointer and a black line helps the user to see where he/she is about to drop the photo. The bottom picture in Figure 3 shows the effect of dropping the photo: an intersection named $Beach\&Sea$ is created containing the photo. This wizard can be applied to a single photo, or a set of photos, and it also allows them to be imported into the Eulr display either just as resources or together with the automatic creation of the relevant categories. The selection of multiple photos from Flickr is achieved by holding down the control key. Releasing the control key allows drag and drop to a category vertex v to place the set of resources in v, whilst keeping the control key depressed enables the drag and drop to a category vertex v to place the automatically generated Eulr structure created as a subcategory of v (which could be the root node). This facilitates easy categorisation of photos with similar tags.

Relationship of EulerView with Euler diagrams.

We give an example to give an idea of the relationship between EulerView and the Euler diagram model. The top part of Figure 4 shows a modified Euler diagram which is an alternative representation of the EulerView shown in Figure 1. This is a standard hierarchical view (without using the overlap facility), where the labelled curves represent a set of tags. For example, the category vertices "Desert, Trees and Desert & Trees" in Figure 1 are all subcategories of "Nice shots" and as such they are presented as 3 separate curves contained within the "Nice shots" curve. The bottom diagram is the more traditional Euler diagram view of this model, using overlap to represent intersection of categories.

3. USER PERCEPTION OF EULR

As a first step in exploring the usefulness of the Euler idea and the application, we wished to investigate user perception; if users had a particularly negative perception of the ideas and the application, then investing further resources in developmental time would have seemed inappropriate. There is much further user testing to be performed in the future, including comparisons of Eulr with the utility and perception of other tagging systems, and the effect of using Eulr on larger scale systems, but here we are primarily concerned with small scale everyday usage of Eulr.

User perception of the system was explored by guiding users through Eulr and its functionlities, having them follow a guided set of 11 tasks and then presenting them with a questionnaire. Finally, users were asked to give their own feedback on their perception of the Eulr concept and its integration into the system.

The test was undertaken by 16 users (14 male), who were computer science students (average age 25.8). The tasks were devised to help clarify the basic concepts involved (e.g. what is an intersection and how is one created) as well as to help the users get acquainted with these functionalities. The first 10 tasks involved the users managing 4 photos, creating an EulerView display, and placing photos in the relevant categories. The last task was much freer, asking the users to categorize about 20 photos by using or modifying the EulerView structure that they had developed earlier.

The questionnaire consisted of 15 questions, with responses measured on a Likert scale of -3 to 3, and it involved questions such as: "Understanding the system at first glance is: hard/easy" and "The software does what you expect: never/always". The overall consensus of the impression of the system was extremely positive (averaging 2.4) and they were keen to use the software again (average of 2.6). Learning to use the system was deemed easy (average 2.3), and it appears that a small initial effort helps the user understand the system (average score of 2.1 for ease at first look, raising to 2.6 for ease of use after some usage). Users believed that it does not require long to become acquainted with the application (average of 2.2), which may have been assisted since they perceived that the exploration of the functionalities of Eulr were quite straightforward (average of 2.0). Potential areas for improvement were the look of the graphical interface (average 1.6), the quality of the graphics (average 1.2) and the use of colour (average 1.8). The users perceived that Eulr speeds up their tagging work (average 2.2).

Users also provided feedback, in their own terms, on the concept of Eulr, and this indicated that the Eulr representation is intuitive and that the assimilation of concepts into a governing framework is excellent. Whilst these results are, by their nature, the subjective opinions of the participants they provide encouragement that the concept is worth further development.

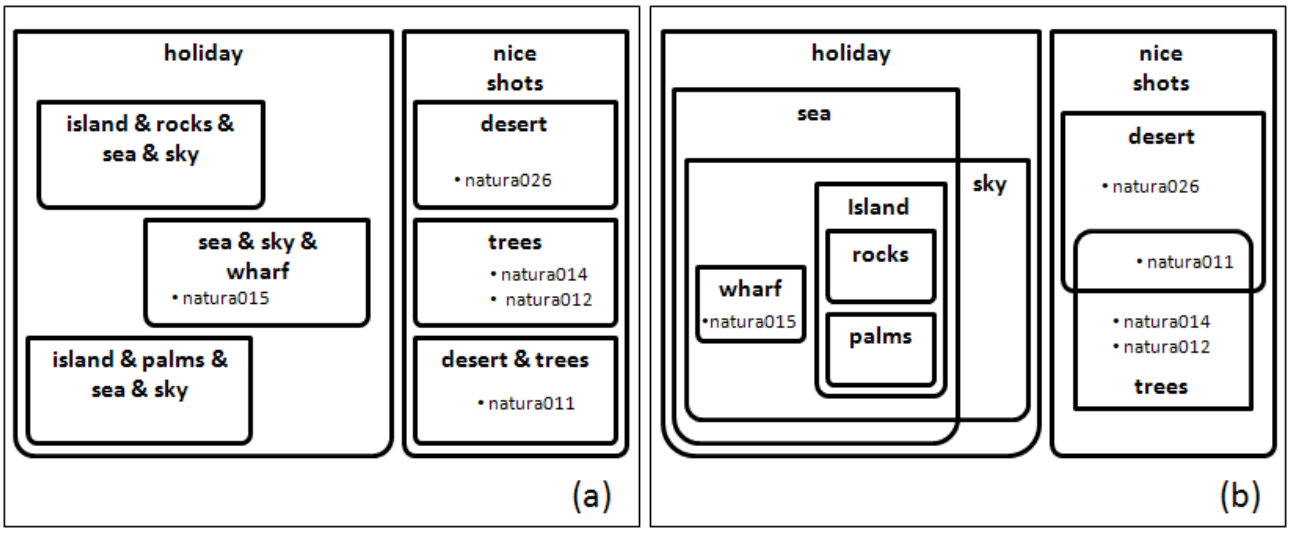


Figure 4: (a) A modified Euler diagram matching the EulerView in Figure 1 (b) a traditional Euler diagram of the same model

4. CONCLUSION

We have developed a novel information storage and display structure called EulerView, which can be used for the systematic management of tags. It has the flexibility of a relaxed categorisation structure (the EulerView display), but also provides an underlying non-hierarchical storage model which is based on Euler diagrams. To ground this theory in practice, we developed the Eulr tool, which is an instantiation of the EulerView model where the resources to be tagged are photos which are stored on Flickr. Functionality has been included to enable users to build multiple non-hierarchical categorisations within which to place links to photos from Flickr. The resource tagging information can be exported to update the tag set of the photos on Flickr. Furthermore, imports of multiple resources from Flickr, with or without an associated Eulr categorisation structure is enabled. User guided tests provided feedback which indicates that users would be keen to use Eulr again.

Our focus here was on user resource-tagging, but we envisage that EulerView will be of great benefit in the future as a single interface to be used for both user tagging and searching tasks; there are many avenues to investigate further related to alternative display options, the level of constraints imposed, and browsing and searching facilities. In the future, more sophisticated techniques to aid visual focussing, as in [15], may be incorporated within the EulerView. As with any tagging system, questions about the utility for large scale systems is an important avenue for future investigation, but the Eulr application here has been developed explicitly for simple small scale user usage.

5. REFERENCES

[1] Findr. http://www.forestandthetrees.com/findr/.

[2] Flickr: Organize your photos. http://flickr.com/photos/organize.

[3] Related Tag Browser. http://www.airtightinteractive.com.

[4] Smark. http://smark.us/.

[5] Tagnautica. http://www.quasimondo.com/tagnautica.php.

[6] Wikipedia: Folksonomy. http://en.wikipedia.org/wiki/Folksonomy.

[7] Wikipedia: Taxonomy. http://en.wikipedia.org/wiki/Linnaean_taxonomy.

[8] Rosario De Chiara and Andrew Fish. Eulerview: a non-hierarchical visualization component. In *VLHCC '07: Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*, pages 145–152, Washington, DC, USA, 2007. IEEE Computer Society.

[9] Douglas Coupland. *Microserfs*. Harper Collins, 1995.

[10] Elin K. Jacob. Classification and categorization: a difference that makes a difference. *Library Trends*, 52(3):515–540, 2004.

[11] Cameron Marlow, Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM.

[12] Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata. Technical report, 2004. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html.

[13] George Robertson, Kim Cameron, Mary Czerwinski, and Daniel Robbins. Polyarchy visualization: visualizing multiple intersecting hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 423–430. ACM Press, 2002.

[14] Thomas W. Malone. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst.*, 1(1):99–112, 1983.

[15] K. Wittenburg and E. Sigman. Visual focussing and transition techniques in a treeviewer for web inforamtion access. In *Visual Languages*, pages 20–27. IEEE Computer Society Press, September 1997.

Ambiguity Detection in Multimodal Systems

Maria Chiara Caschera
Istituto di Ricerche sulla Popolazione e Politiche Sociali, CNR, Via Nizza 128, 00198 Roma
mc.caschera@irpps.cnr.it

Fernando Ferri
Istituto di Ricerche sulla Popolazione e Politiche Sociali, CNR, Via Nizza 128, 00198 Roma
fernando.ferri@irpps.cnr.it

Patrizia Grifoni
Istituto di Ricerche sulla Popolazione e Politiche Sociali, CNR, Via Nizza 128, 00198 Roma
patrizia.grifoni@irpps.cnr.it

ABSTRACT
Multimodal systems support users to communicate in a natural way according to their needs. However, the naturalness of the interaction implies that it is hard to find one and only one interpretation of the user's input. Consequently the necessity to define methods for users' input interpretation and ambiguity detection is arising. This paper proposes a theoretical approach based on a Constraint Multiset Grammar combined with Linear Logic, for representing and detecting ambiguities, and in particular semantic ambiguities, produced by the user's input. It considers user's input as a set of primitives defined as terminal elements of the grammar, composing multimodal sentences. The Linear Logic is used to define rules that allow detecting ambiguities connected to the semantics of the user's input. In particular, the paper presents the main features of the user's input and connections between the elements belonging to a multimodal sentence, and it enables to detect ambiguities that can arise during their interpretation process.

Keywords
Multimodal interfaces, interpretation of multimodal input, multimodal ambiguity, grammar-based language.

1. INTRODUCTION
Multimodal systems provide a natural interaction between users and systems according to the users' needs. However, naturalness implies ambiguities as an intrinsic phenomenon. Indeed, identification of one and only one meaning of multimodal input is a crucial aspect in order to provide a flexible and powerful dialog between the user and the system. The concept of ambiguity is closely connected to the semantic gap between the user's intentions and how she/he is able to convey it, since this fact can lead to more than one interpretation of the user's input. Methods and strategies for interpreting the user's input are closely connected with formal methods for representing multimodal input. In particular, these methods support composite multimodal input aligning them and expressing their relations and their combined semantic representations. After discussing some of these methods, this work proposes a new approach based on the combination of a Constraint Multiset Grammar [2] with the Linear Logic [1] for representing the multimodal input and for detecting and classifying ambiguities connected with their semantics. The literature proposes two different approaches for interpreting the multimodal inputs: a) the first one interprets the input for each modality and these interpretations are combined by a dialog manager, b) the second uses a grammar-based approach. A grammar-based approach is more natural than a dialog based one, since the dialogue is seen as a unique and multimodal communication act, and it allows an easier integrated management of ambiguities. A multi-dimensional grammars is well suited to cover the formal definition of multimodal language syntax, because multimodality is multidimensional in its nature; the approach was hybridized using Linear Logic to provide a robust mechanism to manage ambiguities that are, in general, a relevant problem. The linear logic enables to integrate the different input and to interpret them introducing the concept of action, event and resource (important to model non-linear feature of the multimodal interaction). The proposed approach aims to improve the quality of the interaction with the system allowing the identification of ambiguities connected to the semantics of the user's input. The paper is organized as follows: section 2 presents some works of the literature on multimodal input integration and interpretation, section 3 describes an hybrid approach for detecting ambiguities connected to the semantics of the multimodal input and how it can be used for detecting semantic and lexical ambiguities. Finally section 4 concludes the paper.

AVI'08, 28-30 May , 2008, Napoli, Italy

2. RELATED WORK
In literature several methods for integrating inputs belonging to different modalities of interaction, such as sketch, speech, gesture, gaze and handwriting, have been defined. In particular, speech and gesture inputs can be combined and interpreted using finite-state mechanisms [3]. The finite-state approach is extended in [4], and it is integrated by a declarative multimodal grammar that captures the structure and the interpretation of unimodal and multimodal commands. This grammar consists of a set of grammar rules and terminals. Each one of them is defined by words and gesture symbols. In [4] the interpretation process is defined using an edit-based transducer combined with a finite-state-based interpreter, which directly works on lattice inputs. The "referent resolution approach", used in [5], is a further approach for multimodal interpretation. It finds the most proper referents, such as specific object or objects that are the models to which the users' input has to be matched, to the referring expressions given by the user's input. In this approach, for optimizing reference resolution based on graph matching [6], the probabilistic approach developed in [7] is used. In systems where speech is the prevalent modality, the interpretation process of the user's input uses parse trees [8] with semantic grammars [9]. In detail, the semantic

information is dependent on the syntactic structure of the sentence, because the parse tree is converted into a set of typed feature structure [10], where each feature structure represents objects in the domain and relationships between objects by the use of a set of expert-system style rules. These previous approaches are mainly introduced in order to define the meaning of the multimodal input and they, consequently, are correlated to the management of ambiguities and their semantics. These ambiguities are detected when the previous methods lead to two or more than two different interpretations of the multimodal input. Furthermore, in literature, ambiguities connected to the syntactic structure of the multimodal input are also dealt. This kind of ambiguities appears when alternative structures of the multimodal input can be generated during the interpretation process. However, for brevity, the focus of our paper is the detection of the multimodal ambiguities connected to the semantics, so we postpone the management of the syntactic ambiguities to future works. Ambiguities connected to the semantics of the input are widely discussed in the literature. In particular, a deep analysis about ambiguities has been carried out for natural language. It has produced a widely used classification of this kind of ambiguities [11] in: lexical ambiguity, semantic ambiguity and, pragmatic ambiguity. Lexical ambiguity arises when an object has more than one interpretation; semantic ambiguity arises when different ways of combining the meaning of elements in a sentence produce several interpretations; finally pragmatic ambiguity occurs when a sentence has several interpretations in the context in which it is expressed. While the meaning of semantic ambiguities is independent from the context, the meaning of pragmatic ambiguities is context-dependent. In particular, a pragmatic ambiguity appears when is unclear which element a pronoun or deictic refers to. This classification will be used in the following in order to show how the approach proposed in this paper allows detecting classes of ambiguities. In particular, we will discuss the hybrid approach, which combines Constraint Multiset Grammar with Linear Logic, underlining how it allows providing the interpretation of the multimodal input in an adaptive way. Indeed, the hybrid approach allows defining a language that is able to adapt itself to the interaction thanks to the structure of the Constraint Multiset Grammar, which is attribute-based, and the notion of resource introduced by the Linear Logic. Consequently, the hybrid approach provides an adaptive treatment of the ambiguities.

3. MULTIMODAL AMBIGUITY DETECTION

This section will discuss the hybrid approach we are proposing and how it can be used in order to detect semantic and lexical ambiguities.

3.1 Hybrid approach

The naturalness of interaction in multimodal systems implies that it is hard to find one input that has one and only one interpretation, and that a sentence, separated from the context, is not ambiguous. Clearly, to improve the quality of the user's input interpretation by the system, it is necessary to define methods that can identify potential ambiguities. The main purpose of this paper is to provide a theoretical approach for detecting ambiguities connected with the meaning of the user's input using a formal structure for the multimodal input. First of all we introduce the notion of *multimodal sentence;* it is the unit that, using a *syntactic structure*, an *interpretation function* and a *description*, combines the elements *E* containing information referred to each *concept* given by the different *modalities*, with its *representation* and *time interval*. Here we consider the main features for classifying ambiguities that can arise, and analyse the structure of multimodal sentences and connections between elements that compose them considering the attribute-based structure of the grammar and the concept of recourses provided by the Linear Logic. We have hypothesized to model multimodal interaction combining a Constraint Multiset Grammar with Linear Logic for dealing multimodal sentences and to detect multimodal ambiguities that appear during the interpretation process. This grammar is an advanced Attribute-based Grammar [12], which allows to compute derived attributes of non-terminal symbols using computation embedded into the grammar productions. While the Linear Logic extends the Classical Logic and it introduces the notion of resource and the concept of formulas as resource. Our approach analyses the multimodal input starting from the natural language. Indeed, multimodal input is matched on a sentence in natural language. It considers user's input as composed of a set of terminal elements in the Constraint Multiset Grammar combined with rules expressed by Linear Logic. The terminal elements of the grammar are the building units of the Multimodal Language of our system [13]. So the multimodal input is structured as a parse tree, where each leaf of the tree is a terminal element of the Constraint Multiset Grammar, and it includes information about the specific element. In detail, each non-terminal element of the grammar is decomposed using: 1) the production rules of the grammar; 2) the context rules that are defined through the domain knowledge and the discourse context knowledge; and 3) temporal rules that impose how combining terminal elements according to the fact that their temporal intervals are temporally closed or not. In this structure, if two leaves have the same parent so they have to define redundant element, while, leaves that have different parents are complementary [14]. These terminal elements contain information connected to: i) the input modality related with each element; ii) the representation of the element in the specific modality; iii) the temporal intervals connected to the element; iv) the semantic of the element considering its representation according to the modality; v) the syntactic role that the element plays in the multimodal sentence. Each terminal element E^i of the proposed approach is defined by a 5-pla (E^i_{mod}, E^i_{repr}, E^i_{time}, $E^i_{concept}$, E^i_{role}) [15] with:

- $E^i_{mod:}$ that defines the modality used to create the element E^i,
- E^i_{repr}: that defines the representation of the element E^i in the specific modality,
- E^i_{time}: that defines the temporal interval connected to the element E^i,
- $E^i_{concept}$: that specifies the concept name of the element E^i considering the representation of the element according to the modality,
- E^i_{role}: the syntactic role that the element E^i plays in the multimodal sentence.

For example, let us suppose that the system allows the user to interact using sketch and speech modalities and she/he says the word "*school"*. In this case the element that is defined is the following:

E^i *is* ! (E^i_{mod}=*speech*) ⊗ ! (E^i_{repr} = “ *school*”)) ⊗ !(E^i_{time}=(17,20)) ⊗ ! ($E^i_{concept}$=(*school*)) ⊗ !(E^i_{role}=(*n*))

Now it is possible to describe how the introduced formal method can detect ambiguities connected with the meaning of the multimodal input. The following sections will show the main features for detecting semantic and lexical in multimodal systems. Pragmatic ambiguities, which are connected to the relations between elements and the context, will not deal for brevity.

3.2 Semantic ambiguity detection

An example of semantic ambiguity is the target ambiguity, which appears when the focus of the user is not clear and univocally identifiable. This kind of ambiguity arises when in the same sentence two or more than two possible candidates can be the targets of the user and so they can share the same role in the structure of the sentence. Let us suppose that the user interacts with the mobile system that supports speech and sketch interaction, and she/he needs information about houses in a specific place. The user's input by speech is *“show this near school”*, while the user's input by sketch is given by a red line overlapping a house and a second red line overlapping a shop as shown in the following figure.

Figure 1: Second sketch input of the user

Therefore, the following sentence is produced considering the combination of speech and sketch input (Figure 2):

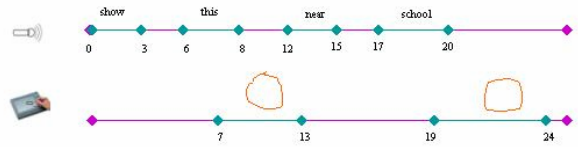


Figure 2: Multimodal input

Considering temporal relationships between speech end sketch modalities inputs, the system aligns the first sketch with the word “*this*” and the second sketch with the word “*school”*. This alignment detects a target ambiguity because the second sketch input identifies a “*shop”* aligned with a “*school”* identified by the speech input, and so these two elements are not coherent at the semantic level. In particular the multimodal sentence includes elements defined by the speech modality that are:

- E^1 *is* ! (E^1_{mod}=*speech*) ⊗ ! (E^1_{repr}= “*show* ⊗ ! (E^1_{time}=(0,3)) ⊗ ! ($E^1_{concept}$=(*verb*)) ⊗ !(E^1_{role}=(*v*))
- E^2 *is* ! (E^3_{mod}=*speech*) ⊗ ! (E^2_{repr} = “*this* ⊗ ! (E^2_{time}=(6,8)) ⊗ ! ($E^2_{concept}$=(*deictic*)) ⊗ !(E^2_{role}=(*deictic*))
- E^3 *is* ! (E^3_{mod}=*speech*) ⊗ ! (E^3_{repr} = “*nea* ⊗ ! (E^3_{time}=(12,15)) ⊗ ! ($E^3_{concept}$=(*adverb*)) ⊗ !(E^3_{role}=(*prep*))
- E^4 *is* ! (E^4_{mod}=*speech*) ⊗ ! (E^4_{repr} = “ *sch*) ⊗ ! (E^4_{time}=(17,20)) ⊗ ! ($E^4_{concept}$=(*school*)) ⊗ !(E^4_{role}=(*n*))

And using the sketch modality the elements of the multimodal sentence are:

- E^7 *is* ! (E^7_{mod}=*sketch*) ⊗ ! (E^7_{repr} =)) ⊗ ! (E^7_{time}=(7,13)) ⊗ ! ($E^7_{concept}$=(*house*)) ⊗ !(E^7_{role}=(*n*))
- E^8 *is* ! (E^8_{mod}=*sketch*) ⊗ ! (E^8_{repr} =)) ⊗ ! (E^8_{time}=(19,24)) ⊗ ! ($E^8_{concept}$=(*shop*)) ⊗ !(E^8_{role}=(*n*))

These elements of the sentence are combined using the following production rules of the natural language:

pp → *prep, np;*	*np* → *n, pp;*	*np* →*deictic;*	*vp* → *v, np;*
s → *np, vp;*	*np* → *art, n;*	*np* → *n;*	*vp* → *v, np, pp;*
np → *art, n, pp;*	*np* → *adj, n;*		

Considering these production rules the parse tree of the previous sentence is defined in Figure 3.

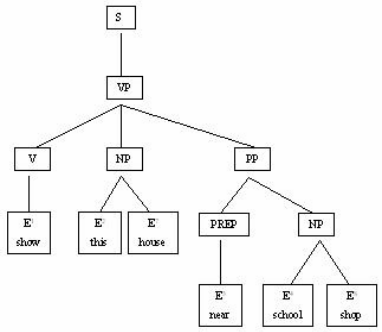


Figure 3: Parse tree of the sentence

In this example the combination of the element E^2 with the element E^7 does not produce semantic ambiguities because the semantic concept connected with E^2 ($E^2_{concept}$=(deictic)) is not incoherent with the semantic concept connected with E^7 ($E^7_{concept}$=(house)). On the contrary the alignment of the element E^4 with the element E^8 defines a target ambiguity because they are leaves of the same node (NP) -which identifies a role-, but the semantic concept connected with E^4 ($E^4_{concept}$=(school)) contrasts with the semantic concept connected with E^8 ($E^8_{concept}$=(shop)), in fact:

$$(E^4_{concept} \neq E^8_{concept}) \otimes (E^4_{role} \equiv E^8_{role})$$

The ambiguity is due to the fact that they refer to two different concepts defining a target ambiguity, while E^4 and E^8 should contain redundant information in order to avoid the target ambiguity because they share the same parent.

3.3 Lexical ambiguity detection

The lexical ambiguity is connected to the semantics of the elements of the language, and it appears when the meaning of an element is not clearly identified. Let us suppose that the user interacts with the system by sketch and speech in this case too.

Figure 4: Sketch input of the user

Using the speech modality the user says: *“show this in Rome”*, while the user simultaneously draws the sketch (Figure 4). Considering this drawing and the set of elements of the

domain, the Figure 4 can be interpreted both as a river and a street. So the meaning of the user's input is not clearly identified. In this case the elements identified by the speech modality in the sentence are four, while the sketch modalities identifies an element that has two different meanings (river and street) and the parse tree connected with the previous cited elements is presented in Figure 5. In this example it is important to underline that the drawing defined by the sketch input can be referred to two different concepts:

- E^5 *is* ! (E^5_{mod}=*sketch*) ⊗ ! (E^5_{repr} = ≈) ⊗ ! (E^5_{time}=(6,9)) ⊗ ! ($E^5_{concept}$=(*river*)) ⊗ !(E^5_{role}=(*n*))
- $E^{5'}$ *is* ! ($E^{5'}_{mod}$=*sketch*) ⊗ ! ($E^{5'}_{repr}$ = ≈) ⊗ ! ($E^{5'}_{time}$=(6,9)) ⊗ ! ($E^{5'}_{concept}$=(*street*)) ⊗ !($E^{5'}_{role}$=(*n*))

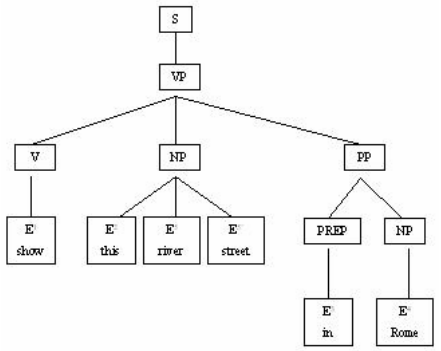


Figure 5: Parse of the user's input

In this case, the alignment of the element E^2 (that defines the element 🔊 "*this*") with the sketched element detects a lexical ambiguity due to the fact that the element defined by sketch can have two different meanings, river (E^5) and street ($E^{5'}$), in the context. So the rule that allows to identify this ambiguity is the following:

$(E^5_{concept} \neq E^{5'}_{concept}) \otimes (E^5_{repr} \equiv E^{5'}_{repr}) \otimes (E^5_{mod} \equiv E^{5'}_{mod}) \otimes (E^5_{role} \equiv E^{5'}_{role})$.

4. CONCLUSIONS

In this paper we have shown how the combination of Constraint Multiset Grammar and Linear Logic can be used to detect ambiguity connected with the semantics of the multimodal input. This approach allows to consider the main features for classifying ambiguities and defining the connection between the structure of the multimodal ambiguous input and the specific class of ambiguity. These features are connected to the modality used to create the elements of the multimodal sentence; the representation connected with the modality of the elements; the temporal intervals connected to the elements; the semantics referring to the elements; and, finally, the syntactic role of the elements in the multimodal sentence. Each one of the elements defined by the user's input are analysed, and rules expressed by Linear Logic allow to detect if an ambiguity connected to the semantics appears. As future work, we will investigate rules for detecting other classes of ambiguities connected with the semantics and ambiguity connected to the syntax of the sentence. We plan to classify syntactic ambiguities and propose rules for detecting this kind of ambiguities.

5. REFERENCES

[1] J.-Y. Girard. 1987. Linear logic. *Theoretical Computer Science, 50*. pp.1-102.

[2] Sitt Sen Chok, K. Marriott. 1995. "Automatic construction of user interfaces from constraint multiset grammars," *vl, 11th International IEEE Symposium on Visual Languages.* p.242-245.

[3] Johnston, M. and S. Bangalore. 2005. Finite-state Multimodal Integration and Understanding. *Journal of Natural Language Engineering 11.2, Cambridge University Press.* pp. 159-187.

[4] Johnston, M. and S. Bangalore. 2005. "Combining Stochastic and Grammar-based Language Processing with Finite-state Edit Machines". *In Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop.*

[5] Joyce Yue Chai, Zahar Prasov, Shaolin Qu. 2006. Cognitive Principles in Robust Multimodal Interpretation. J. Artif. Intell. Res. (JAIR) 27: pp.55-83.

[6] Tsai, W. H., & Fu, K. S. .1979. Error-correcting isomorphism of attributed relational graphs for pattern analysis. IEEE Trans. Sys., Man and Cyb., 9, pp.757–768.

[7] Chai, J. Y., Hong, P., & Zhou, M. X. 2004. A probabilistic approach to reference resolution in multimodal user interfaces. In Proceedings of 9th International Conference on Intelligent User Interfaces (IUI), pp. 70–77.

[8] M. Collins. 1997. Three generative, lexicalised models for statistical parsing. *In Proceedings of the 35th Meeting of the Association for Computational Linguistics and the 7th Conference of the European Chapter of the ACL*. pp. 16-23.

[9] Marsal Gavalda and A. Waibel. 1998. Growing Semantic Grammars. *Proceedings of ACL/ Coling 1998, Montreal, Canada*.

[10] Carpenter, B. 1992. The Logic of Typed Feature Structures. *Cambridge University Press.*

[11] D. M. Berry, E. Kamsties, M. M. Krieger. 2003. From contract drafting to software specification: Linguistic sources of ambiguity. A Handbook. *University of Waterloo, Waterloo, Ontario, Canada, 2003.*

[12] K. Marriott, B. Meyer, and K. Wittenburg. A survey of visual language specification and recognition. *In K. Marriott and B. Meyer, editors, Visual Language Theory, Springer, New York, 1998.* pages 5–85.

[13] D'Ulizia A, Ferri F., Grifoni P. 2007. A Hybrid Grammar-Based Approach to Multimodal Languages Specification, *OTM 2007 Workshop Proceedings, 25-30 November 2007, Vilamoura, Portugal, Springer-Verlag, LNCS 4805*. pp 367-376.

[14] Martin J.C. (1997). Toward Intelligent Cooperation Between Modalities: The Example of a System Enabling Multimodal Interaction with a Map. *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'97) Workshop on "Intelligent Multimodal Systems." Nagoya, Japan.*

[15] Caschera M.C., Ferri F., Grifoni P.. 2007. An Approach for Managing Ambiguities in Multimodal Interaction. *OTM 2007 Ws, Part I, LNCS 4805. Springer-Verlag Berlin Heidelberg 2007.* pp. 387–397.

Fostering Conversation after the Museum Visit: a WOZ Study for a Shared Interface.

Cesare Rocchi, Daniel Tomasini, Oliviero Stock, Massimo Zancanaro
FBK-Irst
Via Sommarive 18
Povo, Trento - Italy
+39 0461 - 314578
{rocchi,tomasini,stock,zancana}@fbk.eu

ABSTRACT
According to recent studies, a museum visit by a small group (e. g. a family or a few friends) can be considered successful if conversation about the experience develops among its members. Often people stop at the museum café to have a break during the visit or before leaving. The museum café is the location that we foresee as ideal to introduce a tabletop interface meant to foster the conversation of the visitors.

We describe a Wizard of Oz study of a system that illustrates the reactions of people to visual stimuli (floating words, images, text snippets) projected on a tabletop interface. The stimuli, dynamically selected taking into account the topic discussed and a set of communicative strategies, are meant to support the conversation about the exhibition and the visit or to foster a topic change, in case the group is discussing something unrelated to the visit. The results of the Wizard of Oz show that people recognized visuals on the table as "cues" for a group conversation about the visit, and interesting insights about the design have emerged.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Evaluation/Methodology, Interaction styles, prototyping, screen design.*

General Terms
Design, Experimentation.

Keywords
Conversation, museum visit, tabletop interface.

1. INTRODUCTION
There have been many research efforts to introduce technological tools to support people during a museum visit: information kiosks, personalized mobile guides, etc. The research trends have mainly focused on personal support for the visitor: e.g. locate a room or provide information about details of interest for the individual. Recently, a new perspective for enhancing the visit has emerged: the small group dimension. In general people tend to visit a cultural site with a family or a group of friends. Petrelli and Not report that only 5% of the visitors come to the museum alone while 45% come in organized groups, 20% with friends and 30% with children [7]. New possibilities, introduced by the social dimension, can be offered by technology: support communication between users during the visit, seamless interaction with personal devices and public displays [8].

We propose a novel aspect: technological tools that provide support after the visit, when visitors can have a conversation about the experience they lived. In particular we investigate a tabletop application placed in the museum café specifically designed to foster conversation about the visit. The table acts as a 'mediator' for the group by means of images, words, text snippets, which are used in a visually dynamic way, floating on the table, to evoke a discussion about the visit. The importance of after visit conversation to enhance the cultural experience has been investigated in [4]; discussing the subject is a key factor that implies personal involvement and learning

A museum visit is a multifaceted experience, which involves cognitive, emotional and social dimensions. A small group scenario involves specifically a plot of interpersonal relationship, e.g. friends or colleagues. The social dimension helps the development of a perspective shift, where learning is less considered as 'owning' concepts and more close to an 'opportunity' to actively interact [4]. This is often referred to as *constructivist* approach, where knowledge is 'created' rather than 'given'. In such a perspective there are opportunities to interact with visits companions and other 'mediators' like labels and booklets. Adopting this viewpoint conversation, and in general 'opportunities to actively interact', are important during the visit but also afterwards. The museum café is a good setting to share thoughts and impressions about the visit.

The technology we envisage acts as a facilitator for human-human natural interaction rather than providing functionalities to access further information. The scenario is set after the museum visit, or after a first part of a visit. The system supports three phases:

a) a phase in which it tries to promotes conversation about the museum visit experience;

b) a phase that supports conversation by providing contents appropriate for the specific topic being discussed by a group and the state of the conversation;

AVI'08, May 28-30, 2008, Napoli, Italy

c) a phase where users explicitly seek further information about some cultural heritage topics by interacting with the system.

The first two phases do not require explicit user interaction. Yet, the behavior of the interface is influenced by: the profiles of the group members (including their social relations and data related to the visit) and their behavior and conversation at the table. The third phase the system acts as a kiosk, allowing the group to browse information related to the museum.

In phases a) and b), the interface uses visual tools (floating words and pictures) as 'mediators' to foster or support a conversation about a specific topic. These visual tools are meant to create a 'space' for interpretation, which can lead the group to fixate ideas, share impressions, exchange opinions and - in general – get along with the spirit of the visit, promoting an interpretative engagement.

The novel aspects of this type of systems makes difficult to base the design on previous similar experiences. At this stage of the project we do not seek for completeness; we rather try to foster initial design ideas. In this paper we describe a Wizard of Oz experiment aimed at investigating the effect of communication strategies that may be employed in phases a) and b). The results of the study show that people recognized visuals on the table as "cues" for a group conversation about the visit. Subjects sometimes feel the need to interact with the display not only for seeking new information but, for example, to zoom images or to have a more detailed view of a specific visual. In general the role of the table has been recognized to provide 'hints' and to support the conversation though, sometimes, the interface was considered too crowded or some visual effect (e.g. flashing) was upsetting.

2. RELATED WORK

Mankoff and colleagues have investigated the role of ambient displays in different scenarios and have proposed a set of heuristics to evaluate their efficacy [5]. One of the first pioneering works on ambient display of media is Tangible Bits [3], which explored techniques to enable background awareness. GroupCast is an office application that senses people passing by. It has a profile of users and displays mutual interest to people [6]. The goal is to create informal interaction opportunities between people. The information displays shared interests of contrasting attitudes.

Drift table is an eccentric experiment of interactivity that displays an aerial photo of England through a hole in a table [1]. The image pans according to the weights on the table. It has been designed to support ludic activities, to stimulate curiosity, exploration and reflection. A qualitative observation highlighted that people got engaged by experimenting weights and narrating about the places spotted.

Hello Wall is a digital wall made of a grid of lights [9]. Depending on the distance of people the wall changes communicative function (ambient, notification, interaction). Abstract light patterns convey information about mood, presence and crowdedness.

3. COMMUNICATION STRATEGIES

Communication strategies tested during the WOZ are inspired by the world of advertisement, where the form of the presentation and the layout play an important role. The message conveyed by an advertisement has to be clear. General guidelines are: a clean layout helps locating what is crucial (usually at the center) and what is peripheral (close to borders); a good contrast of color helps to read and figure out the difference between objects and a non-crowded scene helps focus better on fewer objects.

Alternative approaches make use of semiotics theory and rhetoric. Slogans and catchphrases, for example, are conceived to be easily memorable. Rhymes, verses and words that carry musicality are simple to understand and remember. Rhetorical figures are meant to catch attention by exploiting a deviation from the standard meaning of words. For example *metaphor* exploits analogy to explain an unknown thing through the comparison with another familiar thing. Our graphical interface exploits spatial patterns of elements floating over the surface. Such patterns have been devised to exploit graphical metaphors, used by the system to display the state of the conversation, the relations between the topics discussed and to evoke discussions about the visit.

We have implemented and tested with users the following strategies:

Orbiting. *An image rotates on itself in the center of the table while another image, smaller, rotates around it.* This pattern is meant to convey the idea that there is a semantic relation, e.g. they represent a similar scene in two different frescoes. The orbiting image can also work as a potential topic for the next discussion. This strategy exploits the metaphor that physical proximity between objects indicates semantic closeness. The semiotic code behind this strategy is to recall the same pattern that occurs between a planet and its satellite.

Closeness. A similar strategy, which exploits patterns in the positioning objects, is built on the idea of *objects, which rotate around a point*. This spatial pattern is meant to communicate a group relationship. The idea behind this strategy is related to the sharing of a feature, in our case a point, between a set of objects.

Attention grabbing. Since our system is dynamic, in that stimuli can change according to the conversation, we also consider Weber's law as a relevant idea to account for. Weber's law describes the relation between the extent of a stimuli and the perceived intensity of it. To induce the perception of a change, the trigger stimulus has to be proportional to the current intensity of the stimuli perceived. For example, if a display is empty (e.g. black background) the introduction of a graphical object is easily perceivable. In contrast, on a display crowded of floating words the addition of a new word, especially if similar in size and color, is more difficult to recognize. The implementation of *flashing and pulsing is meant to shift the attention of the subjects from the current stimulus to the one highlighted*.

Not related content. To create curiosity and grab attention a non-related object can be used. In our implementation we introduced *a word that was not related to the frescoes of the visit*. This is a technique borrowed from advertisement, which exploited the contrast of an object in an unrelated scenario.

Dimension. *To highlight the relevance of an object to the ongoing discussion the object is enlarged, to have a greater size with respect to the others shown on the table.* The semiotic code behind this metaphor is meant to convey the idea that more relevant stimuli are more visible and emerge from the rest.

4. WOZ EXPERIMENT

A Wizard of Oz experiment has been performed to study the reaction of the users to an active table in the museum café. We focused on two phases of the tabletop application: foster a conversation about the visit and provide support when it is already occurring. We hypothesize that data available to the system are: images and words about the exhibition. The image repository includes overall representations of exhibits and some closer view of relevant details. Words have been extracted from exhibits' labels. Moreover, an automatic speech system is able to detect the occurrence of a number of words and allows the system to understand the topic of the conversation.

Three groups of four people were invited to visit a reconstruction of a painted room at the FBK-Irst. People were asked to visit the exhibition as if they were a group of friends. We did not provide any indication about the way to conduct the visit, nor we set up a time limit. Subjects were given a four-page booklet to help them during the visit. Each group took from ten to twelve minutes to visit the four frescoes.

Figure 1. A picture of the setting.

After the visit people were conducted to another room and they were invited to sit at a table while waiting for the experimenter to come back. Some coffee was offered (see Figure 1). A beamer had been installed on the ceiling to display the visual stimuli directly on the table. A camera positioned on the ceiling was used to observe the subjects and a panoramic microphone was installed to listen to the conversation. The wizard, located in another room, was therefore able to monitor the group behavior and control the presentation of the visual stimuli projected onto the table. During the conversation the wizard tested the strategies presented above in different situations: e.g. group talking about the visit, about a single exhibit and about topics unrelated to the exhibition

4.1 Analysis of the Observations

The start state of the system was a black background with no information displayed. Reactions of people when the first visual effect appeared were meaningful. Regardless if the stimulus was a word or a picture, people noticed the presence of an object almost immediately and they talked each other about that.

They almost always recognized that the visual effects were somehow related to the topic discussed. For few seconds some group shifted the discussion to the behavior of the table, wondering why the system selected that information. Some people tried to touch the table and interact with the graphical interface.

The need of interaction was sometime driven by curiosity but in some occasion it was motivated by an attempt to organize the space on the table, in particular to isolate the objects related to the discussion. One subject also has tried to sweep away a group of objects to clean the stage.

A recurrent pattern, which happens across all groups, is the following. There are some stimuli on the stage; subjects start a discussion about one (always an image); the wizard moves that stimulus in the center of stage, enlarges it and makes it rotate slowly. This pattern shows the employment of two of our strategies applied: the center of the table as a shared place, the dimension of an object to show its relevance to the discussion. In this situation subjects have been talking about the image for a while (two or three minutes, depending on the group). The image in the center has also been exploited to highlight and propose new details to discuss or used as a frame to refer to a part of the fresco.

To provide more evidence about the effectiveness of this pattern we also tried to watch the effect created by its 'negation'. In some situation, when subjects were involved in a discussion supported by an enlarged image rotating in the center, the wizard made it progressively disappear through a fade effect. This behavior created disappointment to all the member of the group ("why? We were talking about that!"). They often tried to justify it as a malfunction of the system.

The wizard also tried to experiment some technique to allow a shift in the topic of the discussion. The first technique exploited makes use of catching effects, like flashing and pulsing. During ongoing discussion, with the relevant image in the center, the wizard has applied a flashing or pulsing behavior to one object already on stage. Subjects noticed the behavior almost immediately but clearly stated disappointment. One subject tried to quit it by touching and asked explicitly to turn it off.

Another technique meant to allow a topic change exploits an orbital pattern of movement. This is supposed to make the subjects recognizing the relation between two objects. The wizard makes an image to orbit around the image being discussed in the center. Subjects do not seem influenced by such a behavior and keep talking about the same topic.

When a topic is concluded, people sometime try to exploit the data on the stage to propose a new discussion. This often happens when a subject points at a visual data close to her and tries to catch others' attention (e.g. "look at this?" or "this is a scene of the first fresco").

Every group, at some stage of the experiment, has tried to "test" the system. This usually happens at the beginning when people try to find out the behavior or the "logic" behind the system. At least a subject for each group tries to say a word and expects the system to show it, sometimes indeed bending to get closer to the table.

Finally people noticed the stimulus that was not related to the domain of the visit (the word "hay making"). Some subject tried to discuss if it was a detail not noticed in the fresco, but nobody was able to find a relation with the current context.

5. INTERVIEWS

After the study, an experimenter debriefed the group about the purpose of the study and conducted a semi-structured interviewed aimed at eliciting the subjective impressions of the group

regarding: the roles of the images and the words displayed, the need of interaction and the role of the system in guiding a conversation.

The system was recognized as a useful tool to wrap up a visit, especially in case people were not acquainted with the exhibition. All the subjects expressed the need of an explicit interaction with the table. This confirms our observation: interaction is not only needed as a way of browsing for more information but also as a way to manipulate objects and organize the space.

Subjects reported the feeling that the table sometimes 'follows' the conversation and tries to propose new hints. They also said to be upset in case of weird behavior, especially when the image supporting the conversation disappears.

As for the conceptualization of the table we can report contrasting opinions. Some subject has considered the table as a tool to satisfy a "knowledge need", e.g. understand better a detail or find further information. Others, instead, have considered the table as a tool to support the sharing of the experience during the visit. The same subjects, in fact, would have liked more images and texts, besides those strictly related to visit, as a way to expand the discussion.

An important discussion emerged about the crowdedness. Sometimes stimuli have been perceived as too many and led to some confusion or indecision on the topic to discuss.

All the groups reported that when the discussion of a topic was finished they exploited the stimuli on the table to start a new conversation.

Finally, graphic-intensive effects like pulsing and flashing, have been considered too upsetting, especially when there is an ongoing conversation. Some subject reported the feeling to be 'forced' to look at those stimuli.

6. CONCLUSION

In this paper, we have presented communication strategies of a system aimed at supporting the conversation of a small group of visitors after a museum visit. The system is designed as a table for the museum café. The aim of this work was to develop and test initial ideas about the impact of such a system may potentially have on the museum visitors.

The results of a WOZ confirmed the communicative efficacy of the prototype and provided some interesting hints for the design. In particular, we realized that interaction could be useful throughout the scenario in order to organize the space or to promote a new topic of discussion. Furthermore, we learnt that users, once a conversation has started and the system supports it through related images, feel upset and irritated if the system changes the contents displayed.

This work represents only the first step of the design process and the results here need to be validated in a more ecological study possibly with different classes of users.

7. REFERENCES

[1] Gaver, W. W., Bowers, J., Boucher, A., Gellerson, H., Pennington, S., Schmidt, A., Steed, A., Villars, N., and Walker, B. 2004. The drift table: designing for ludic engagement. In CHI '04 Extended Abstracts on Human Factors in Computing Systems

[2] Intille, S. S. 2002. Change Blind Information Display for Ubiquitous Computing Environments. In Proceedings of the 4th international Conference on Ubiquitous Computing , 2002

[3] Ishii, H. and Ullmer, B. 1997 Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In Proceedings of Conference on Human Factors in Computing Systems CHI. Atlanta, Georgia pp. 234-241.

[4] Leinhardt, G., Knutson K. 2004. Listening in on Museum Conversations. Altamira Press.

[5] Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Lederer, S., and Ames, M. 2003. Heuristic evaluation of ambient displays. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '03. ACM, New York, NY, 169-176.

[6] McCarthy, J. F., Costa, T. J., and Liongosari, E. S. 2001. UniCast, OutCast & GroupCast: Three Steps Toward Ubiquitous, Peripheral Displays. In Proceedings of the 3rd international Conference on Ubiquitous Computing. Atlanta, Georgia, USA.

[7] Petrelli, D., Not, E. 2005. User-centred Design of Flexible Hypermedia for a Mobile Guide: Reflections on the HyperAudio Experience. In User Modeling and User-Adapted Interaction vol. 15, numbers 3-4 pp. 303-338.

[8] Rocchi C., Stock O., Zancanaro M., Kruppa M., and Krüger A. 2004. The Museum Visit: Generating Seamless Personalized Presentations on Multiple Devices. In Proceedings of the Intelligent User Interfaces. January 13-16, Island of Madeira, Portugal

[9] Streitz, N. A., Rocker, C., Prante, T., Alphen, D. v., Stenzel, R., and Magerkurth, C. 2005. Designing Smart Artifacts for Smart Environments. Computer 38, 3 (Mar. 2005), 41-49.

Exploring Emotions and Multimodality in Digitally Augmented Puppeteering

Lassi A. Liikkanen, Giulio Jacucci,
Eero Huvio, Toni Laitinen
Helsinki Institute for Information Technology HIIT
P.O. Box 9800, FI-02015 TKK, Finland
{firstname.surname}@hiit.fi

Elisabeth Andre
University of Augsburg
Eichleitnerstr.30
86159 Augsburg, Germany
andre@informatik.uni-augsburg.de

ABSTRACT

Recently, multimodal and affective technologies have been adopted to support expressive and engaging interaction, bringing up a plethora of new research questions. Among the challenges, two essential topics are 1) how to devise truly multimodal systems that can be used seamlessly for customized performance and content generation, and 2) how to utilize the tracking of emotional cues and respond to them in order to create affective interaction loops. We present PuppetWall, a multi-user, multimodal system intended for digitally augmented puppeteering. This application allows natural interaction to control puppets and manipulate playgrounds comprising background, props, and puppets. PuppetWall utilizes hand movement tracking, a multi-touch display and emotion speech recognition input for interfacing. Here we document the technical features of the system and an initial evaluation. The evaluation involved two professional actors and also aimed at exploring naturally emerging expressive speech categories. We conclude by summarizing challenges in tracking emotional cues from acoustic features and their relevance for the design of affective interactive systems.

Categories and Subject Descriptors

H5.2. **[Information Interfaces and Presentation]:** User Interfaces – *Input devices and strategies*, *Evaluation/methodology*, *Interaction styles*

General Terms

Design, Experimentation, Human Factors.

Keywords

Gestural interaction, Affective computing, Interactive Installations

1. INTRODUCTION

Natural interfaces can enable users to interact with advanced visual applications in a more embodied and expressive way. The latest development in multimodal processing concerns the tracking of expressive and emotional cues. These new interface technologies hold promise for providing tools to build more empathetic, surprising, and engaging applications. They could lead to innovative applications in which media are not just created and browsed but are also augmented in real time using multimodal and emotionally intelligent inputs. Our vision here is to support performative interaction [7] that encourages users to animate rich media and facilitate the genesis of new formats or practices in the new media field. Initial evidence of the relevance of these practices can be found in naturalistic trials of large multi-touch displays that make possible picture browsing and collage in a bodily way [13] or on systems that support the easy creation of comic strips from mobile pictures [16]. In this area the key research challenges for multimodal and emotion interface technologies include identifying the modalities to be used as input, investigating expressive features in each modality, and finally using them to create engaging interaction loops that motivate users to communicate more expressively.

This paper explores how to use multimodal emotional and expressive cues in digitally augmented puppeteering. The work is organized as follows: 1) reviewing related systems and input components found from the literature; 2) presenting an exemplar application, PuppetWall, that provides a medium for digital puppeteering with editable scenes, props, and puppets, and 3) providing feedback from initial evaluative activities regarding how inexperienced users perform with the help of the system. We conclude by summarizing our findings for future research.

2. RELATED WORK

The previous literature, which we will first consider concerns the development of independent input components for multimodal systems which could also be relevant for digital puppeteering interfaces. The interface can be a data glove and a custom sign language, which directly control the behavior of the digital character (for example, see [2]), without tracking and exploiting expressions or emotions. A more complete approach is I-Shadows [11], an interactive installation which utilizes Chinese shadow puppetry concept for kids creating narrative.

The use of emotion tracking for various kinds of interactive applications has been investigated. These studies may be valuable in showing how to decode or alter the affective states of a user. Existing interactive systems track affective states to influence in a direct or indirect way the essential contents of an interactive application. McQuiggan and Lester [9] have designed agents that are able to respond empathically to the gaming situation of the user. AffectivePainting [18] supports self-expression by adapting instantaneously to the perceived emotional state of a viewer, which is recognized from his or her facial expressions. Some empathic interface agents apply physiological measurements to sense users' emotional states

AVI'08, May 28-30, 2008, Napoli, Italy.

[15]. Gilleade et al. [5] measure users' frustrations to drive the adaptive behavior of an interactive system. There are also systems that extend the concept of empathy to account for the relation developed between the user and a virtual reality installation [8]. Cavazza et al. [3] present multimodal actors in a mixed-reality interactive storytelling application in which the positions, attitudes and gestures of the spectators are tracked, influencing the unfolding of the story.

Camurri et al. [2] introduce what they call multisensory integrated expressive environments as a framework for mixed-reality applications for performing arts and culture-oriented applications. They report an example where the lips and facial expression of an actress are tracked and the expressive cues are used to process her voice in real time. SenToy [12] allows users to express themselves by interacting with a tangible doll that is equipped with sensors to capture the users' gestures.. Isbister et al. [6] study uses 3D shapes to communicate emotions to the system and to the design team. However, we are not aware of any work which has applied the tracking of expressive cues from actors to animate or control puppet-like virtual characters.

3. PUPPETWALL

PuppetWall is a multi-user, multimodal installation for collective interaction based on the concept of traditional puppet theatre. When interacting with PuppetWall, users hold a wand in their hands that controls a puppet on a large touch screen in front of them. The touch screen is used to manipulate the playground, which consists of characters, props, and a background. The aim is to provide a platform for exploring emotion and multimodality with an interactive installation. Here we report on the design and details of the first prototype application.

3.1 System Overview

The PuppetWall system includes several input modalities for explicit and implicit control and a large multi-touch screen to visualize and edit the visual animations and scenes. An overview of the system is shown in Figure 1. The hardware of the prototype consists of both standard equipment and custom-made devices. The application runs on a single relatively high-performance PC (as of 2007) workstation and utilizes a Linux operating system. The workstation is equipped with IEEE1394 (FireWire) ports and a 3D accelerated graphics adapter. Input/output devices include a standard stereo microphone, pair of active speakers, a video projector (DLP, 1280 x 768 pixels), and three high-speed, high-resolution FireWire digital cameras, one of them equipped with an infra-red (IR) filter and a wide-angle lens (see 3.2.3). Interaction with the system is based on

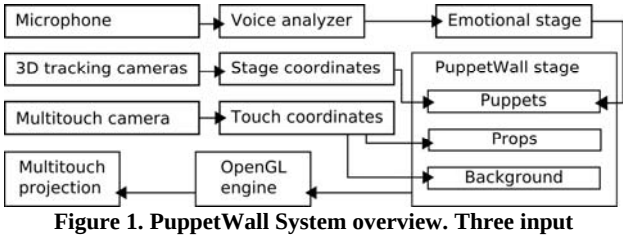


Figure 1. PuppetWall System overview. Three input components. Speech for tracking emotional state, 3D tracking of character control, and a touch screen to interact with objects or to edit puppets.

three inputs: hand movements via the detection of the MagicWand movements (see 3.2.1), direct manipulation through a touch screen (visualized in Figure 2; see 3.2.3) and voice input for tracking acoustic features of speech (see 3.2.2). The application reacts to these inputs to produce a visual 2.5-dimensional representation of virtual puppet theatre playground.

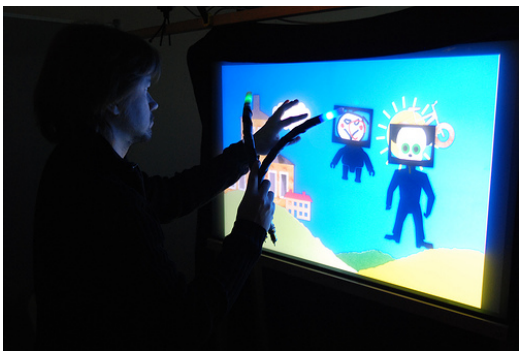


Figure 2. Prototype of PuppetWall and a user holding a MagicWand in their right hand and interacting with a prop with their left. There are two characters on the stage, partially hiding two props (then sun and a bicycle).

3.2 Input modalities

3.2.1 MagicWands for 3D hand tracking

The characters on stage are controlled with custom-made wands (MagicWands) which incorporate a light source, a single LED of variable color. This concept is similar to that of VisionWand [20]. Characters are moved and rotated according to the motion of the illuminated end of the wand and the users can mover one or more wands to control the motion of the puppets. A MagicWand is a plastic stick approximately thirty centimetres in length, consisting of a power source and a super-bright LED at the top end. Wands were assembled using standard electronic components. The super-bright LEDs are detected using a pair of digital cameras operating at 30 frames per second and mounted above the display. The camera image is used to calculate the 3D position of each wand. This happens by comparing the location of a bright spot on both of the camera images and the imaginary normal lines of the cameras. The movement is then interpreted into two-dimensional movements relative to the screen. All three coordinates can be used to control the character and different colours in the LED of the MagicWands are used to differentiate the characters. The wands are equipped with a power switch.

3.2.2 Emotional speech recognition

One essential requirement of the system is to be able to detect and respond to the user's emotions. Currently we are attempting to achieve this using emotional speech recognition. The user's voice, an essential element in building the narrative in this interactive storytelling environment, is captured using a single stereo microphone which feeds into a speech classifier. The classifier, called EmoVoice, is based on Naïve Bayesin classification of reduced feature sets (see [19]). This means that in the target language it has been trained to categorize the component should be able to discriminate between the defined emotional categories from arbitrary spoken input. The training of the initial version of the classifier was carried out with an

extensive enacted Finnish emotional speech corpus including six emotion categories (see [17]). In an off-line state, it achieves some 45% accuracy. The preliminary setup is intended for testing (see Initial Evaluation below) and the hardware and the training corpus are subject to change in the future.

3.2.3 Touch screen for direct manipulation of objects and characters

A multi-touch screen (1 m wide) is used for displaying the PuppetWall playground and allows objects (props) and characters to be manipulated directly. The system enables multiple hand-tracking and individual hand posture and gesture tracking. Interfacing the screen is based on detecting changes in the IR luminosity from the screen surface, relying on a high-resolution, high-frequency camera and a robust computer vision algorithm. This concept is similar to HoloWall [10]. The technique requires an IR illumination of the screen from behind to level the incoming background IR signal. The movements on the screen surface are captured by the camera, which is also located behind the screen. A diffusing surface which is attached to the back of the screen surface blurs the object's IR image of the object, but when a user touches the screen it will show as a bright sharp spot in the IR camera image. These technological features create the conditions for a multi-user and multi-touch installation appropriate for public spaces (cf. [13]).

3.3 Visual Outputs

The main view of PuppetWall interface is called a playground and is comprised of characters, props, and the background (see Figure 2). All visuals are created with a custom-made 3D graphical engine based on OpenGL libraries. The interface presented on the touch screen is created with one, two or four projectors. The resulting screen resolution is a multiple 1280 x 768 pixels. The current prototype employs one projector.

3.3.1 PuppetWall basic view

Puppets are moved according to the movement detected from MagicWands. Puppets are able to swing around their pivot point so when the MagicWand is moved swiftly rotationally they also do a full rotation. Props – clouds, buildings and vehicles – can be moved and manipulated (resized, transformed, moved) on the fly by touch and gestures. Objects can be re-sized by touching them with multiple fingers and pulling touch points closer or pushing them further away from each other. Vehicle and building props will change into a different, larger one when certain size is reached. As an example: the positions of the sun and moon can be changed by rotating the plane containing them. They are placed on the opposite sides of the plane and the lighting conditions of the stage will change according to the state of the plane; it is lighter when the sun is up. The background elements are currently stationary.

3.3.2 Character editing mode

When a puppet is touched, the system enters an editing mode illustrated in Figure 3. In this mode, the user can modify the character by changing the puppet's head or body. They are lined up on the screen and the user can select different ones with a pulling gesture so that the desired shape moves toward the center (gesture-based browsing). Heads can be customized by drawing over the face with a finger, enlarging or shrinking the face or moving its relative position in the head frame.

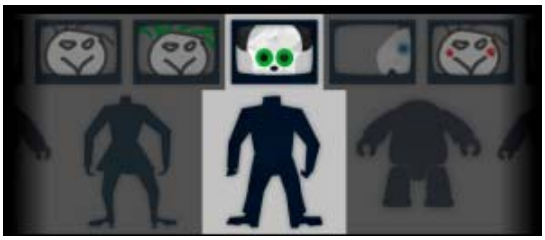


Figure 3. The editing mode. Heads and bodies can be selected by pulling them into the highlighted area.

4. INITIAL EVALUATION

An informal evaluation was organized with two performing arts professionals with a background in improvisational theater. They were involved in an explorative session with the first functional prototype. In the session, they experimented with in improvised and directed story-telling using the system for the first time. The experimental session was videotaped and the audio was additionally recorded with collar microphones to compile a corpus of naturally occurring interaction. The session began with a minimal debriefing and ended up with a structured interview for feedback from the interactive session.

It was observed that the actors could easily utilize the installation with minimal instructions. The actors used the touch interface to modify the puppets and the props, and MagicWands were used successfully to animate the characters. The users seemed to enjoy PuppetWall and created eight short stories with it. The actors, familiar with improvisation, suggested the use of implicit interaction strategies, for example using breathing sensors and adding more surprising elements to the scene. Also the actors complained that while constraints are useful to make things happen in improvisation, they felt too limited by the control of the puppet, as they could not implement all ideas. In addition to new development ideas and usability issues observed from video the analysis, we found that the corpus extracted from the speech naturally elicited during the session, could be meaningfully classified with EmoVoice. The most reliable classification appeared between what could be called 'user' and 'character' voices (68% off-line discrimination). The user voice was low, inactive, and constrained whereas the 'character' voice was active, engaging, and openly emotional.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced PuppetWall, an interactive application for exploring affective interaction and multimodal inputs in an environment intended for multiple, simultaneous users. In addition to providing the technical details, we have described an informal evaluation of the system. Our evaluation demonstrated the feasibility of the concept and also provided a preliminary corpus of affective speech. An important result from the analysis of the corpus with the EmoVoice classifier was the demonstration of how the neutral user and character voices are differentiated along a 'dimension of activation'.

In the future, even if we are able to confront the problem of how to decode user emotions, we still face the additional problem of responding to these emotions. While decoding has received a lot of attention, the other half of the work has barely started and currently, no clear guidelines exist on how to engineer affective responses or to augment emotion. In current HCI, the best-

known collection of techniques is called Emotioneering [4] a set of heuristics for emotional game design. Their problem is their considerable domain dependence. Only a few heuristics, such as the use of symbols, can be transferred to other domains. Additional examples from the literature show context-dependent solutions, for instance analyzing call center requests for later prioritization according to affective status [14] or applications utilizing emotion recognition in the form of a game to help individuals to recognize and manage emotions [1, 14]. One generic approach available in some contexts, as with PuppetWall, might be to recruit professionals in the domain in question to participate in the design process. This co-design can be helpful to exploit the vast knowledge that the experts possess about expressivity in their domain.

In conclusion, the prototype of PuppetWall presented here is the first step in developing a platform studying multimodal and affective interaction techniques. While this step provided useful indications on the feasibility and relevance of the concept, several questions remain open, for instance, which modalities to address and how gestural information could be utilized. However, from the initial evaluation and later co-design (to be documented) we have gained considerable knowledge and many ideas for future development and user research that will, we hope, highlight PuppetWall as a state-of-the-art example of collocated, emotionally augmented interactive installation.

6. ACKNOWLEDGMENTS

This work has been funded by the European Union (EU) 6th Framework Research Programme project CALLAS (ref. 034800). We are grateful to Tommi Ilmonen and John Evans, who designed and implemented the prototype, and thank Jérôme Urbain and Stephen W. Gilroy for providing some references.

7. REFERENCES

[1] Bersak, D., McDarby, G., Augenblick, N., McDarby, P., McDonnell, D., McDonald, B. and Karkun, R. Intelligent biofeedback using an immersive competitive environment. In Proceedings of the Designing Ubiquitous Computing Games Workshop at UbiComp (2001).

[2] Camurri, A., Volpe, G., De Poli, G. and Leman, M. Communicating expressiveness and affect in multimodal interactive systems. IEEE Multimedia, 12, 1 (Jan-Mar 2005), 43-53.

[3] Cavazza, M., Charles, F., Mead, S. J., Martin, O., Marichal, X. and Nandi, A. Multimodal acting in mixed reality interactive storytelling. IEEE Multimedia, 11, 3 (Jul-Sep 2004), 30-39.

[4] Freeman, D. Creating emotion in games. The craft and art of emotioneering. New Riders, Indianapolis, IN, 2003.

[5] Gilleade, K. M. and Dix, A. Using frustration in the design of adaptive videogames. In Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology (Singapore, 2004). ACM Press.

[6] Isbister, K. and Hook, K. Evaluating affective interactions. International Journal of Human-Computer Studies, 65, 4 (Apr 2007), 273-274.

[7] Jacucci, G. Interaction as Performance. Doctoral dissertation, University of Oulu, 2004.

[8] Lugrin, J. L., Cavazza, M., Palmer, M. and Crooks, S. AI-Mediated Interaction in Virtual Reality Art. In Proceedings of the Intelligent Technologies for Interactive Entertainment: First International Conference (INTETAIN 2005) (Madonna di Campiglio, Italy, 2005). Springer-Verlag.

[9] McQuiggan, S. W. and Lester, J. C. Modeling and evaluating empathy in embodied companion agents. International Journal of Human-Computer Studies, 65, 4 (Apr 2007), 348-360.

[10] Nobuyuki, M. and Jun, R. HoloWall: designing a finger, hand, body, and object sensitive wall. In Proceedings of the 10th annual ACM symposium on User Interface Software and Technology (UIST) (Banff, Alberta, Canada, 1997). ACM.

[11] Paiva, A., Fernandes, M. and Brisson, A. Children as affective designers - i-shadows development process. Humaine WP9 Workshop on Innovative Approaches for Evaluating Affective Systems(2006), (accessed,

[12] Paiva, A., Prada, R., Chaves, R., Vala, M., Bullock, A., Andersson, G. and Höök, K. Towards tangibility in gameplay: building a tangible affective interface for a computer game. In Proceedings of the 5th international conference on Multimodal interfaces (Vancouver, BC, 5-7 November, 2003).

[13] Peltonen, P., Kurvinen, E., Salovaara, A., Jacucci, G., Ilmonen, T., Evans, J., Oulasvirta, A. and Saarikko, P. "It's mine, don't touch!": Interactions at a large multi-touch display in a city center. In Proceedings of the CHI2008 (to appear, 2008).

[14] Petrushin, V. A. Emotion Recognition In Speech Signal: Experimental Study, Development, And Application. In Proceedings of the Sixth International Conference on Spoken Language Processing (ICSLP 2000) (Beijing, China, 2000).

[15] Prendinger, H. and Ishizuka, M. Human physiology as a basis for designing and evaluating affective communication with life-like characters. IEICE Transactions on Information and Systems, E88D, 11 (Nov 2005), 2453-2460.

[16] Salovaara, A. Appropriation of a MMS-based comic creator: from system functionalities to resources for action. In Proceedings of the SIGCHI conference on Human factors in computing systems (San Jose, CA, April 28-May 4, 2007). New York, NY: ACM Press.

[17] Seppänen, T., Toivanen, J. and Väyrynen, E. MediaTeam speech corpus: a first large Finnish emotional speech database. In Proceedings of the Proceedings of XV International Conference of Phonetic Science (Barcelona, Spain, 2003).

[18] Shugrina, M., Betke, M. and Collomosse, J. Empathic painting: interactive stylization through observed emotional state. In Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering (NPAR 2006) (Annecy, France, 2006). ACM Press.

[19] Vogt, T. and Andre, E. Comparing Feature Sets for Acted and Spontaneous Speech in View of Automatic Emotion Recognition. Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on (2005), 474-477.

[20] Xiang, C. and Ravin, B. VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. In Proceedings of the 16th annual ACM symposium on User Interface Software and Technology (UIST) (Vancouver, Canada, 2003). ACM.

Face Bubble: Photo Browsing by Faces

Jun Xiao
Hewlett-Packard Laboratories
Palo Alto, CA USA
jun.xiao2@hp.com

Tong Zhang
Hewlett-Packard Laboratories
Palo Alto, CA USA
tong.zhang@hp.com

ABSTRACT
Face recognition technology presents an opportunity in computer automation to help people better organize their personal photo collections. However, the robustness of the technology needs to improve and how users interact with face clusters needs to go beyond the traditional file folder metaphor. We designed a visualization called face bubble that supports both fast one-glance view and filtering and exploration of photo collections based on face clustering results. Our clustering algorithm provides a better accuracy rate than previous work and our circular space filling visual design offers an alternative UI based on the traditional weighted list view. Other visualization techniques such as a fisheye view are also integrated into the interface for fast image browsing. Finally, fine tuning of the design based on user feedback improved the aesthetics of the visualization.

Author Keywords
Face detection, face recognition, clustering, filtering, and visualization.

ACM Classification Keywords
H5.2.User Interface: [Graphical user interfaces (GUI), User-centered design].

INTRODUCTION
Digital cameras and mobile phone cameras have become increasingly ubiquitous and the cost for people to take and store the photos has decreased rapidly over time. As a result, the size of personal digital photo collections is growing exponentially. Many commercial applications and services, such as Microsoft's Photo Gallery, Adobe's Photoshop Elements, Goolge's Picasa, and Yahoo's Flickr, try to better support users in searching and organizing photo collections, but they often require people to manually enter and apply tags or labels to the photos. The main challenge for image search and organization, we believe, is how to make related user tasks easy and intuitive and the experience enjoyable and intriguing. We intend to apply automation to augment human perception and cognition abilities when it is most advantageous to do so, and enhance the user experience by introducing automation seamlessly into the workflow without adding unnecessary complexity and being intrusive.

AVI'08, 28-30 May , 2008, Napoli, Italy

In particular, we focus our effort on typical consumer photo collections taken with point and shoot cameras. For those photos, the most important metadata besides a timestamp are the subjects, that is, people who are in the pictures. This presents a unique opportunity and challenge for computer automation to help people organize their personal photo collections. On the one hand, users often find it difficult to retrieve photos of a particular person or group of people out of their ever increasing photo collections to create a photo book. On the other hand, current automatic face detection and recognition technologies are not robust enough to consistently classify people's faces with varying lighting conditions and scenes and different facial expressions and orientation. The false positive and false negative rates may require excessive user intervention or supervising to correct automation mistakes.

In this paper, we describe a novel user interface that takes full advantage of an advanced face clustering technology. In the next section, we first briefly review several previous works on automatic grouping and annotating photo collections with image metadata.

RELATED WORK
The major hurdle for computer-aided media organization is the semantic gap. Researchers have explored various paths to bridging this gap with regard to personal photo collections. Knowing *when* and *where* the photos are taken, and *who* are in the photos, for example, can greatly help interpret *what* the photos are about.

Automatically detecting time events in photo collections has been well studied [6, 8]. The algorithms for clustering photos using timestamps vary on how event segmentation is conducted. Apple introduced time clustering in commercial software via iPhoto'08[1]. The system maybe is not novel on the automatic creation of time events, but it has intuitive event viewing, splitting and merging user interface. For example, a new feature called skimming allows users to

slide the mouse across the event icon to quickly browse all the photos within an event.

Lately, rapid improvement in GPS and cellular technology enables more people to add location information to captured photos. An increasing number of photos over the web are tagged with the exact coordinates at which they were taken. Research has been conducted to cluster such geo-referenced photos geographically and display them on digital maps and as a result, add rich spatial context for image search and organization [2, 9].

Besides time and location metadata, researchers have long tackled the problem of identifying the people who appear in photo collections [5], which arguably is a more relevant problem for personal photo collections. However, with more than a decade of face detection and recognition research, there is still a gap for reliable image retrieval from collections using such technologies. A complicating factor in consumer photo collections is that faces in those photos are not always well-lighted or well-composed, which makes even face detection, let alone recognition, a difficult task. To mitigate the clustering errors, interactive semi-automatic cluster annotation mechanisms have been proposed [4].

FACE CLUSTERING ALGORITHM

Our face clustering algorithm involves multiple steps. First, face regions are detected in each photo in the collection and a skin tone filter is employed to screen out false alarms. Facial features are extracted and similarity values between each face pair are computed by a face recognition engine to form an affinity matrix. Agglomerative clustering begins with an initial partition where each instance forms a singleton cluster. Then the most similar two clusters are selected and merged to one cluster. The merging operations repeat until the similarity value between the two merging clusters falls below a stopping threshold. One important auxiliary constraint is followed, however. Faces appearing simultaneously in a photo must belong to different people. This constraint controls the merging procedure as supervised learning. There are two passes in the clustering procedure. We first employ complete-linkage algorithm to ensure each cluster only contains the most similar faces of the same person [7]. Then a single-linkage algorithm is used upon obtained clusters to consolidate similar clusters until the similarity value between the nearest clusters drops below a threshold. Additionally, we applied the k-nearest neighbor classifier to further consolidate the face clusters. In particular, faces in small clusters are compared with face models in larger dominant clusters.

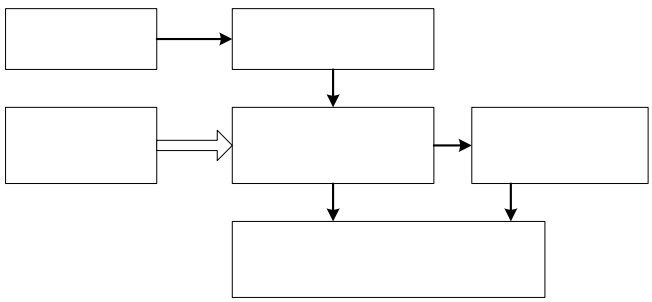


Figure 1. Framework of face clustering algorithm.

Testing results of the face clustering algorithm are reported in [7]. Although there were many situations where the faces in pictures were so dark, noisy and distorted even for human beings to distinguish, the face clustering algorithm achieved 58% recall and 96% precision rate. This result indicates that faces within one cluster mostly do belong to just one person. But faces of the same person may be split into multiple groups. Without the final cluster consolidation step, the algorithm can actually achieve 100% precision rate with 46% recall rate. We further exploited this performance characteristic in our design of the face bubble user interface and made merging two groups much easier than splitting groups.

CIRCULAR SPACE FILLING ALGORITHM

First introduced by the online photo sharing site Flickr, tag clouds have become increasingly popular among community websites. More frequently used tags or words are depicted in a larger font or otherwise emphasized. There are several variations in visual design on how the tags should be laid out and clustered together, but often they are based on text blocks and are rectangular in shape. The use of rectangular, space-filling regions in visualization is actually a well-studied topic and the tree-map is a notable technique in this area [3]. Many variations of the basic tree-map technique have been proposed, but most are space filling representations of a tree using nested rectangles.

Inspired by the idea of tag clouds, we designed a weighted list visual representation of face clusters that affords one-glance viewing for users to get an overall sense of their photo collections. Because people's faces naturally lead to round shapes, we felt that a design as shown in Figure 2 would be a good visual mapping of the face clusters, with the sizes of the bubbles representing the sizes of the clusters. This calls for a space filling algorithm that works for circular shapes.

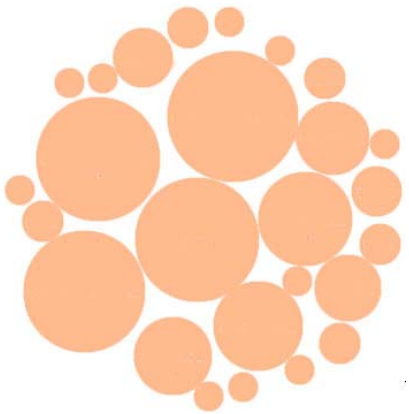


Figure 2. Weighted list visual representation of face clusters.

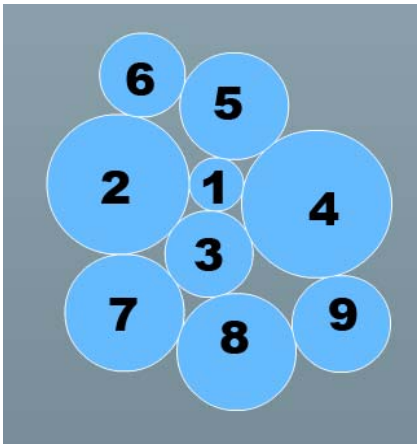
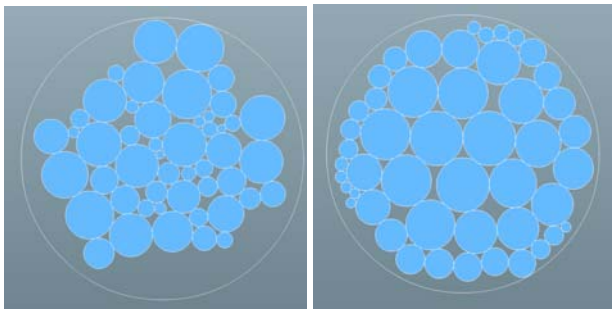


Figure 3. Circular space filling uses a greedy packing algorithm to generate the layout.

The input of the algorithm is a list of numbers depicting the relative sizes of each bubble and the overall bounding circle. The output of the algorithm should be the locations of the bubbles, given that the bubbles are packed to each other as close as possible. As shown in Figure 3, the circular filling algorithm is a greedy algorithm. It works by positioning the next bubble tangent to the two existing bubbles that are closest to the center without intersecting with any other bubbles. Although this greedy algorithm may not most efficiently use the given space, it can provide a very good approximation to the solution when the number of bubbles is reasonably large (n>50) (see Figure 4a). If the algorithm performs a sort in bubble size on the input list, the layout solution will be optimal (see Figure 4b). However, although such a layout most efficiently uses the space, it lacks the randomness that makes the view as intriguing as the tag cloud. In addition, preserving the order of the input list might be important in certain scenarios.

After reaching an initial layout by the greedy search, the algorithm performs several optimization steps (see Figure 5). First, it translates the bounding circle to find a smaller bounding radius. Then it attempts to move the bubbles toward the bounding circle's perimeter. Any intersections of the bubbles are checked and the moving directions of the bubbles are randomized. The end result is a visually more



(a) (b)

Figure 4. (a) Random layout or (b) sorted layout.

(a) (b) (c)

Figure 5. Fine tunings of bubble layout: (a) original circular space filling layout result (b) after translating the bounding circle (c) after randomly dispersing the bubbles

pleasing graph with less space requirements. On a 2.4G mainstream PC, it takes around 0.1 second for the algorithm to generate a layout with a thousand random bubbles.

FACE BUBBLE VISUALIZATION

By applying the circular filling algorithm to face clustering results, we developed an interactive visualization in Flex. Face clustering output from the clustering algorithm is streamed into the Flash program in XML format (see Figure 6a) and the most representative faces are cropped with masks according to the region coordinators specified in the XML. The faces are scaled in sizes with regard to the cluster sizes. Instead of being instantly visible, the faces smoothly sweep in and increase its alpha value with randomized delay and speed, thus providing a bubbling effect. Also, various hidden lightening masks with a blurring filter are created on the background so that each bubble has a dissolving effect around the edges. This gives a subtle watercolor effect to the bubbles. Finally, upon testing the design prototype with users, we found that packing the bubbles too closely actually worked against the visual aesthetics of the display. Therefore, we created small gaps between bubbles in order to separate the bubbles more (see Figure 6b).

We defined interaction mechanisms for the interface where

(a) (b)

Figure 6. From (a) face clustering result to (b) face bubble visualization.

Figure 7. Zoom in on one person (the male face on the bottom of the face ring).

Figure 8. Fisheye image browsing interface.

users can merge clusters by drag and drop and animation shows the bubble merging effect. Users can also click on the background to request an alternative layout. Sizes of the clusters appear as tool tips that shows the number of photos a particular person has in the collection. Users can also annotate the faces by typing directly over the face bubbles.

In addition to the one-glance view of photo collection, users can further explore the clusters. Users can focus or zoom in on particular face (see Figure 7) and then all bubbles are pushed to the boundary with the zoomed-in face highlighted on the face ring. The inner circle shows people who have been in photos with the focused person. We again use the circular space filling algorithm here and the size of bubbles corresponds to the co-occurrence frequency of the persons. Users can directly click on the face bubbles on the boundary to inspect other face clusters to quickly gain knowledge about the relationship between people in the photo collection. Or users can click on the face bubbles in the inner circle to inspect the actual photos that have both the person on the boundary ring and the click target. Finally, we also designed a fisheye image browsing UI to visualize the individual or group photos (see Figure 8). It allows fast and smooth scrolling through image collections.

FUTURE WORK

We will continue our research to improve the face clustering and representative face selection algorithm. We plan to conduct formal user studies with personal photo collections and real tasks, comparing face bubble with other photo organizers and browsers. We also intend to explore better ways to visualize uncertainty in the clustering.

REFERENCES

1. Apple iPhoto'08. http://www.apple.com/ilife/iphoto/.
2. Ahern, S., Naaman, M., Nair, R. and Yang, J.H. World explorer: Visualizing aggregate data from unstructured text in geo-referenced collections. *Proc. JCDL 2007*, ACM Press (2007), 1-10.
3. Bederson, B. Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. *Proc. UIST 2001*, ACM Press (2001), 71-80.
4. Cui, J., Wen, F., Xiao, R., Tian, Y. and Tang, X. EasyAlbum: An interactive photo annotation system based on face clustering and re-ranking. *Proc. CHI 2007*. ACM Press (2007), 367-376.
5. Girgensohn, A., Adcock, J., Wilcox, L. Leveraging face recognition technology to find and organize photos. *Proc. MIR'04*, ACM Press (2004), 99-106.
6. Graham, A., Garcia-Molina, H., Paepcke, A., and Winograd, T. Time as the essence for photo browsing through personal digital libraries. *Proc. JCDL 2002*, ACM Press (2002), 326-335.
7. Gu, L., Zhang, T. and Ding, X. Clustering Consumer Photos Based on Face Recognition. *Proc. ICME07*, IEEE (2007), 1998-2001.
8. Platt, J.C., Czerwinski, M. and Field, B.A. Phototoc: Automatic clustering for browsing personal photographs. Technical Report MSR-TR-2002-17, Microsoft Research, February 2002.
9. Toyama, K., Logan, R., Roseway, A. and Anandan, P. Geographic location tags on digital images. *Proc. ACM Multimedia'03*, ACM Press (2003), 156-166.

Browsing a Website with Topographic Hints*

S. Rossi, A. Inserra and E. Burattini
Dip. di Scienze Fisiche
University of Naples "Federico II"
Naples, Italy
silrossi@unina.it,ernb@na.infn.it

ABSTRACT

This work aimed to propose an adaptive web site in the field of cultural heritage that can dynamically suggest links, based on not intrusive profiling methodologies integrated with topographical information. A fundamental issue, typical in web sites that refer to real sites, is to help the user to orient himself geographically. Our system can support the user in its exploration of physical/virtual space suggesting new physical locations structured as a thematic itinerary through the excavations.

1. INTRODUCTION

Web personalization is the process of adapting the information content and the format of an interface in order to meet the individual needs of a single user, starting from his browsing behavior and from his interests. A particular class of these systems is represented by the Recommendation Systems. Such systems are meant to suggest links to products, services and information which are considered to be attractive for the user. Such techniques are often considered an essential part for the customization process of web sites because they support the mechanism of adaptation to the characteristics of each user [5]. Recommendation Systems and Adaptive Hypermedia are intimately linked: they both have the goal to identify the contents that may correspond to the user interests, but while Adaptive Hypermedia aims to filter information, the goal of Recommendation System is to provide additional sources of information.

Most of the Recommendation Systems require an active participation from the user. They require that the user makes his informative needs explicit (for example, by providing the system with series of keywords that better represent his interests), or require that the user expresses an opinion on documents (for example, by assigning a vote). While these methodologies are useful in the case of selling advices, in our opinion they cannot be easily applied in the case of navigation in websites. These methodologies are highly invasive and there is not only the risk to bore the user, but also to make the system not ductile to quick changes in the behavior of users over time. Faced with these problems, we draw attention especially to those systems that, using an implicit user profiling, implement adaptivity and recommendation simply by observing and analyzing the user interaction with the system [3, 6].

The goal of this work was to design and implement a recommendation system in the field of cultural heritage, with the aim of helping and involving the users during the visit of a web site, by identifying the different types of users and creating the appropriate suggestions. These recommendations are meant to help the user in finding interesting contents and, at the same time, to facilitate the user to orient himself in hyperspace integrating a physical space with a virtual one [7]. The system, in fact, starting from the user profile, integrated with the topographical information contained in the database, helps the user to find a "path" throughout the navigation.

One of the problems, in large web sites, is not just of finding the right contents but also to locate and recover information one has seen before [2]. During the last years, spatial navigation has been proposed as a new metaphor of interaction in large portals, supported by the possibility of having geo-localized maps and portable GPS devices. However, this methodology has also been proposed as a way to enhance the interaction in common World-Wide Web sites, where the lack of apparent structure and the hyper-link navigation make the interaction problematic [8]. In the case of web sites connected to a real site, like a museum or, as in our case, an excavation site, the underneath spatial structure of the contents is already present and can be easily used to help the user to orient himself geographically, namely to understand the nature of information that surrounds him. In this way the user can perceive his location in the virtual space and consequently in the corresponding real space. More specifically, the objective of the system is not only to select relevant information for a user, but also to adjust the level of presentation of each topic to the physical space of the Herculaneum excavation. Therefore, we have the problem to choose what are the information to show (as in all personalization systems), but also to generate recommendations as much as possible appropriate to the space context, generating a personalized tour. So, our system can also support the user exploration of physical/virtual space, helping him to find what he is looking for and suggesting the new physical locations structured as a thematic itinerary through

*This work was partially supported by the S.Co.P.E. project.

AVI '08, 28-30 May , 2008, Napoli, Italy.

the excavations.

Finally, let us consider that a website related to a real site may be also used as a support before and during the visit [7]. To provide the user with virtual thematic paths on the web site, which reflect real possible paths, may be useful to remember contents, starting from their relative locations, and to train the user for the real visit. Human beings have naturally the ability to recall objects in space depending on their locations.

1.1 The Herculaneum Excavations Hypermedia

The Herculaneum Excavations Hypermedia[1] comes from a similar system realized stand-alone, on CD-ROM support [9], and from an earlier on-line version presented in [10]. The knowledge base contains a huge set of information, including historical data and mythological tales, excavation reports, description of recovered buildings and objects, painting and so on. This information comes under the form of texts, videos, images, contemporary and old maps. The knowledge base was appropriately linked – following the experts' suggestions – and the results of this analysis have been summarized in a tree where single chunks of knowledge are identified and displayed (see Fig. 1). The node labeled "building" represents one of the main points of view from which the knowledge on Herculaneum Excavations may be explored and sub-nodes show the ways in which such knowledge may be enhanced.

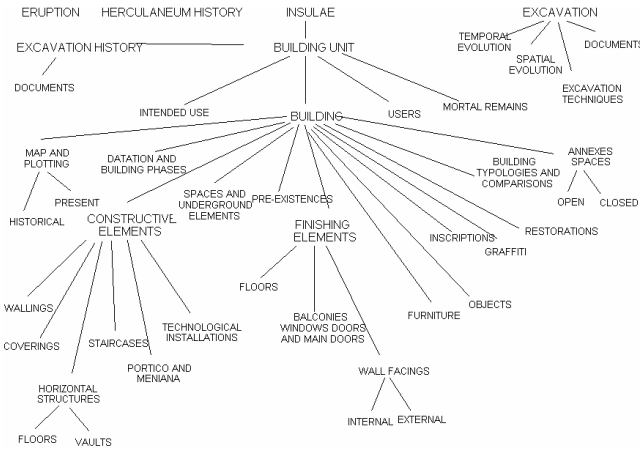


Figure 1: The knowledge tree for the Herculaneum Excavations Hypermedia.

In [10] we proposed a context independent architecture composed by modules that can be reused and adapted. Our system has no *a priori* knowledge of the web contents, but is able to manage a high level representation (a knowledge tree) defined on this set. A content element in the database represents an instance of a node of the tree. For example a resource instance may be the "excavation report" of a particular building or the "description of a fresco". Such instance gets the description of its correspondent class of the tree.

The work presented here, starting from the purposes of his predecessors, progresses using a different technique for the user modeling and, thereafter, for the personalization of the contents. Moreover, we integrated a recommendation facility based on an implicit user modeling and topographical information.

[1]http://www.ercolano.unina.it

2. MODELING USERS AND RESOURCES IN THE SAME SPACE

The browsing behavior and the selection of particular information content are a source of knowledge in order to select relevant contents and to modify the layout of an interface. The user model has to be modified by the choices made by the user. Moreover, the selection of resources "preferred" for the user depends on the state of the user model. This leads us to the need to represent the totality of resources through a resource model that can be easily linked to the user model.

With the help of a group of architects, historians and archaeologists we detected the main properties that may characterize any information content of our hypermedia. Those features are related to the types of informative content that a resource may have and to the interests that a resource may match. For example, a feature is the "historical content" - i.e. how much a resource class contains information from historical data. We defined a resource class as a vector $\overrightarrow{r} = (w_1, \ldots, w_n)$ in the space $\mathcal{R}^n$, where n is the number of characteristics or features. The vector model of a resource class represents how much every specific characteristic is present in the class of resources. Differently from the common usage in data mining, the vector, which represents a class of resources, does not contain the frequency of occurrence of some specific words within the text [4]. It is representative for the whole class, it does not depends on a single text and represents how much a specific class can be classified according to some typologies of texts. Our approach is more similar to the "Concept Profiling" methodologies [11], in fact, all the resource classes represent abstract topics, rather than specific words or sets of related words. Moreover, they are organized in a hierarchical structure and the relationships between resources are implicitly specified.

In the recommendation module developed for the Herculaneum Excavations we decided to implement a user model that takes into account only the current behavior of the user, starting from the specification of the behaviors of "ideal" users. Differently from the classical use of stereotypes [1] we do not try to classify a user in a specific class, but we start from the assumption that a real use may exhibit a behavior that is a combination of ideal classes. A user model is a vector $\overrightarrow{u}$ in the space of ideal user classes. When a user session starts, the user model vector components are set to zero. The user model vectors are represented as attribute–value couples and such values represent the percentage of similarity of the user model to ideal user classes. This is to say that the user model $\overrightarrow{u_j}$ of the user j is a linear combination of ideal user's models ($\overrightarrow{u_i}$), such as $\overrightarrow{u_j} = \alpha\overrightarrow{u_1}, \beta\overrightarrow{u_2}, ..., \gamma\overrightarrow{u_n}$, where Greek letters represent percentages. It is fundamental that each ideal class of users does not share any behavior with other ideal classes, i.e. the vectors representing ideal classes are an orthogonal base.

One of the main problems of systems that use keywords representations is that the user model and the resources are not directly comparable. In our system, however, the user model can be mapped in the same space of resources. In fact, each ideal user is characterized by an ideal resource, representing the optimum content for that particular ideal

user ($\overrightarrow{u_i} \rightarrow \overrightarrow{r_i}$). This set of optimal resources is an orthogonal base in the space of resources. The vectors $\overrightarrow{r_1}, \ldots, \overrightarrow{r_n}$ of optimum resources for ideal users represent a squared matrix U that we use in order to evaluate the ideal resource for the current user, to give recommendation (see Sec.3), and to modify the user model according to his browsing behavior. Let us highlight that an optimal resource $\overrightarrow{r_i}$ may not correspond to any of the real resources $\overrightarrow{r_j}$ in the database.

Every time the user selects a resource, the user model has to be update. The selection of a resource can be either a click on a link or other actions enabled by the systems, such as saving personal notes, search on the database, and so on. Every time the user selects a resource $\overrightarrow{r_k}$ we evaluate the correspondent user model for that particular resource: $\overrightarrow{u_k} = \overrightarrow{r_k} \times U^{-1}$. This vector $(u_1, u_2, \ldots, u_{14})$ represents qualitatively how much we have to modify the current user model in order to take into account his last action. In particular, we modify the current user model making a weighted average between the current values (u_u) of the user model and those arising from the last interaction with the system (u_k). The new components of the user model will be:

$$u_{new} = \frac{(u_u * n_{click}) + u_k}{n_{click} + 1},$$

where n_{click} is the number of interactions with the system.

In this way we can minimize errors because, if a user clicks only few times on different topics, this will have a little effect comparing to the whole interaction. Moreover, we are able, after few interactions, to have a well defined user model in order to give recommendations.

3. SELECTING THE RESOURCE CLASS

In the previous section we discussed how the user model is modified according to his actions. In this section we will explain how the system selects the interesting resources according to the user model. Starting from the current user model, the system evaluates the ideal resource $\overrightarrow{r_i} = \overrightarrow{u_i} \times U$ for the user. As we already said, to an ideal resource may not correspond any of the real resources, and so, the system evaluates the "distance" from this vector to the real resources $(\overrightarrow{r_j})$ in the web site.

To evaluate the distance between real resources and the ideal resource the system evaluates the angle between each couple of vectors as follows: $cos\theta = \frac{\overrightarrow{r_i} \cdot \overrightarrow{r_j}}{|\overrightarrow{r_i}||\overrightarrow{r_j}|}$. Once we fixed a threshold angle, the system suggests to the user all the resource classes whose "distance" is less than the threshold. Let notice that a single interaction with the system will lead to have an ideal resource for the user model that overlaps with the selected resource. This means that, after a single interaction, the distance between these two vectors is equal to zero (i.e., smaller than the threshold). However, the system does not have to start his suggestion after only one interaction.

In order to evaluate the minimum number of interactions needed before starting the recommendation process and the value of the threshold we performed a testing process. We conducted four set of tests. During the tests, the user was requested to browse the web site with a specific information goal (see Fig.2, tests 1.1, 1.3 and 1.4). In particular test 1.1 asked to find information about two different and not related given topics (for example about frescos and technological installations). Test 1.3 asked to find information about two given related topics (for example doors and windows) and test 1.4 asked to find information about a particular given topic (for example frescos) representing a class of resources. Another test was conducted with a random interaction (see Fig.2 test 1.2). From the results obtained we fixed a threshold for the distance and a minimum number of interactions required for the recommendation process.

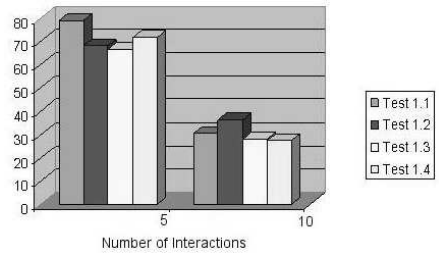


Figure 2: Value of the angle θ between the user model and the resource to recommend.

Finally we have to consider the case of two or more resource classes that have a distance less than the threshold. Let us recall that the resources are structured as a tree. If the selected resources have the same resource father in the tree (for example both the classes "floors" and "balconies" has the same father "finishing elements", see Fig.1), the system suggests the father, otherwise the system chooses the resource with a smaller distance.

3.1 Thematic Tours

The creation of a path through a hypermedia needs to solve two problems: to decide which information is interesting and to determine the modalities of visualization of the web pages. Concerning to the first problem, the representation by classes allows us to understand if and how much a user is interested to a particular class, and to select, in this way, a set of resources to suggest. To solve the second problem one has to characterize each single instance to decide the order in which the resources have to be suggested. In order to have a flexible system that does not depends upon the help of experts for adding new resource instances, we decided to have a characterization of resources at class level. Therefore, all the resources of a class are equivalent for the system. The only thing that characterizes a resource instance is its topographical information. In this way the choice for the recommendation does not depend only from the interests of the users, but also on the relative locations of resources in the virtual space. Moreover, even if we were able to have additional information on the single instances, ordering the presentation of resources according to their physical locations it is fundamental for the orientation of the user and for preparing the user to the real visit.

The topographical information defines, for all contiguous buildings, their respective geographic positions, such as north, south, east and west. The resources are combined to form a graph. At each edge of the graph it is associated a distance in meters, that represents the cost to cover the edge, such as the distance from the entrance of one building to the entrance of the next one.

During the first phase of the interaction, the suggestions proposed by the systems refer to buildings that are "near" to the current position of the user in the virtual space (see

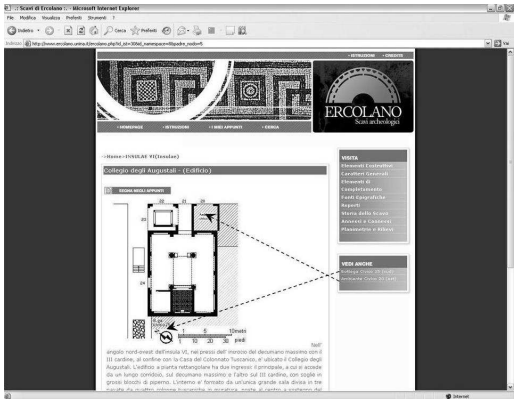


Figure 3: The interface of the Herculaneum Excavation website. The navigation bar is on the top right while the recommendation bar is the right–bottom one.

Fig.3). In fact, in this phase, the system does not have sufficient information on the user. Then, when the system has enough information on the user, it assists the user during the navigation, creating dynamically a tour of the buildings that have the properties the user is looking for. For example, the system may create a tour covering all the buildings that have frescos. This tour is created adding a list of links in the recommendation bar. This list contains links to the buildings that are interesting for the user and with which it is possible to define and to construct a thematic tour of buildings. The user is guided with links to web pages indicating the direction to follow, starting from his actual position in the virtual space. Moreover, the list of the links will be presented adding topographical information to get to all the interesting buildings. Examples of such information will be "turn on the right", "walk down", etc (like as the user is walking in the space). While planning the tour, buildings on the path may be classified as buildings of interest, and therefore they should be suggested, and buildings of no interest. If a building is of interest, the user is directly able to click on the relative link (the specific resource within the building, for example the frescos within the building) and he will be guided "to continue" the tour. On the contrary, if the building is not of interest, there is not a corresponding link. We decided to add a reference also to those building, while describing the path, in order to give more precise indications on the path to follow. In fact, to go from one interesting building to another, the user has to go over the not interesting ones. In the list of recommendations the system adds information like, for example, "come though the Building One". That may constitute an help for the user, in fact, the suggested path directly introduces the link to the appealing buildings, separating them from those buildings of not interest that are only a mandatory passage in the path. Finally, let us recall that the thematic tour is presented to the user in an apposite area of the interface and does not directly constrain the normal browsing activity of the user.

4. DISCUSSION

In this paper we presented a first approach aimed at an integration of topographical information within an informative web site. Moreover, the Herculaneum excavation hypermedia is able to implicitly create user profiles without requesting any direct information to the users. Starting from this profiling activity and the topographical information about the corresponding real site of the excavation, the system is able to suggest to the users personalized tours of the contents of the hypermedia. Finally, the suggestions about the links to follow or the places to visit are given to the user as he is walking in the real space. In our opinion this approach will improve the involvement of the user during the navigation, and the recall of physical locations during the real visit. Finally, as future work, we will extend our portal in order to be easily browsed also using PDA and we will integrate GPS information. In this way, a common web site can be easily used both from home, to search information, and during the visit to the excavation as a personal mobile guide. Moreover, we will extend the level of details of topographical information in order to deal with objects within the building area.

5. REFERENCES

[1] A. Kobsa. User modeling: Recent work, prospects and hazards. In *Adaptive User Interfaces: Principles and Practice*, pages 111–128. North-Holland, 1993.

[2] A. Dieberger. Providing spatial navigation for the world wide web. In A. U. Frank and W. Kuhn, editors, *Spatial Information Theory - A Theoretical Basis for GIS (COSIT'95)*, pages 93–106. Springer, 1995.

[3] Y. Hijikata. Implicit user profiling for on demand relevance feedback. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces*, pages 198–205, New York, NY, USA, 2004. ACM.

[4] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowl. Discovery*, 5(1/2):115–153, 2001.

[5] L. Terveen and W. Hill. Beyond recommender systems: Helping people help each other. In J. Carroll, editor, *HCI in the New Millennium*. Addison Wesley, 2001.

[6] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 675–684, New York, NY, USA, 2004. ACM.

[7] E. Not, D. Petrelli, O. Stock, C. Strapparava, and M. Zancanaro. Person-oriented guided visits in a physical museum. In *ICHIM*, pages 69–79, 1997.

[8] A. Dieberger and A. U. Frank. A city metaphor to support navigation in complex information spaces. *Journal of Visual Languages and Computing*, 9(6):597–622, 1998.

[9] E. Burattini, F. Gaudino, and L. Serino. Hypermedia knowledge acquisition and a bdi agent for navigation assistance. a case study: Herculaneum excavations. In *Europ. Conf. on Cognitive Science*, pages 437–440, 1999.

[10] S. Rossi, V. Scognamiglio, and E. Burattini. Web contents and structural adaptivity by knowledge tree: The herculaneum excavation hypermedia. In *Proc. of the third Inter. Conf. on Web Information Systems and Technologies WEBIST 07*, pages 270–275, 2007.

[11] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. *The Adaptive Web*, pages 54–89, 2007.

Visual Tag Authoring: picture extraction via localized, collaborative tagging

Andrea Bellucci, Stefano Levialdi Ghiron
Computer Science Department
Universitá Sapienza di Roma
Via Salaria 113.
Roma, Italy
{bellucci, levialdi}@di.uniroma1.it

Ignacio Aedo, Alessio Malizia
DEI Laboratory
Computer Science Department
Universidad Carlos III de Madrid
Avda. de la Universidad, 30.
28911-Leganés, Madrid, (Spain)
{alessio.malizia,
ignacio.aedo}@uc3m.es

ABSTRACT

In this work we present a system to encode location based information extracted from a media collection (the Flickr tagging system) into a single 2D physical label. This information is clustered by using locations metadata (*geotags*) and key-words (*tags*) associated to pictures. Our system helps two types of users: the user authoring the physical label and the final user who retrieves up-to-date information scanning the label with his/her camera phone. Preliminary results for a given seed word (the tag *Napoli*) on 3000 photographs are presented, together with some ad-hoc weighting factors that help in finding significant pictures (representing places) that can be associated to a specific area.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous

Keywords

collaborative tagging, clustering, geo-referenced photographs

1. INTRODUCTION

As described in [5] the introduction of new internet-oriented applications like blogs, wikis, newsfeeds, social networks, and bookmarking tools, has made the user-web-user interaction a natural activity for a wide number of users. In fact, collaborative tagging is performed via the web, and tags are both easy to assign and to generate from popular knowledge. In many instances of everyday life, where we may choose between the possibility of consulting a technical guide (for instance a tourist guide for travelers) or, differently, speak with people to obtain advice (talk to tourists that have been at the site of interest), we generally find that this second strategy is the most fruitful one; in most cases, it will also be more timely. This is to some extent exactly the strategy employed when using a social network (relying on others opinion over a content). Examples of the acquired benefits of socializing knowledge on the web are: Del.icio.us[1], a service for tagging, storing and sharing web bookmarks and Wikipedia[2] which is a free encyclopedia with user-generated content, produced by voluntary contributors. The system described here collects information (photograph tags) from Flickr[3](a system giving, to users, the possibility to publish and tag photographs they own); such tags are processed to obtain clusters by using the k-mean algorithm [3] so prompting the user with a list of photo thumbnails (and relative tags) that are *closest* (both spatially and ideally semantically) to a given tag (provided by the user). The experiments we conducted started from the tag *Napoli*, took into account 3000 pictures (present in Flickr and associated with this tag) producing 9 clusters.

AVI '08, 28-30 May , 2008, Napoli, Italy.

2. RELATED WORK

Retrieving information in a collaborative tagging system has been studied in various papers; here we are interested in the approaches described in [6] and [7]. In [6] authors present an algorithm (*FolkRank*) providing suitable ranking mechanisms, similar to those used by web search engines, but taking into account the structure of relationship in folksonomies (the combination of *folk* and *taxonomy* is generally used as a synonimous for tagging systems). While in [7] authors discuss how to improve search and exploration in a collaborative tagging system by means of tags clustering. Furthermore in [8], researchers report on how to automate information extraction from a large user-contributed photo collection. In particular, they apply geographic clustering to identify representative tags for a chosen area. They also show how computer vision algorithms can be successfully employed to increase the precision of retrieved photos. As stated in [4] location information (such as geographic coordinates), associated to content, can help in automatically understanding photo's semantics and can be easily acquired, indexed and searched. The use of location-based metadata has also been studied in [9] where the presented system allows: *a)* to share tagged images (photographs and related

[1]http://del.icio.us
[2]http://en.wikipedia.org
[3]http://www.flickr.com

tags) and *b)* to collect tag patterns present in geo-tagged photos.
In our previous work [10] we introduced the prototype of a semiautomatic system generating visual tags by gathering information from collaborative tagging. The user authoring the visual tag interacted with a list of given tags (key-words obtained employing Flickr's clustering service) and selected tags which best capture his/her needs. A preview of retrieved photos guided the authoring user in refining his/her selections. Once selected, these tags can be easly encoded into a visual tag, a 2D physical label like the Japanese QR Codes [1], which offers the possibility to associate objects of the real world to digital information.

3. PICTURES, TAGS AND PHYSICAL LABELS

Differently from our previous work, we present here the advantages of the correlation between pictures togheter with their tags and the spatial location information automatically acquired by a positioning system. By using geographic clustering techniques, we aim at identifying a set of tags (and relative photos) which are representative for a given area (identified by a *seed tag*). As *seed tag* we mean to search for images starting from a given tag or area. As we suggested in [10], these tags can be further encoded into a single visual tag. Barcodes are an example of visual codes: machine readable labels easy to generate and print. Differently, QR Codes represent the next generation of physical labels. QR Codes storing addresses and URLs may appear in magazines, on signs, buses, or simply any object which could encode information retained useful (see Figure 1). A final user having a PDA or a mobile phone equipped with a camera can easly scan and decode this label so to display the retrieved information. By employing tags of a collaborative tagging system as information encoded into a visual tag we can quickly provide complementary information when and where a user actually needs it.

Figure 1: Example of a visual tag placed on a building.

3.1 Formal definitions

In order to describe our approach we, firstly, introduce formal definitions on the dataset used in our experiments. The dataset consists of photos and associated metadata extracted from Flickr, by means of the Flickr API[4] service.
We define $\mathbb{P} = \{\, p \mid p \text{ is the tuple } (\alpha, o_p, \mathrm{G}_p, \mathbb{T}_p) \,\}$ as the set of all photos in the Flickr database, $\mathbb{T} = \{\, t \mid t \text{ is a tag} \,\}$ as the set of all tags and $\mathbb{U} = \{\, u \mid u \text{ is a user} \,\}$ as the set of all users. Each $p \in \mathbb{P}$ is represented by the tuple $(\alpha, o_p, \mathrm{G}_p, \mathbb{T}_p)$ where α is an unique ID number for the given photo, $o_p \in \mathbb{U}$ is the photo owner (identified by an unique ID number), $\mathrm{G}_p = (\ell_1, \ell_2)$ is the couple of geo-referenced information about the photo (latitude and longitude) and $\mathbb{T}_p \subseteq \mathbb{T}$ is the subset of associated tags chosen by the owner. For a given *seed tag* t we identify $\mathbb{P}(t) = \{\, p \in \mathbb{P} \mid t \in \mathbb{T}_p \,\}$ as the set of all photos tagged with t. For any subset $\mathbb{S} \subseteq \mathbb{P}$ we define $\mathbb{S}(t) = \{\mathbb{S} \cap \mathbb{P}(t)\}$ as the subset of all photos in $\mathbb{S}$ which are tagged with t. We denote with $\mathbb{U}(\mathbb{S}) = \{o_p \in \mathbb{U} | p \in \mathbb{S}\}$, $\mathbb{U}(\mathbb{S}) \subseteq \mathbb{U}$ the subset of the users in $\mathbb{S}$.

[4]http://www.flickr.com/services/api

3.2 Clustering

Differently from [10], instead of employing the Flickr clustering service we developed our own clustering application. Starting from a given tag (i.e. the seed word *Napoli*) we grouped all the $p \in \mathbb{P}(Napoli)$ employing the k-means clustering technique (see Figure 2).

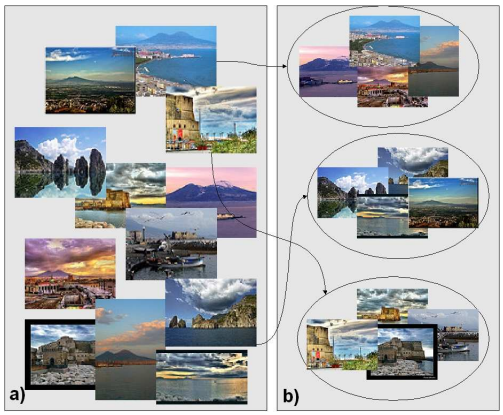


Figure 2: Example of images grouped into three different clusters.

In our system, the metric used to determine cluster membership is the geographic distance between photographed places, calculated by the geo-referenced metadata in each G_p. The k-means [3] is one of the most popular clustering algorithms due to ease of use and implementation. However, its randomness in choosing initial points, makes it hard to obtain reliable results without too many iterations over the entire clustering process. Furthermore, the results strictly depend on the number of chosen clusters and, often, there is no way of knowing a priori how many clusters best fit with the data. We employed the following naive approach to solve this problem: we compared the results of multiple runs of the algorithm with different k groups and choose the best one according to the Schwarz Criterion (or Bayesan Information Criterion)[2]; where starting points were chosen taking k random samples from the dataset. We plan to investigate other strategies in choosing clusters' number, also taking into account the geographic features of the dataset. For example, it could be interesting to assign a threshold for the cluster diameter in order to capture different zooming levels in a given area. In fact, diameter size (the granularity) plays an important role in geographic clustering and is strictly related to the area defined by the seed tag: if the seed tag is the name of a nation (i.e. *Italy*) one can be interested in grouping data about cities, while if the seed tag is the name of a city (*Napoli* in our case) it can be reasonable

to demand data on blocks.

3.3 Ranking

After performing the clustering phase, the system assigns a score to tags in each cluster $\mathbb{C}_i(Napoli)$ (we call it $\mathbb{C}_i$) in order to determine representative tags. Tags are scored using a $TF \times IDF$ (Term Frequency $\times$ Inverse Document Frequency) value as in [8], but with different meanings for terms in the formula. $TF \times IDF$ is a metric widely used in information retrieval for ranking documents: this measure evaluates how important a word is with respect to a document in a collection or corpus. In our case we used a variation of the $TF \times IDF$ weighting factor to taking into account the different features of our dataset (tags) compared to the traditional information retrieval ones (documents).
In order to compute the tag's score we defined the *Tag Frequency* $TF_{\mathbb{C}_i}(t) = \frac{|\mathbb{C}_i(t)|}{\sum_{p \in \mathbb{C}_i} |\mathbb{T}_p|}$ for a tag t, as the number of times t recurs in a cluster over the total number of tags in the same cluster. *Photo Frequency* $PF_{\mathbb{C}_i}(t) = \frac{|\mathbb{C}_i(t)|}{|\mathbb{C}_i|}$ is a measure of how important the tag is with respect to the seed word (*Napoli*), in terms of photos tagged with t. The *Inverse Photo Frequency* $IPF_{\mathbb{C}_i}(t) = \log \frac{\sum_j |\mathbb{C}_j|}{\sum_j |\mathbb{C}_j(t)|}$, instead, indicates how significant the tag t is for this cluster by taking the logarithm of the quotient between the number of photos in all clusters and the number of photo in all clusters also tagged with t. Finally, we then included a fourth parameter, called *Tag Authorship* $TA_{\mathbb{C}_i}(t) = \frac{|\mathbb{U}(\mathbb{C}_i(t))|}{|\mathbb{U}(\mathbb{C}_i)|}$ to measure the importance of a tag inside the cluster relative to the number of different users.
Each tag score was assigned using the following formula: $weight(t, \mathbb{C}_i) = TF \times PF \times IPF \times TA$.

4. OUR SYSTEM

We distinguished two different classes of user: the author of the visual tag and the final user retrieving up-to-date multimedia contents by means of his/her camera-phone. In this work we focused on the visual editor to help the authoring user in codifying information onto a visual tag. After selecting a seed word (the Italian city of *Napoli* in our experiments) our *Visual Tag Authoring* system displays the n most representative tags for each detected cluster along with representative photos for each tag (see Figure 3). As mentioned above, the granularity greatly affects clustering results and thus the overall system performance (so, for instance, *church* can be further refined by using sub-clustering). The system chooses the n tags with the highest score, filtered by a variable threshold manually set, representing the *feature vector* for the cluster. In this way representative photos are selected from the whole Flickr database (not only from geo-referenced ones). The *representativeness* of a photo with regard to a given tag has been computed through the *"most interesting"* function as provided by Flickr API. The main idea behind our approach is to generate a set of images (selected by the *feature vector*) for automatically creating a visual tag describing a location of interest [10]. Through the system interface the authoring user can select different information about the cluster (such as tags in the *feature vector* along with their score, the number of distinct users and the number of photo shots) to be encoded in the visual tag (the QR Code). All this information can be employed (by the final user) to retrieve relevant photos for the selected area. We are currently exploring which information is best suited for the users' needs. For example, users might want to get simply the most representative images for a given tag, but also the most recent ones or the best rated shots as suggested by the Flickr's community.

5. EXPERIMENTAL RESULTS

In this section we present results of our experiments (for the purpose of this work we evaluated only preliminary results). We report on the performance evaluated using a set of about 3000 geo-referenced photos for the seed word *Napoli*. Photo shots, and related metadata, were collected from Flickr API, filtering the 3000 geo-referenced ones from a whole dataset of over 57000 pictures. We grouped photos in our dataset into 9 clusters (based on locations metadata) following the procedure described in 3.2. Two of these clusters have been discarded because they were related to places geographically far from our area of interest (i.e. Figure 4 refers to a photo of the Anaheim Angels' baseball player Mike *Napoli*, taken in Orange County, California).

Figure 4: The baseball player Mike *Napoli*.

We discarded other two clusters because their *popularity index* was below a fixed threshold (we estimated the *popularity index* of a cluster $\mathbb{C}$ as the sum of the ratios between the number of tags used by each user and the number of his/her photos in the cluster). As in [8], images' representativeness was judged by users. If the images contain a view of a location, that evaluators considered pertaining to the related tag, the photo is marked as a representative one, otherwise it could be marked as non-representative. In Figure 5 precision[5] for the top 5 images, retrieved for each tag, is shown. After performing the evaluation phase, we think

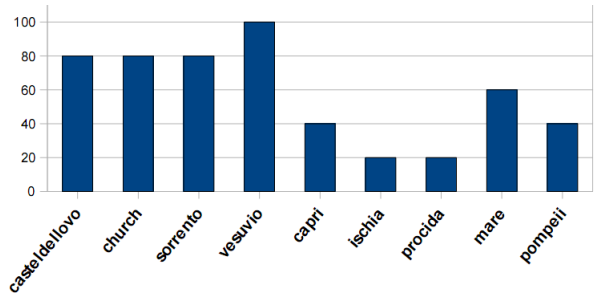


Figure 5: Precision for the 5 most representative images for each tag.

that useful features included in our system are: *a)* the importance of localization information combined with image

[5]In information retrieval, *precision* is the percentage of documents returned that are relevant to a query.

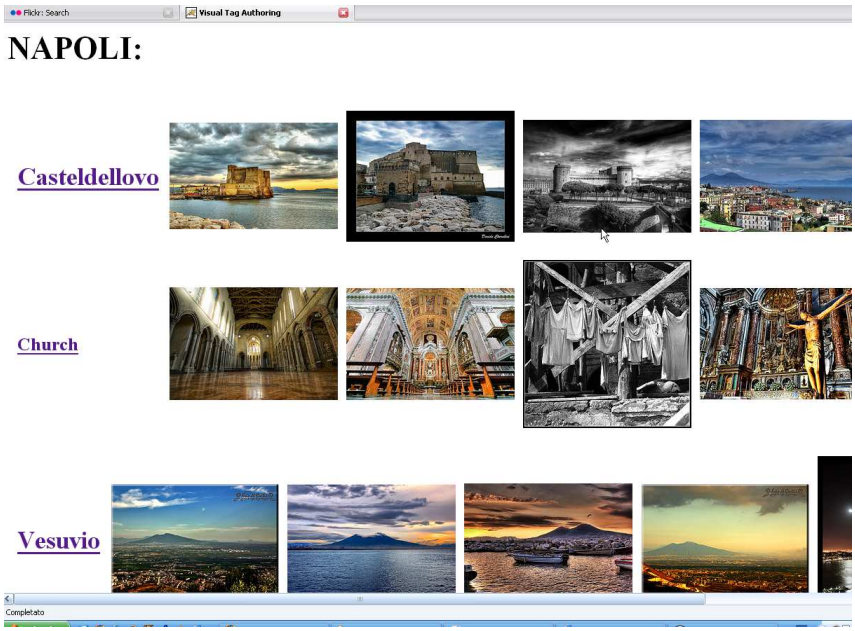


Figure 3: A snapshot of the system's interface presenting a partial view of the resulting clusters.

tags in retrieving relevant photos, *b)* the usefulness of having a preview in the system, using picture thumbnails and *c)* obtained pictures are up-to-date (have been recently taken).

6. CONCLUSIONS AND FUTURE WORK

In this work we have shown how to retrieve up-to-date visual information in the physical world, starting from community-shared geo-tagged photographs. We have, firstly, demonstrated how geographic clustering can be successfully used to extract a vector of representative tags and relevant photos for an area of interest. These tags can be further encoded in physical 2D labels (like QR codes [1]) and employed to quickly retrieve multimedia information from the web. In this way we have established a link between online shared contents (through the collaborative tagging system) and the final user (equipped with mobile devices). We plan to report on experiments resulting from populated datasets (i.e. for the seed tag *Rome* we have about 800000 tagged photos, of which about 200000 geographically located). In the future we aim to employ different collaborative tagging systems (such as Youtube[6]) in order to extend the retrieval process to different multimedia content. At the same time we will focus our efforts on exploring different content retrieving and displaying approaches for users carrying mobile devices.

7. REFERENCES

[1] L. E. Holmquist. Tagging the world. *interactions*, 13(4):51–ff, 2006.

[2] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[3] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1978.

[4] K. Toyama, R. Logan, and A. Roseway. Geographic location tags on digital images. In L. A. Rowe, H. M. Vin, T. Plagemann, P. J. Shenoy, and J. R. Smith, editors, *ACM Multimedia*, pages 156–166. ACM, 2003.

[5] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools (i): A general review. *D-Lib Magazine*, 11(4), 2005.

[6] A. Hotho, R. Juschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, 2006. Springer.

[7] F. S. Grigory Begelman, Philipp Keller. Automated tag clustering:improving search and exploration in the tag space. In *WWW06: Proceedings of the 2006 International Conference on the World Wide Web*, 2006.

[8] L. Kennedy, M. Naaman, S. Ahern, R. Nair, and T. Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 631–640, New York, NY, USA, 2007. ACM.

[9] M. Naaman, A. Paepcke, and H. Garcia-Molina. From Where to What: Metadata Sharing for Digital Photographs with Geographic Coordinates. In *10th International Conference on Cooperative Information Systems (COOPIS)*, 2003.

[10] A. Bellucci, S. Levialdi, and A. Malizia. Visual tagging through social collaboration: A concept paper. In *INTERACT (2)*, volume 4663 of *Lecture Notes in Computer Science*, pages 268–271. Springer, 2007.

[6]http://www.youtube.com

Time2Hide: Spatial Searches and Clutter Alleviation for the Desktop

George Lepouras, Aggelos Papatriantafyllou
Dept. of Computer Science and Technology,
University of Peloponnese,
Tripolis, 22100, Greece
+302710372201
{G.Lepouras, pag36}@uop.gr

Akrivi Katifori
Dept. of Informatics and Telecommunications,
University of Athens
Panepistimioupolis, Ilissia, 157 84
Athens, Greece
+3021071275241
vivi@di.uoa.gr

Alan Dix
Computing Department, Lancaster University
LA1 4YR, Lancaster, UK
a.dix@comp.lancs.ac.uk

ABSTRACT
With information abundance the user's desktop is often cluttered with files and folders. Existing tools partially address the clutter problem. Time2Hide enhances desktop functionality by allowing icons that are not used for a long time to gradually fade and merge with the background. This aims to alleviate the problem of icon clutter. Users can also perform spatial searches, defining areas of the desktop they wish to search for icons; can reveal one or more hidden files or can go back in time animating the desktop and its changes. With Time2Hide users can still use the desktop as a place for storing files and folders, without worrying about the possible clutter and without being afraid that the files might be moved to an unknown location. The new desktop has been implemented and evaluated. Evaluation results reveal that such an enhanced desktop can significantly support users and propose suggestions for further improvements.

Categories and Subject Descriptors
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous

General Terms
Design, Experimentation, Human Factors.

Keywords
Desktop tool, icon clutter, spatial search, personal information management.

1. INTRODUCTION
The desktop metaphor has now been around for more than a quarter of a century. It first appeared in Xerox Star [6] and it is currently found with small variations in the core of most interaction environments. The desktop metaphor was aiming to create a working environment that would resemble that of the user's desk. Document files would be piled on the desktop as real world documents would be piled on the desk. The user would be able to click on a file and launch the associated application to edit the file. While this was the original notion, users have developed their own interaction routines and methods that take advantage of the desktop metaphor, sometimes in peculiar ways.

Recent studies have investigated the users' habits in regard to the desktop usage and the personal information management routines and several interesting results have been noted [9]. The studies reveal that the desktop is often employed for the following:

Quick access. The desktop is the first thing visible to the users when they switch on their computer and, as a result, by many it is used as a means to gain quick access to programs and files.

Reminding. Sometimes files are placed on the desktop by the users not only for quick access but also to remind them to process them or to do a particular task [2] [3] [7] [8] [10].

Temporary Storage. The desktop is sometimes also used as a temporary storage area. It has been noted as a common practice to first place on the desktop files coming from outside sources and later file them to their proper location [8]. Again, in this case items tend to be forgotten and leading to disarray.

The use of spatial arrangements on the desktop has also been recorded in [8] and [11]. Another issue noted in some of the users is that they seemed to consciously or subconsciously ignore certain files, folders or programs on their desktop [8]. When their attention was brought to this, users said they got used to them and started considering them part of the background and, consequently, ignoring them completely.

Furthermore, as some of the users explained [8], after leaving an icon at a specific position for some time they are reluctant to move it because they are afraid they won't be able to retrieve it. This fact implies the existence of a spatial memory as to the function of specific items in specific positions. Users that have realized its importance sometimes prefer a less tidy desktop and folder hierarchy to filing away files that they got used to in a specific place and then forgetting where they are.

Clutter seems to be "visually distracting" and "dizzying" for most of the users as noted in [8] and [11] and it is the most common reason why they re-arrange and tidy – up their desktop.

AVI'08, 28-30 May , 2008, Napoli, Italy

However, maintaining their desktop well arranged is a task few users have the luxury of time to perform regularly. As a result, clutter may become a source of distress for them. To this end, we have designed and implemented an enhanced version of a desktop that tackles the problems of clutter and supports spatial and temporal searches.

2. PREVIOUS WORK

In [5] the authors introduce a novel desktop metaphor called Lifestreams, which functions as a diary of all the documents a user creates or receives. The user can also store documents to be used in the future such as calendar items, reminders and to-do lists. A basic limitation of that system was the one-dimensional arrangement of documents.

Dourish et al. [4] present a prototype document management system named Presto, which provides rich interaction with documents. The system allows grouping of documents in collections based on their attributes. To interact with document spaces the authors have implemented Vista, a generic browser for Presto document spaces. Vista looks like a traditional PC desktop and it gives a generalized view of the document space, supporting organizational tasks and launching other applications. The browser could be employed as a tool for reducing icon clutter on the desktop, however this could demand extra effort from the user's end.

In [12] Rekimoto proposed a time centric desktop where the user can travel in time to visit past and future information spaces. The described desktop supports browsing in time, offering a variety of visualization techniques to cater for different user needs.

The latest edition of Mac OS (Mac OS X 10.5 "Leopard") was made available at the end of October 2007. It includes -among others- a tool called Time Machine [1]. Time machine can be set to archive and keep snapshots of all files on the user's computer. The tool is mainly a back up utility which apart from storing the files it also allows moving back in time to view changes. To this end, the tool can be used to support personal information management, but it cannot help reduce desktop clutter.

Microsoft Windows XP on the other hand include a tool that aims to help reduce the icon clutter on the desktop. Desktop cleaner can be set to run every 60 days to move unused icons to a pre-determined folder. This approach certainly relieves the desktop clutter problem. The negative side-effect is that the user cannot customize the time period or the location for storing unused icons. Worse than that, the desktop cleaner by moving icons to another location rearranges the desktop, a feature which studies [8] have noted is not welcomed by most users.

3. DESKTOP DESIGN

We have created the Time2Hide desktop, which supports users in their personal file management with two major design goals:

1. Free space on the desktop and reduce clutter without rearranging the desktop, and

2. Aid both temporal and spatial searches on the desktop.

These goals were derived from the previously mentioned studies on desktop usage and personal information management. While users dislike clutter they are often afraid to move or archive a file in case they forget its new location or simply they do not have the time to re-arrange a cluttered desktop. On the other hand users frequently tend to spatially arrange files on their desktop and a tool, which would support spatial searches, could prove to be useful for them.

The metaphor for Time2Hide is based on the finding [8] that users tend to disregard icons they see on a daily basis if they do not use them. To this end, icons in Time2Hide start to gradually fade and hide in the desktop. This leaves the desktop clear from any icons the user has not accessed for some time. However, no icons are deleted or moved to another location. All icons remain on the desktop albeit some hidden. Also, all icons (file, folder, application, etc.) are treated uniformly. The user can hide them immediately, let them disappear or make them always visible.

Time2Hide desktop is a proposal for a new desktop which offers a number of new functions to the users, while resembling the original Windows desktop. Users can customize most aspects of the desktop and its ability to hide icons. Users can:

- Define a global time period in days during which icons will gradually fade. To avoid situations where the user switches on the computer after a long period only to find all the icons missing, days correspond to working time, i.e. time the user is logged in.
- Define the disappearance time for specific icons. The user can also define that the icon will always be visible. All settings made specifically for certain icons override the general settings.
- Define an initial time before the icons start to fade.

Figure 1. Area selection

Once the preconfigured time elapses, icons are hidden and the desktop area is cleared. To restore hidden icons on the desktop the user may select one of the following functions:

- Right-click and drag to select an area on the desktop. While dragging, the cursor changes from arrow to shovel to denote the 'digging up' function and a tip appears at the lower, right end of the selection box displaying the number of icons selected, the number of hidden icons and the number of deleted or moved to a different location icons (Figure 1).

  Once the mouse button is released a popup menu appears with the option to list all files in the selected area. If the user selects the "*List Files*" option a new window appears, listing all files along with information concerning days left before the icon disappears, current

status (visible, hidden, or deleted), creation data, last accessed date and deleted date if applicable (Figure 2).

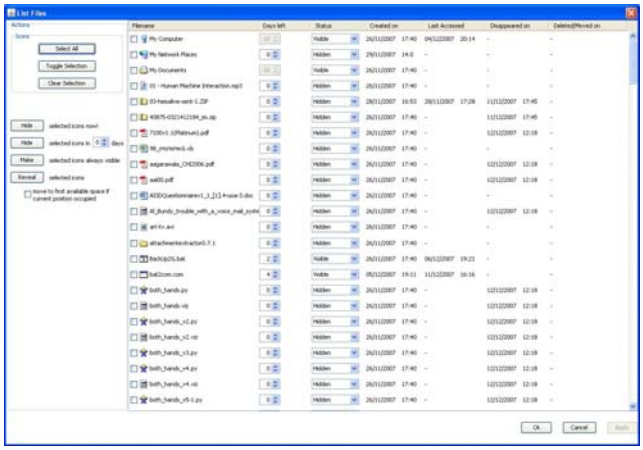


Figure 2. List Files dialog box

Through this window the user can select one or more icons to perform actions such as "*hide*" or "*reveal*" immediately, set the selected as "*always visible*" or change the current number of days before the icons disappear.

- If the user right clicks on any clear area of the desktop the "*List Files*" option displays all files. Through the popup menu that appears the user can also select to "*Search*", "*Go Back in Time*" and "*Customize*".
- If the user does not r*emember an*y of the data needed to perform a *search* she can elect to "*Go Back in Time*" to reveal a hidden icon. Depending on the customization option the user can either use animation or directly go back to specific dates. When the user finds the file she can right click on it and reveal it.

Figure 3. Back in Time animated

Once a hidden icon is revealed it may be the case that a new icon has taken its place. In such a case the two icons are placed *on top of each* other with a slight displacement. The user can also customize the desktop to initially reveal the icon in its former location and if the space is occupied to animate and move it to the first (starting from the top left corner of the screen) location. The animation helps the user notice the change in the icon's location.

If the animation option is chosen the user can also set the animation granularity: days, hours or minutes. By right-clicking on a clear area of the desktop and selecting the "*Go Back in* Time" option a square appears in the middle of the screen displaying the date the snapshot of the screen corresponds to. If the user has chosen hours or minutes as the animation granularity, apart from the date the box also displays the exact time period (e.g. 17:00-17:59) (Figure 3).

As illustrated in the figure, icons that have been deleted or moved in the meantime, are shown with a special icon (red X) to denote they are unavailable. By rotating the wheel on the mouse the user can go back or forth in time. The user can also choose to move to a particular date directly from a calendar and be presented with a snapshot of that day's desktop.

4. IMPLEMENTATION

Time2Hide desktop has been implemented in Java to enhance portability and it employs a MySQL database to keep a log of every change on the desktop. The current version of the system emulates the Microsoft Windows XP desktop, but future versions will be able to emulate Mac OS X style of desktop. The system runs as a full screen application covering the existing Windows desktop.

When it is executed for the first time Time2Hide scans the user's desktop folder and populates the emulated desktop with all the icons (files, shortcuts and folders) it finds there. After that, Time2Hide keeps a log of all changes that happen on the emulated desktop and at the same time monitors the user's desktop folder. This is necessary because some applications (such as web browsers) tend to save downloaded files on the desktop and this action has to be reflected on the emulated desktop as well.

5. EVALUATION

Time2Hide was evaluated over a four weeks' period. To this end, eight users (four females and four males) were selected with more than three years experience with computers. All of them used their desktop in their everyday interaction either for quick access to files, for reminding purposes or for temporary storage. To monitor the users' interaction with Time2Hide desktop a questionnaire was compiled consisting of four major parts: user's profile, current desktop status, user's view of hiding icons functionality and user's view of spatial searches. The questionnaire contained forty-nine questions in total, most of which were closed Likert scale questions and the rest open questions. User's profile was completed once, while the other three parts were being completed at the end of each week. This scheme provided a better view over time of the subject's interaction and opinion regarding the desktop under evaluation.

5.1 Summary of Evaluation Results

The application was successful in its first aim to alleviate icon clutter on the desktop. Even from the first week some users noted a difference while the majority of users felt that the icon clutter was reduced after four weeks of usage. But icon fading proved to be a stress factor for some users. Three of the users admitted that there was at least one point in time they were afraid they would loose information/icons residing on their desktop. However, after four weeks of everyday use all felt very confident they could find hidden files, except for one user who felt moderately confident.

During the same period all of the users uncovered at least one icon, while five out of the eight deliberately hid icons on the

desktop. Of the eight users during the evaluation period two never performed a search or list files action on a specific sub-area of the desktop. The response of the rest at this type of search function was positive on its usefulness and very positive on its ease of use.

All but one user performed at least once a "*Go back in Time*" action using the option for animation. The users' feedback was overall very positive on the usefulness of the function with negative comments on ease of use stemming from the absence of an alternative to the wheel mouse method. This problem was apparent with laptop users who had to revert to an external mouse instead of the touchpad.

Nevertheless, the overall reaction to Time2Hide desktop was very positive as already stated. Users in general found the desktop easy to use and not at all frustrating.

While there was not a direct question on rating the usefulness of each functionality, users' response on listing the most negative and the most positive aspects of the new desktop indicates that the majority of the users expressed a strong positive opinion on the hiding functionality failing to express any comments (negative or positive) on the search functionality. However, this could also be attributed to the fact that users did not use the search function very often.

6. FUTURE WORK

A number of enhancements is foreseen for future versions: The first is to allow the user to move to the future and make changes to future snapshots of the desktop. This will offer the possibility of using the desktop as a reminder.

The method for traveling back and forth in time will also change. Based on the users' comments a slider will be added allowing the user to change the desktop in a user-defined time frame.

Another enhancement will be a function for naming user-defined areas on the desktop. This will help users to better organize their desktop. The user will be able to define an area and set rules for the icons on that area. For example, the user will be able to define areas for certain types of documents and to automatically arrange the icons on different clusters on the desktop.

Apart from these enhancements we would like to try different metaphors for icon hiding. For example, instead of the icon becoming more transparent as time passes it could shrink to a small size without fully disappearing. This may help users that feel anxious when icons hide from their view.

7. CONCLUSIONS

In this paper we have presented Time2Hide desktop, which aims to enhance the common desktop with new functionality. Results from the user evaluation are encouraging. While we do not claim that all users will welcome such a desktop we feel that Time2Hide offers new capabilities that can assist a large percentage of users in their everyday interaction.

We also feel that once we address the issues found, we should continue the evaluation on a longer term and with a larger user sample as there exist some questions that have to be clarified. The main question is what will happen after a few months of usage when there will be a great amount of icons hidden in the desktop.

So far, two users reported that once some icons were on the verge of becoming invisible they started to clean up, deleting unwanted icons and putting the rest in folders. On the other hand two other users started almost immediately to hide icons they "*did not know what to do with them*" with a mental note to manage them later. Yet, the rest of the users did not do anything, they just left the icons disappear. We hope that a long-term evaluation will reveal how will users interact with Time2Hide and if it affects the way they perform tasks.

8. REFERENCES

[1] Apple Mac OS X 10.5 Leopard Time Machine. http://www.apple.com/macosx/leopard/features/timemachine.html

[2] Barreau, D., and Nardi, B. Finding and reminding: File organization from the desktop. SIGCHI Bull. 27, 3 (July 1995), 39–43.

[3] Boardman, R. and Sasse, Mt. A., "Stuff Goes into the Computer and Doesn't Come Out": A Cross-tool Study of Personal Information Management. CHI 2004, 24-29 April, Vienna, Austria

[4] Dourish, P., Edwards, W. K., LaMarca, A., and Salisbury, M. 1999. Presto: an experimental architecture for fluid interactive document spaces. ACM Trans. Comput.-Hum. Interact. 6, 2 (Jun. 1999), 133-161.

[5] Freeman E., and Gelernter, D., Lifestreams: A storage model for personal data, SIGMOD Record, volume 25, number 1, pages 80-86, 1996.

[6] Johnson, J., Roberts, T., Verplank, W., Smith, D.C., Irby, C., Beard, M., and Mackey, K. "The Xerox Star: A Retrospective," *IEEE Computer*, 22. (9) (1989), 1l-29.

[7] Kamaruddin, A., and Dix, A., Understanding Physicality on Desktop: Preliminary Results. Physicality 2006, 6-7 February 2006, Lancaster.

[8] Katifori, A., Lepouras, G., Dix, A., Kamaruddin, A., Evaluating the Significance of the Desktop Area in Everyday Computer Use, Proceedings of the First International Conference on Advances in Computer-Human Interaction, ACHI 2008, 10-15 February 2008, Martinique.

[9] Kaye, J. '., Vertesi, J., Avery, S., Dafoe, A., David, S., Onaga, L., Rosero, I., and Pinch, T. 2006. To have and to hold: exploring the personal archive. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006).

[10] Malone, T.W. (1983) How do People organize their desks? Implications for the design of office information systems. ACM Transactions on Office Information Systems 1(1):99-12.

[11] Ravasio,.P., Guttormsen,, S., and Krueger, H. In Pursuit of Desktop Evolution: User Problems and Practices With Modern Desktop Systems. ACM Transactions on Computer-Human Interaction, Vol. 11, No. 2, June 2004.

[12] Rekimoto, J., Time-machine computing: a time-centric approach for the information environment Symposium on User Interface Software and Technology, Proceedings of the 12th annual ACM symposium on User interface software and technology Pages: 45 - 54 Asheville, North Carolina, United States.

Users' quest for an optimized representation of a multi-device space

Dzmitry Aliakseyeu
Media Interaction Group
Philips Research Eindhoven
5656AA Eindhoven, the Netherlands
dzmitry.aliakseyeu@philips.com

Andrés Lucero
Department of Industrial Design
Eindhoven University of Technology
Den Dolech 2, 5600MB Eindhoven, the Netherlands
a.a.lucero@tue.nl

Jean-Bernard Martens
Department of Industrial Design
Eindhoven University of Technology
Den Dolech 2, 5600MB Eindhoven, the Netherlands
j.b.o.s.martens@tue.nl

ABSTRACT
A plethora of reaching techniques, intended for moving objects between locations distant to the user, have recently been proposed and tested. One of the most promising techniques is the Radar View. Up till now, the focus has been mostly on how a user can interact efficiently with a given radar map, not on how these maps are created and maintained. It is for instance unclear whether or not users would appreciate the possibility of adapting such radar maps to particular tasks and personal preferences. In this paper we address this question by means of a prolonged user study with the Sketch Radar prototype. The study demonstrates that users do indeed modify the default maps in order to improve interactions for particular tasks. It also provides insights into how and why the default physical map is modified.

Categories and Subject Descriptors
H.5.m [Information Interfaces and Presentation (e.g., HCI)]: Miscellaneous.

General Terms
Design, Human Factors, Performance.

Keywords
Interaction techniques, map, spatial, reaching, large-display systems, multi-display systems.

1. INTRODUCTION
Thanks to the rapidly reducing cost of display and network technologies, situations in which many different devices with heterogeneous display sizes interact together are becoming commonplace. Often these environments present a mixture of personal devices such as Personal Digital Assistants (PDAs), tablet and laptop PCs, and shared devices such as large displays. In a device-cluttered space, such as the one shown in Figure 1 (left), the tasks of identifying a particular device and facilitating the transfer of objects from one device to another, also referred to as multi-device (display) reaching, becomes frequent. Therefore, alternative techniques for performing such interactions have lately received a fair share of attention.

A number of interaction techniques have been developed that aim at intuitive and efficient reaching between different devices. The recent study by Nacenta et al. [7] suggests that Radar View might be a very efficient technique for multi-device reaching. Map-based techniques such as Radar View [7] have the potential to support intuitive system identification and interaction without necessarily requiring physical proximity to the system they interact with (although they might profit from it). The success of map-based techniques relies on being able to associate a physical device with its representation on the map. In this paper, we report on a user study that explores whether or not users appreciate the possibility of adapting such radar maps to particular tasks and personal preferences. Or, in other words, if users are given freedom to modify the Radar View representation, will they strive to optimize this representation? If so, which criteria do they use to motivate changes?

The study was done using the Sketch Radar prototype [1]. With it, a user is able to control how and what information is presented on the map at any time. The users are free to adjust the map to make it fit better to a particular task or to their preference. We strived for a natural setting where people would be engaged in an activity over an extended period of time. We also wanted our participants to focus on the activity supported by the tool rather than on the interface with the tool itself. Therefore, we created a user study in the form of a game.

2. RELATED WORK
The Radar technique uses a reduced representation (a map) of the surrounding environment. When the pen touches an object, such as a file, the map appears. The user can place the object at a desired location by moving the pen to that target location. Radar View is hence similar to the World in Miniature [9], but in two dimensions. Users do not need to physically move to access a remote system, but the required precision of their actions increases when more devices need to be represented within a radar map of fixed size and resolution.

A recent study [7] has experimentally compared several multi-display reaching techniques. Radar View was found to be faster than the other techniques and was also subjectively preferred. The success of map-based techniques such as Radar View [7] relies on being able to associate a physical device with its representation on the map. Or in other words Radar Views support Stimulus-Response Compatibility (SRC). SRC was introduced in 1953 by Fitts et al [6]. It was shown that the speed and accuracy of

AVI'08, 28-30 May , 2008, Napoli, Italy

responding is dependent on how compatible stimuli and response are. Duncan [5] has studied spatial SRC and found that when spatially distributed stimuli (lights) and responses (buttons) have a compatible arrangement subjects were able to respond faster than when the arrangement was incompatible. However the effect of SRC is unclear when more complex tasks need to be solved. It was also shown that the spatial organization of displays allows efficient access to them, in the sense that it outperforms existing tree- or list-based approaches (such as File explorer or Favorites in Internet explorer) [8].

The only known example of a system that uses the radar metaphor and that addresses how physical devices can be arranged on a map is ARIS [3, 4]. ARIS uses an iconic map of a space as part of an interface for performing application relocation and input redirection. The differences with the Sketch Radar technique that was used in this experiment are the following. First, Sketch Radar aims at supporting a different task, i.e. placing and retrieving files, not relocating applications. Second, Sketch Radar is not limited to devices that have screens, but can include other devices such as printers. Third, because Sketch Radar does not necessarily rely on a physical layout, such as the devices in a single room, it allows combining distant devices in a single map. The nature of the tasks and spaces in ARIS implies that the flexibility in map layout offered by Sketch Radar is not required.

3. User study

A preliminary pilot study showed that some users adjust the default physical map when told that they will be required to repeat prescribed tasks. The goal of this study was to determine whether such behavior is also observed in a natural setting where people are engaged in an activity over an extended period of time. In order to make our users focus on the activity supported by the tool rather than on the interface or the tool itself, the setting for the user study consisted of a game.

Our main research question can be formulated as follows:
Given the freedom to modify the Radar View representation in real time, will users strive to optimize this representation? If so, which criteria will be used to motivate changes (nature of the task, prior knowledge of the environment, spatial location, etc.)?

Ultimately the study would also allow answering the second question:
How much and in which way does the nature of the task performed in a multi-device environment affect the map representation, if at all?

In order to improve validity of the study and reduce the effect of different playing strategies that participants might employ during the game, the study was divided into two parts. The first part consisted of several controlled sessions in which participants performed preset tasks. The second part was an unconstrained gaming situation.

3.1 Game and task description

Feeding Boris is a tamagochi-like game and was inspired by the Feeding Youshi game presented in [2]. The main goal of the game was to feed a virtual cat called Boris. Boris is continuously traveling between different computers to find a "safe" hiding place. Depending on the players' actions Boris becomes hungry or unhappy, which in turn determines his most likely hiding place.

The computers that play a part in the game are not directly

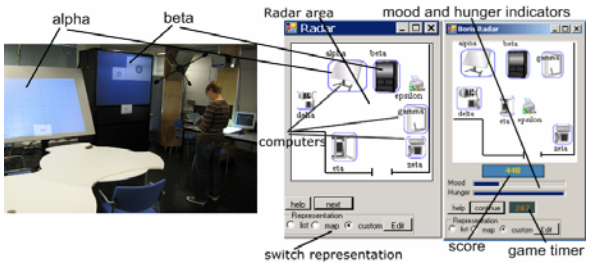


Figure 1. Environment used in the "Feeding the cat" experiment (left), Sketch Radar main window with the room plan (center) and Sketch Radar in game mode (right).

accessible, they only provide visual information (i.e., only the displayed output of the computers is available). For example the player can find out where the cat is by either exploring computers one-level-at-a-time through Boris Radar or by physically moving around to check the screens of the computers. However, to feed the cat the player needs to use Boris Radar.

A TabletPC with the Sketch Radar prototype (Figure 1) software was used to access and explore the different computers, to gather food and to feed Boris. Additionally, using the Sketch Radar editing capabilities users were able to control how and what information was presented on the map during the experiment.

In order to examine the effect of the specific task both Boris's movements and the meal locations were non-random. For example, Boris would only hide on 3 of the 10 computers, and specific kinds of food would only appear on specific computers. During the first part of the study participants were receiving different hints (for example "Boris usually hides on computers with large screens." or "Boris has found a new hiding place in computer Theta."

The test started in a single room which contained multiple devices that the participant needed to interact with: two PCs with their displays switched on (Zeta and Delta), one PC with the display switched off (Eta), one tabletop display (Gamma), one printer (Epsilon) and two wall displays (Alpha and Beta) (Figure 1). All devices were clearly labeled with their respective names. During the course of the study two new rooms were introduced, each room contained a single PC with a display (Theta and Kappa).

The experiment was conducted with 7 participants (2 females and 5 males) between the ages of 23 and 35. All participants had previous experience with graphical user interfaces, but not with Sketch Radar. The environment where the study took place was familiar to all participants. The participants were tested individually. The experiment consisted of three parts: tutorial, controlled sessions, and free form game.

In the first part participants performed multiple training tasks with the Sketch Radar application on the TabletPC, following a map building tutorial. The duration of this first part varied across participants from 30-60 minutes.

The second part lasted for three days and included one 20-40 minute session per day. On the first day participants received the TabletPC with a preloaded physical map of the first room. All systems were presented equally on the map (in terms of geometrical size) in a position that closely corresponded to their actual physical position within the room. The participants were also positioned inside the same room. Their task consisted of feeding the cat with specific food. During the experiment two

more rooms with one computer in each were introduced.

The third part of the experiment was the actual game. It also lasted for three days (with 15-30 minutes playing sessions every day). Users started from the maps and knowledge that they had acquired from the second part of the experiment. Users were free to choose where they wanted to be physically, but all of them chose to play the game from within the first room (which contained most of the systems). The goal of the game was to acquire as many points as possible by feeding Boris, in a given time. Participants were aware of the fact that the one who collected the maximum score would get a prize.

3.2 Results

The evaluation showed that users indeed changed the layout of the map to make it more suitable for the particular task that they needed to perform. Most of the participants (5/7) only adjusted the map before and after test sessions, but not during the session itself. By the end of the experiment all participants had created their own representation, only 2 participants used the preset physical map during the first part of the experiment, but changed it after the first game session. All other participants switched to their own representation after the first session of the first part.

There are some more specific observations that were made during the experiment:

1) Physical location provides strong external cues, while custom-made representations which are often based on internal cues that might be forgotten or changed, need repetitive usage to be remembered. Between sessions some participants (3/7) had forgotten about acquired patterns of cat and food behavior. Therefore their own representation created during a previous session did not make sense to them anymore, and even caused confusion.

2) In the post interview where participants were asked to describe computers that shared the same task-related property, the description usually relied on properties provided in the game hints (6), names (3), look (2) or/and location on the map (2). For example if the provided hint stated that "Boris is hiding on computers with large displays", the most common answer on the question: "Where does Boris usually hide?", would be "Large computers Alpha, Beta, and Gamma".

3) If to the known group of computers (for example "Large computers where Boris hides") a new computer is added ("This is a new computer Boris also can hide here"), even without giving it any specific properties, it will acquire the properties of the group. So first time it will be referred as a "new one", and after that it will usually be referred together with the rest of the group so "Large computers Alpha, Beta, Gamma, and Theta [new computer]". This new computer Theta that is actually physically small is placed in the group of "large computers" which no longer corresponds to the physical size but more to the fact that Boris can be found on them. Therefore "large computers" evolves from being a property of the computer to becoming a label. This was observed with 4 out of 7 participants.

5) When placed in a separate room, where participants could not see the screens of the devices, only one participant moved from a physical to a purely task-oriented map. Others commented that if from the beginning they would not be able to see devices and content of their screens it might be quite possible that they would adjust the map more drastically.

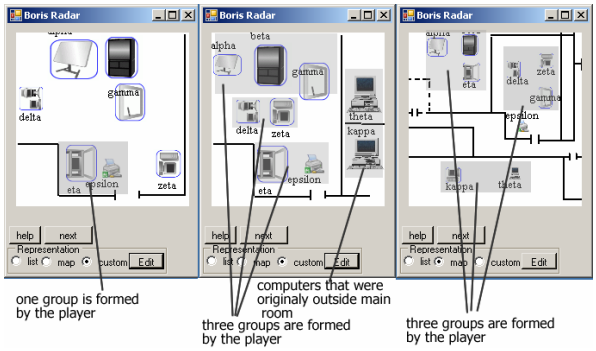


Figure 2. Different levels of modification. The map is moderately distorted, with one user-defined group (computers that have only one hiding place and one type of food) (left); the map is strongly distorted with fouruser-defined groups (center); the custom map is completely distorted (right).

6) Four common steps in the evolution of custom-made maps could be identified:

1. The physical maps are only slightly distorted. The icons that represent those devices are slightly resized and repositioned to make movements shorter. No specific grouping is made. (5/7)
2. The map is moderately distorted (Figure 2). Some grouping is made. For example, computers where food appears more often are grouped together. However participants try to maintain as much as possible a correspondence to physical location. (5/7)
3. The map is strongly distorted (Figure 2). Only the computers that have screens and that are located in the first room retain a position that correlates strongly with the actual physical location. Computers that do not have screens are positioned freely based on different properties. Computers that were originally outside of the first room were positioned freely, although still kept outside of the room boundaries. (6/7)
4. The map is completely distorted (Figure 2). Computers are grouped based on certain properties, no correspondence with physical location. However some order-based spatial relationships between computers are retained (such as this computer is to the left, right or in front of that computer). (4/7)

7) During the experiment, all devices with screens were constantly displaying information about their status. The same information was available through the SketchRadar, but in order to obtain this information, participants needed to go through several steps. We observed that during the game participants very often instead of exploring the device representation on the TabletPC were first checking the content of surrounding displays, locating the cat or needed type of food and only then accessed the food or cat through the TabletPC. They would only start to look for the cat through the TabletPC if it was not visible on any of the screens. We believe that is why most of the participants did change the map but also tried to partly keep some references to the physical location of devices.

The speed with which this transformation occurred varied between participants (Figure 3). Some participants skipped steps in between. Two participants immediately after the first session created custom-made representations that were moderately distorted. One participant moved back to the physical map used it for two consequent sessions and then jumped to the strongly distorted representation (Level 3).

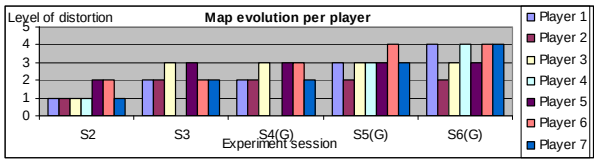


Figure 3. Level of map distortion on every session, for every player (during first session all players used the physical map).

8) While creating their own representation participants only adjusted location (7/7) and size (6/7). Other features of the Map Builder, such as sketching or adding text, were not used. Several participants commented that they were thinking of adding some labels, but none of them actually did.

9) Participants usually grouped computers based on the kind of food they provide, the amount of clicks needed to reach a specific kind of food (7/7), how often the computers are visited by Boris (6/7), if the computers have a screen or not (7/7), and if the computer is located inside or outside of the room (7/7).

10) In addition to grouping, some participants reduced the distances between computers to improve movement time, and some changed (usually increased) the size of computers to more efficiently use empty space.

Figure 3 illustrates how the map evolved during the course of the experiment. After the first 4 sessions, 3 out of the 7 participants reached a stable representation that they no longer modified. The post questionnaire revealed that the main reason for avoiding additional changes was that these users felt they had already experienced the representation extensively, and that any change to this established representation could cause confusion and therefore reduce performance in the game.

Based on these results we can formulate an answer to the first research question. During prolonged usage of a modifiable Radar View representation, users do strive to optimize the representation based on the task and personal preferences. The nature of the task is the main criterion for motivating the change; other less important criteria are the location of devices, the amount of available space, the visibility of devices, and the type of devices.

It's however still unclear if the new representation is more efficient than a physical location-based representation. It also remains difficult to derive how exactly and why tasks affected the change. The study also did not address mobility which is an important aspect that might influence the perception of the map and behavior of users, especially if the representation of the environment does not anymore match physical locations. Different approaches might be used to solve this issue, for example, the mobile device can be represented on the map as another static device, or the system can automatically position the device based on its distances from other devices represented on the map (for example, the mobile device can be shown next to the static device that is currently closest).

3.3 Design guidelines

Based on the results of the study we can formulate the following guidelines for building reaching interaction techniques that use a map-like representation:

- If the number of computers is small and they all have observable screens and interaction occurs only inside the represented area, a simple physical mapping such as the iconic map in ARIS system [4] is the best representation.
- If the interaction occurs outside of an environment, even in case when the environment is known to the users, it is wise to use a representation that allows better task-oriented interaction. However the mapping should be very clear to the users so they can easily remember it.
- In mixed environments a tool that allows some adjustments of the map has been proven to be useful.
- In situations where available space is limited, the exact spatial locations of devices can be sacrificed in favor of looser, order-based, relations.

4. Conclusions

One of the most promising reaching techniques is Radar View. We performed a user study that explored whether or not users appreciate the possibility of adapting radar maps to particular tasks and personal preferences and if so, which criteria are used to motivate these changes. A modified version of the Sketch Radar prototype, which provides an easy and quick way to manage maps of available devices, was used in the experiment.

The study confirmed that users indeed modify the map for different reasons, namely to more clearly represent the type or visibility of individual computers, and to clarify task-related relationships between computers. Since no explicit performance measures were gathered within the experiment, it remains undecided whether or not user-defined representations are more efficient than representations that agree closely with physical locations.

5. REFERENCES

[1] Aliakseyeu, D. & Martens, J.-B. (2006) Sketch Radar: A Novel Technique for Multi-Device Interaction. Proceedings of HCI'2006, Vol. 2, British HCI Group, 45-49.

[2] Bell, M., Chalmers, M., Barkhuus, L., Hall, M., Sherwood, S., Tennent, P., Brown, B., Rowland, D., Benford, S., Capra, M., Hampshire, A. Interweaving Mobile Games With Everyday Life. *Proc. of CHI 2006*, ACM Press (2006), 417-426

[3] Biehl, J.T. and Bailey, B.P. ARIS: An Interface for Application Relocation in an Interactive Space. *In Proc. of Graphics Interface*, *2004*, 107-116.

[4] Biehl, J.T. and Bailey, B.P. A Toolset for Constructing and Supporting Iconic Interfaces for Interactive Workspaces. *Proc of Interact 2005*, Springer, 699-712.

[5] Duncan, J. Response Selection Rules in Spatial Choice Reaction Tasks. In *Attention and Performance VI* Dornic, S. Ed., Erlbaum (1977), 49-61.

[6] Fitts, P. M., & Deininger, R. L. S-R compatibility: Correspondence among paired elements within stimulus and response codes. In *Journal of Experimental Psychology*, 48 (1954), 483-492.

[7] Nacenta, M.A., Aliakseyeu, D., Subramanian, S., and Gutwin, C. A comparison of techniques for Multi-Display Reaching. *Proc. of CHI 2002*, ACM Press (2002), 371-380.

[8] Robertson, G., Czerwinski, M., Larson, K. Data mountain: using spatial memory for document management. Proc. of UIST 1998, ACM Press (1998), 153-162.

[9] Stoakley, R., Conway, M., Pausch, R. Virtual reality on a WIM: interactive worlds in miniature. *Proc. of CHI 1995*, ACM Press (1995), 265-272.

Multiview User Interfaces with an Automultiscopic Display

Wojciech Matusik
Adobe Inc.
wmatusik@adobe.com

Clifton Forlines
Mitsubishi Electric Research Labs
forlines@merl.com

Hanspeter Pfister
Harvard University
pfister@seas.harvard.edu

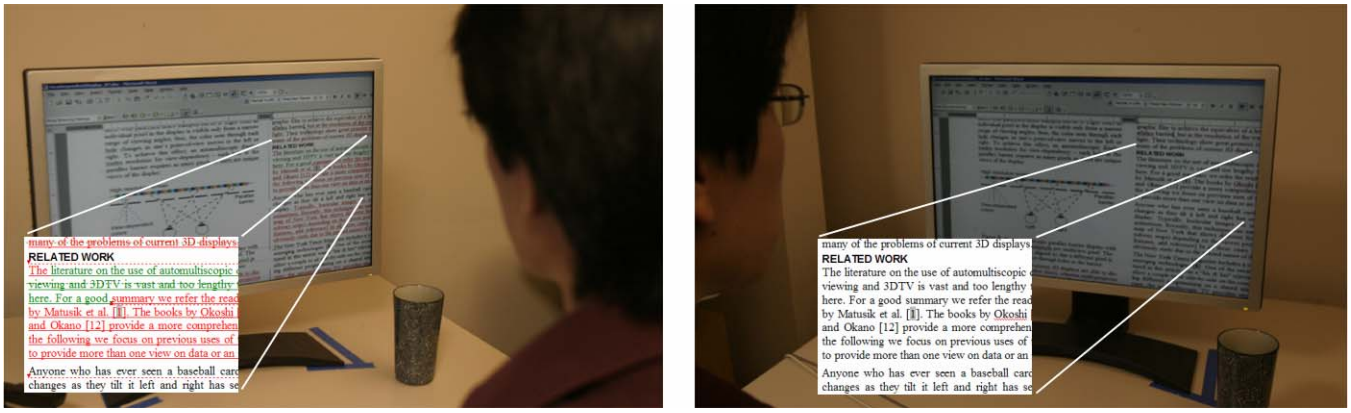


Figure 1: A display that reveals both the marked-up (*left*) and final (*right*) versions of a document to different POVs.

ABSTRACT

Automultiscopic displays show 3D stereoscopic images that can be viewed from any viewpoint without special glasses. These displays are becoming widely available and affordable. In this paper, we describe how an automultiscopic display, built for viewing 3D images, can be repurposed to display 2D interfaces that appear differently from different points-of-view. For single-user applications, point-of-view becomes a means of input and a user is able to reveal different views of an application by simply moving their head left and right. For multi-user applications, a single-display application can show each member of the group a different variation of the interface. We outline three types of multi-view interfaces and illustrate each with example applications.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design

Keywords: display, automultiscopic, multi-view

1. INTRODUCTION

Automultiscopic displays show 3D stereoscopic images that can be viewed by multiple people from many viewpoints without special glasses. These displays consist of an array of view-dependent pixels that reveal a different color to each eye of the user based on their point-of-view. View-dependent pixels can be implemented using conventional high-resolution displays and parallax-barriers (Figure 2). While the optical principles of multiview displays have been know for a century [6], only recently have high-resolution displays made them practical. Laptops with automultiscopic displays are commercially available [12], and (at the time of this writing) high-quality automultiscopic desktop monitors (such as the one shown in Figure 1) are available for around $3000 [10].

While automultiscopic displays are designed and built with the viewing of 3D images in mind, the properties of a multiview display create new and exciting opportunities for 2D user interfaces as well. If one thinks of a pixel's color as a function of not only its x and y coordinates, but also the point-of-view of the user, POV becomes a lightweight means of changing the appearance of the application. Where the color of a pixel is typically a function of its position on the screen, the color of a multiview pixel can be defined as:

$$pixelcolor_{x,y} = f(x, y, POV)$$

Figure 1 shows how the content of a graphical user interface changes as a function of point-of-view. In this example, the user of this view-dependent display is able to alter the appearance of the application by simply moving their head. Similarly, a group working in front of a view-dependent display can have separate views of the application.

The contribution of this paper is the outlining and illustration of several example view-dependent user interfaces using a commercial automultiscopic display. We group these examples into three groups based on the amount of continuity among views. We believe that these examples should aid the interested reader in their own exploration of multi-view user interfaces.

AVI'08, 28-30 May , 2008, Napoli, Italy

2. DEVICE DESCRIPTION

Commercial automultiscopic displays [7, 11] are based on parallax barriers or lenticular sheets placed on top of high resolution screens. Researchers have also used multi-projector systems [3, 8] to create these "glasses-free" 3D displays. Figure 2 shows a diagram of an automultiscopic display that reveals a different image to each eye of the user for the purpose of creating a 3D image. A high-resolution display is placed behind a parallax barrier – an opaque sheet with patterned holes stamped out of it. Light from an individual pixel in the display is visible only from a narrow range of viewing angles; thus, the color seen through each hole changes with changes to one's point-of-view.

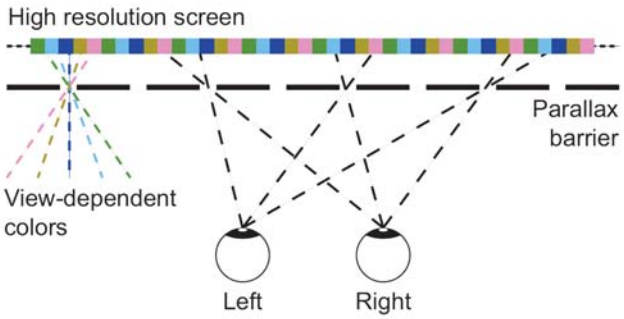


Figure 2: An automultiscopic parallax barrier display with five view-dependent subpixels per multiview pixel. The barrier and screen are aligned so that a different pixel is revealed to each eye through holes in the barrier.

Current commercial flat-screen 3D displays are able to display up to nine perspective views, whereas multi-projector 3D displays project 16 or more views. To project k views at m x n resolution requires an underlying image with km x n pixels. Vertical slits or lenses result in a k-fold loss in horizontal resolution. Slanting the slits or lenses at a small angle balances the loss of resolution in both directions. Parallax-barriers and lenticular sheets provide only horizontal parallax. Horizontal and vertical parallax is obtained using arrays of spherical lenses, or integral lens sheets. However, integral displays sacrifice significant spatial resolution in both dimensions to gain full parallax.

Important parameters of lenticular sheets and parallax-barriers are the number of lenticules (slits) per inch and the field-of-view (in degrees) of the viewing zone. The number of lenticules or slits corresponds to the effective multiview resolution of the display, i.e., the number of multiview pixels. The field of view of the display determines when the multiple views begin to repeat or "jump" back from view N to view 1. For example, on a display with 30 degrees field of view, a viewer moving their head toward the right will eventually move past 30 degrees, at which point, the image they see warps back to the image they would see if their head were 30 degrees back toward the left. While parallax-barriers reduce some of the brightness and sharpness of the image, lenticular sheets suffer from increased blurriness due to light diffusion inside the transparent sheet substrate. All multi-view displays suffer from crosstalk between views.

The company Holografika (http://www.holografika.com/) has developed a multiview display that uses a sheet of holographic film to achieve the equivalent of a lenticular or parallalax barrier, but at the resolution of the wavelength of light. Their technology show great promise to minimize many of the problems of current 3D displays.

3. RELATED WORK

The literature on the use of automultiscopic displays for 3D viewing and 3DTV is vast and too lengthy to be discussed here. For a good summary we refer the reader to the paper by Matusik et al. [8]. The books by Okoshi [9] and Javidi and Okano [2] provide a more comprehensive review. In the following we focus on previous uses of this technology to provide more than one view on data or an interface.

Anyone who has ever seen a baseball card whose image changes as they tilt it left and right has seen a lenticular display. Typically, lenticular images are used for simple animations. Recently, this technique was used to produce a map of New York that shows different features (such as subway stops) depending on the viewer position. In all these cases the images are obviously static due to the printed nature of the material.

The New York Times Magazine includes a yearly article on emerging technologies [5]. One of the recent projects featured in this article was a "his & her" television that would allow a couple to sit side-by-side on the couch while watching different programming on a shared television. This idea has recently been realized by Sharp Electronics, who have demonstrated a Triple Directional Viewing LCD capable of displaying different images to a driver, front seat, and back seat passengers. While similar to our solution, these multi-view displays are meant for multiple people and do not explore the use of multiple views for a single user.

Kakehi et al. [4] presented a view-dependent tabletop for group use. This system uses lumisty film and a Fresnel lens as a rear-projected tabletop with a separate projector for each user. These authors demonstrated several example applications, including a card game in which a player's cards were only visible to that player and were "face down" for other people standing around the display.

4. EXAMPLE APPLICATIONS

In this section, we describe several example applications that take advantage of a multi-view display. These examples use the users' point of view relative to the display as an input device; therefore, head movement becomes a lightweight means of getting different views of an interface.

While only a few examples are described in detail, we hope that these examples will allow the reader to recognize the value of multi-view interfaces for many types of applications. We organize these examples according to a *Continuity Among Views* and group them into the following areas:

1. Discrete Information
2. Layered, Registered Information
3. Continuous Variable Manipulation

Figure 3: (*top row*) A series of registered geospatial images displayed in different point of views. (*bottom row*) Different views show a continuous range of values in a high-dynamic range image.

4.1 Discrete views

In these examples, the views visible from different points-of-view contain discrete information. The effect is that the content of the screen changes completely as one moves their head into a different view region in front of the display.

Applications that allow the user to view data at multiple levels of magnification could benefit from a multi-view display that mapped different zoom levels to different points of view. For example, Figure 4 illustrates a system in which the area immediately around the mouse pointer is magnified and displayed in the far left and far right points of view. A user working with this application could simply move their head to the left or right to get a detailed view of the area of the display immediately around the mouse pointer. This would allow the user to make an accurate selection or perhaps just get a better view of a part of the display without mode switching into a zooming tool for what is a temporary change of magnification.

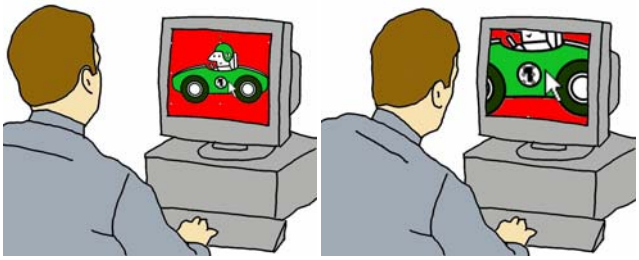


Figure 4: A magnified view around the mouse cursor is revealed from some points-of-view. A quick glimpse gives the user a detailed view without the need to switch tools.

Many applications could better support different uses of the same data by taking advantage of a multi-view display. For example, a multi-view enabled web browser could render a web page in the traditional manner in the primary point-of-view, and render the same web page in a summarized manner more appropriate for skimming in other points-of-view. A user searching through many pages could quickly skim pages by viewing them from one point-of-view and then switch to another point of view when a more detailed reading is desired.

4.2 Registered Layers of Information

The examples in this section take advantage of a multi-view display by showing different registered layers of information in each point of view. By allowing the user to control the layers of information by simply moving their head, their hands are free to continue performing input to control other facets of the application.

In our first example, we used a set of eight registered images showing different systems within the human body. The viewer is able to view different systems by moving their head left and right. The registration among images allows the viewer to see the relationship among the systems.

Our second example application uses layered geospatial information from Google Earth [1]. In this example, shown in Figure 3, different layers of information are visible from different points-of-view. All of the register images include a satellite photograph of a city, but different points of view augment this photograph with different layered information. From one view, the streets of the city are highlighted, from another the location of subway stations are visible, and so on. Were all of these layers of information made visible at the same time, the result would be a cluttered an unusable interface. By splitting these layers up into different view regions, we allow the user to switch between layers by simply moving their head left and right.

4.3 Mapping a Continuous Variable to Multiple Views

In these examples, we map a continuous variable to the user's view position. Any application with which the visualization can be controlled through the manipulation of a continuous variable should be able to take advantage of this method of control.

Our first example addresses a problem with high-dynamic range images: they contain more color information than can be properly displayed on a most displays. Applications built to view HDR images include GUI controls for manipulating the color range of interest. This range of interest is mapped to the visible range of the display, and portions of the image that fall above or below this range are clipped. We have mapped different ranges to different views on our automultiscopic display so that the viewer can inspect different color ranges by moving their head left and right

in front of the display. An example set of ranges is shown in Figure 3.

Our second example addresses problems that arise when viewing 3D volumetric information on a 2D display. One solution is to render the information in the dataset in a semi-transparent manner so that distance parts are visible through close parts. Another solution is to give the user controls over a clipping plane that they can use to view 2D slices of the 3D dataset. We built an example volumetric data viewer that maps different 2D slices through the data to different points of view. The user of this example application changes the clipping plane by moving their head relative to the display.

Many applications attempt to balance the need to display lots of information with the desire to maintain an easily assessable, clutter free interface. Oftentimes, application developers allow the user to control the level of detail that is displayed in the application. We propose that multiple levels of detail can be displayed simultaneously on a multi-view display and that the user can choose the most appropriate level of detail by changing their point of view relative to the display.

5. MULTI-USER CONSIDERATIONS

Multi-view user interfaces have the potential to greatly impact the field of CSCW and single-display groupware. Kakehi et al. have enumerated many of the potential uses of a multi-view collaborative tabletop [4], and a repurposed automultiscopic display can provide many of the same benefits. For example, a single application could be modified to display labels, menus, and text in different languages in each view region of the display. Rather than specifically set a language preference, a user would simply position themselves in the region that matched their preferred language. Similarly, other appearance preferences, such as font size, could be mapped to viewing position. In a competitive gaming scenario, each group member could be given private information that was not visible to competitors.

6. CONCLUSION

The demand for 3D applications is bringing automultiscopic displays into the mainstream because of their ability to display 3D content without the need for special glasses. The increased availability of these displays is creating an opportunity for the repurposing of such displays to provide multi-view interfaces for 2D applications. We have presented the means of using an automultiscopic display for multi-view interfaces, and have illustrated a collection of example applications that take advantage of multiple views. It is our hope that these examples will convince the interested reader of the value of multi-view interfaces and of the relative ease with which they can be experimented with using commercial devices.

7. ACKNOWLEDGMENTS

The authors would like to thank their colleagues for their helpful discussions and their help in producing this paper.

8. REFERENCES

1. Google Earth. Google Inc. http://earth.google.com/
2. Javidi, B. and Okano, F. eds., Three-Dimensional Television, Video, and Display Technologies, Springer Verlag, 2002.
3. Jeon, H.I., Jung, N.H., Choi, J.S., Jung, Y., Huh, Y., and Kim, J.S. Super multiview 3d display system using reflective vibrating scanner array. In *Stereoscopic Displays and Virtual Reality Systems VIII* (June 2001), Proceedings of SPIE, pp. 175–186.
4. Kakehi, Y. Iida, M., Naemura, T., Shirai, Y., Matsushita, M., and Ohguro, T. Lumisight Table: Interactive View-Dependent Display-Table Surrounded by Mutiple Users. ACM SIGGRAPH 2004 Emerging Technologies, etech_0016, Los Angeles.
5. Krantz, M. Television That Leaps Off the Screen. New York Times; Arts and Leisure Desk Late Edition - Final, Section 2, Page 1, Column 1, 1995 words.
6. Lippmann, G. (1908) Epreuves reversibles donnant la sensation du relief. Journal of Physics 7(4), 821–825.
7. Lipton L., and Feldman M. A new stereoscopic display technology: The synthagram. In *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems* (Jan. 2002), vol. 4660, pp. 229–235.
8. Matusik, W., and Pfister, H. 3D TV: A Scalable System for Real-Time Acquistion, Transmission and Autostereoscopic Display of Dynamic Scenes, *ACM Transactions on Graphics (TOG) SIGGRAPH*, (August 2004), 23(3), pp. 814-824.
9. Okoshi, T. 1976. Three-Dimensional Imaging Techniques. Academic Press.
10. Opticality Inc. http://www.opticalitycorporation.com/
11. Schmidt, A., and Grasnick, A. Multi-viewpoint autostereoscopic displays from 4d-vision. In *SPIE Stereoscopic Displays and Virtual Reality Systems* (Jan. 2002), vol. 4660, pp. 212–221.
12. Sharp Inc. http://www.sharp3d.com/

Adapting a Single-user, Single-display Molecular Visualization Application for Use in a Multi-user, Multi-Display Environment

Clifton Forlines[1,2], Ryan Lilien[1]

[1] Department of Computer Science
University of Toronto
Toronto, ON, Canada
lilien@cs.toronto.edu

[2] Mitsubishi Electric Research Labs
201 Broadway, 8th Floor
Cambridge, MA 02139 USA
forlines@merl.com

ABSTRACT

In this paper, we discuss the adaptation of an open-source single-user, single-display molecular visualization application for use in a multi-display, multi-user environment. Jmol, a popular, open-source Java applet for viewing PDB files, is modified in such a manner that allows synchronized coordinated views of the same molecule to be displayed in a multi-display workspace. Each display in the workspace is driven by a separate PC, and coordinated views are achieved through the passing of RasMol script commands over the network. The environment includes a tabletop display capable of sensing touch-input, two large vertical displays, and a TabletPC. The presentation of large molecules is adapted to best take advantage of the different qualities of each display, and a set of interaction techniques that allow groups working in this environment to better collaborate are also presented.

1. INTRODUCTION

Scientists wishing to understand the function of a protein must gain an understanding of its 3D shape: unlike in design, in molecular biology function follows form. Because of the bandwidth of the human visual perception system, 3D visualization is an appropriate and widely used means of conveying protein structure, and thus protein function. Proteins are themselves not visible: they are too small to reflect visible light in a meaningful way. Rather than being a constraint, this characteristic of proteins allows the scientist to choose among a large number of visual representations for proteins and other macromolecules, each one of which highlights certain features of the structure (An overview of the many ways in which a macromolecule can be represented visually and the strengths and weaknesses of these techniques can be found in [22]).

For the student or scientist, there are literally hundreds of software visualization packages to choose among. Some of the more popular packages are compared in [22]. While some choice from this diverse set should meet any single individual's needs, groups of researchers wishing to work collaboratively with a visualization application will run into many problems. These problems stem from the single-display, single-user assumptions that most application developers make. For example, personal computer workstations typically have a single mouse and keyboard, which can be difficult to share among the members in a group. Similarly, a group oftentimes has trouble crowding around a single desktop display. While a large projected display gives a good view to every group member, this type of presentation forces everyone into a shoulder-to-shoulder position, which may not be conducive to collaboration. Finally, groups working together often pass in and out of periods of independent work, which is not possible when sharing a single application.

Teams working together typically work face-to-face around a tabletop, sometimes surrounding themselves with materials hung or projected on the walls of their workspace. It seems natural that applications used by teams should be made compatible with this type of work environment. In this paper, we present an adaptation of Jmol [11], a popular open-source molecular visualization application, for use by a small group working together in a table-centric, multi-display computational environment, such as those described in [5, 20, 24] (Figure 1). While these environments are rare today, they will likely become commonplace, and the developers of molecular visualization applications may design their tools to best take advantage of the space; however, for now the adaptation of existing tools for use in these workspaces is a worthwhile endeavor.

AVI'08, 28-30 May , 2008, Napoli, Italy

2. RELATED WORK

2.1 Wrapping Single-user Software

Building software "wrapper" applications for multi-user settings has been the subject of much research. Greenberg [6] surveyed and discussed a large number of such projects all performed with the goal of providing shared-views among distributed, remote worksites. These projects shared the goal of ensuring that the same view is displayed on different remote machines so that separated users have a shared context for remote collaboration.

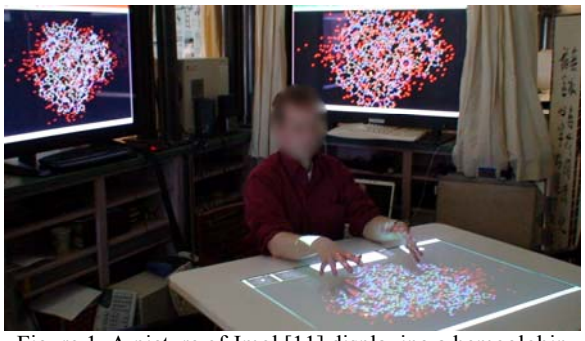


Figure 1. A picture of Jmol [11] displaying a hemoglobin molecule in our mutli-display, table-centric workspace.

Forlines et al. [5] demonstrated a system in which multiple instances of a geospatial application ran on multiple machines in a workspace, with each machine rendering a different view. Rather than keep the multiple instances of the software perfectly in sync, they allowed the machines to display slightly different, coordinated views of the same data with the goal of providing a group multiple point-of-views of the same geospatial location.

2.2 Multi-user Perception

One well-known phenomenon about individuals' perception is that objects are most easily recognized when presented at their canonical orientation [12]. This presents a problem with some tabletop interfaces because objects that have a strong axis of orientation may be upside-down or sideways to some people gathered around the table.

While there is no inherent axis of orientation for a protein, in practice groups familiar with a particular protein tend to arbitrarily set a canonical orientation. This orientation arises organically from the users' interaction with the molecule. Thus, although a biologist is likely familiar and fluent when dealing with an arbitrary orientation of their molecule, when given the choice they will likely adopt a single preferred orientation for a familiar protein. For these structures, vertical displays may be most appropriate for presentation to a group.

Unfamiliar proteins or unfamiliar regions of a protein without a universal "up" lack this canonical orientation and may benefit from being viewed in a particular orientation. For example, when trying to identify common folds between pairs of proteins, accepted perceptual psychology theories state that it would be helpful if the features in question were aligned with one another. With each member of a group sitting at a different side of the table, they are all provided with a different point-of-view of the protein; thus, it is more likely that at least one member of the team will have an advantageous view of the target features. This potential benefit of a tabletop display is in contrast to a vertical display, on which a feature that is presented in a disadvantageous rotation for one group member is presented in a disadvantageous rotation for all group members. It would seem that a collection of horizontal and vertical displays would be best for groups dealing with both familiar and unfamiliar structures.

2.3 Benefits of Multiple Displays and New Types of Displays

Multi-display workstations have become commonplace in recent years, and the performance and preferential benefits of using multiple displays has been the focus of many research projects [3, 7, 9, 21]. Similarly, large displays have been investigated, and have been found to have performance and preferential advantages [1]. The large number of pixels available in these workstations allows multiple views of a dataset to be presented simultaneously [15], which may aid not only the user's understanding of a dataset [19], but also the coordination of a group working together [5].

For viewing 3D structures, stereoscopic displays create the illusion of depth in an image (for a good overview of stereo technologies, see [10,16]). Intuitively, using a 3D display should lead to a better understanding of 3D structures. One variation of stereoscopic displays is the immersive CAVE environment, in which a user stands within a 6-sided workspace with images projected on the walls, floor, and ceiling [2]. CAVE environments have been shown to increase performance for some spatial tasks. While both stereoscopic displays and CAVE environments should aid a molecular biologist working alone, most of these technologies are not appropriate for group use.

2.4 Molecular Visualization

Cyrys Levinthal led a team of researchers at MIT in the 1960s that built the first computer system for the visualization and manipulation of molecular structures [13]. Using a monochrome oscilloscope, their system displayed molecules as simple wireframe models that rotated on the screen. Recognizing the usefulness of non-physical visualizations, many university and industrial groups built or purchased molecular visualization software to run on their department's mainframe computers.

Roger Sayle spent the early 1990s building the molecular visualization application RasMol [18] while working as a graduate student. RasMol's distinguishing feature was that it ran fast enough on personal computers to be useful to the large number of students and scientists without access to expensive mainframe computers. While the first, RasMol is by no means the only such application: there are over a hundred freely available molecular visualization applications available as of the time of this writing.

While there are many choices, these applications share the characteristic that they were built for personal computers and make the assumption that they will be used by a single-user. Similarly, most applications assume that there is only one display attached to the computer. Groups wishing to use these applications must share a single keyboard and mouse and crowd around a shared display.

Perhaps most similar to this paper, John Tate led a team of researchers in the development of a collaborative molecular visualization tool called MICE, the Molecular Interactive Collaborative Environment [23]. MICE uses VRML and a web-based interface to provide *distributed* researchers with a *single shared view* of a molecular scene. Our project differs in that our goal is to provide *co-located* researchers with *multiple, related views* of a molecular scene.

3. SYSTEM OVERVIEW

Figure 2 shows an overview of our system. The main component of the system is a Windows PC attached to a DiamondTouch [4] tabletop input device. Gestural commands on the tabletop control the local instance of Jmol running on the same machine. Networked client PCs running their own instances of Jmol connect to the tabletop machine on startup and receive appearance and Point-of-View (POV) scripts from the table. Finally, a second client application built to run on a TabletPC connects to the tabletop on startup and sends selection and appearance scripts to the table machine.

4. COMPONENT 1 – TABLETOP DISPLAY

At the heart of the system is a DiamondTouch input device. This table is capable of sensing and distinguishing multiple points of contact from up to four group members. This table is connected to a PC that acts as the main server of our application.

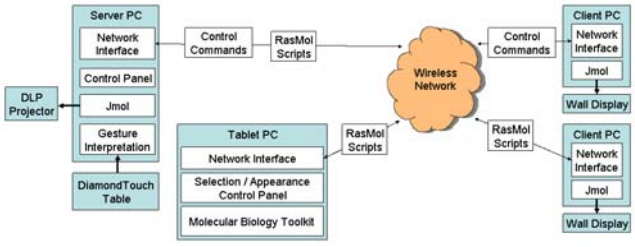


Figure 2. System Overview. Our multi-user, multi-display environment contains a tabletop display, a tablet display, and two wall displays driven by the four machines pictured above.

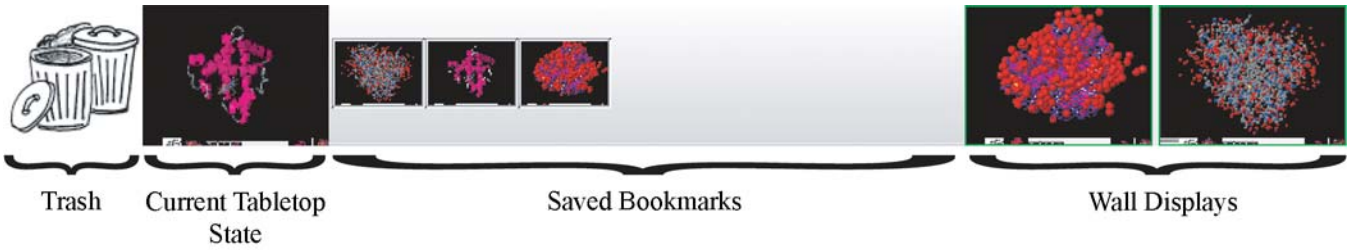


Figure 3. The Control Bar. This control contains a representation for the current tabletop state, bookmarks for saved states, representations for each wall display in the workspace, and a trashcan for deleting bookmarks.

4.1 Gestures for Point-of-view Control

To allow for quick and natural camera control, we implemented a set of gestures for controlling point-of-view on the tabletop. Touches made by any user on the table are mapped by a gesture interpreted to one of four commands.

Touches with a single finger tumble the molecule on the table according to the ArcBall rotation mechanism. Touching the table with two fingers and spreading them apart zooms the camera in, while pulling them together zooms the camera out. When a user grabs the tabletop with their whole hand, their touches are interpreted as panning commands and can "drag" the molecule around in the image plane. Finally, if a user touches the table with a closed fist, they can rotate the molecule in the image plane.

Many of the commands for selecting structures in a molecule and changing the visual presentation of a molecule are accessible in Jmol through a right-mouse click. To enable our users to access these commands, we added a thumb-tap gesture which results in a right-mouse click. When touching the table with ones index finger, a quick tap with ones thumb executes the command and pops up the contextual menu.

One final command added to a recent version of the prototype is a single-finger dwell. After dwelling on a portion of the molecule, the tabletop responds by centering the point-of-view on the atom directly under ones fingertip. All subsequent rotations occur around this new center.

4.2 Multi-user Input

Any one of four users sitting around the table can perform the gestural input described in the previous section. In our system, we choose to implement a simple floor-control mechanism that gives control of the application to the first user who touches the tabletop. All subsequent touches by other users sitting around the table are ignored until the initial participant completes their command and lifts their hands from the table.

4.3 Control Bar

Along one edge of the table, the application displays a control bar that provides the group with some additional functionality to control the behavior of the wall displays and to aid in collaborative discussions. A close-up of the control bar is shown in Figure 3.

The control bar is divided into three regions. On the left is a WIM (World in Miniature) representing the current state of the tabletop display. When the molecule is manipulated on the table, the appearance of this WIM updates to reflect the current state of the application. Touching and dragging this WIM into the middle region creates a new bookmark. Bookmarks are miniature WIMs that save the current state of the visualization as a script file. When bookmarks are clicked, or dragged onto the tabletop WIM, the script file is loaded and the previous state revisited. Using bookmarks, a group can easily save and return to a previous portion of the conversation – allowing groups to easily explore tangents and forks without loosing their place. Bookmarks can be removed from this area by dragging them to the trash.

On the far right of the control panel are WIMs for each of the wall displays currently connected to the system. A user can drag the tabletop WIM to a wall WIM to set the state of the wall machine, or can drag a bookmark onto a wall WIM to load that saved state on the wall machine. Finally, by clicking on any wall WIM, a user can enable / disable the point-of-view synchronization between the table and wall described in the next section.

Our prototype includes a single control panel. An alternative design would replicate this tool along each edge of the table. Bookmarks could either be shared among all of the control bars, or kept separate, allowing each user to bookmark moments in the conversation that they found important individually.

5. COMPONENT 2 – WALL DISPLAY

Each wall display is driven by a separate PC running Jmol and communicating to the tabletop server machine over the network. Communication messages consist mostly of RasMol script commands for changing the loaded protein or molecule, point-of-view, or appearance of the structure. By default, when a PDB file [17] is loaded on the tabletop, a load script is sent to all of the wall displays in the system. When a user alters the point of view on the tabletop, each wall display receives a message, which when executed causes them to display the same POV.

While POV and load commands are synchronized among the machines in the workspace, appearance commands are not. Only when the tabletop's appearance or a bookmark's appearance is specifically sent to a wall display does the appearance of the wall display change. In this way, a group can easily compose the workspace to display multiple views of the same structure using different representations of the molecules (Figure 4). As pointed out by Roberts [19], by simultaneously displaying the data in multiple ways, users may understand the information through different perspectives, overcome possible misinterpretations and perform interactive investigative visualization through correlating the information among views.

Two final commands from the tabletop server are listened for by the wall machines. The first command instructs the wall to ignore

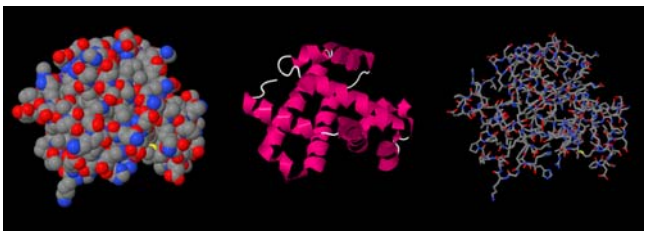


Figure 4. Three representations of the protein hemoglobin – space fill, cartoon, and stick. Each highlights different facets of the molecule and are used for different purposes.

point-of-view scripts from the tabletop. When a user clicks on a wall WIM on the tabletop control bar, this command is sent over the network to the corresponding machine. By freezing the point-of-view, a group can arrange their collection of displays to present multiple POV of the same protein or molecule. A second click on the WIM unfreezes the wall display. The second command is sent when the group quits Jmol on the tabletop display, and this command instructs the wall display to shut down and exit the application immediately.

Figure 5. Selections on the tablet are reflected on the tabletop display. In this figure, one of the four chains in the protein is selected. Selected atoms appear in light yellow.

6. COMPONENT 3 – TABLET DISPLAY

Often in molecular visualization, biologists choose to highlight the subset of residues or individual molecular components involved in a particular molecular function or interaction. This is useful because a protein may have hundreds of residues with only about 10 residues being important to any one particular interaction.

After experimenting with an early version of the system, it became clear that selecting sub-structures and individual atoms was too difficult and cumbersome. Indeed, accurate selection of small targets is well known to be difficult with ones fingers. To address this limitation, we built a second client application that runs on a TabletPC and allows for the quick and accurate selection of chains[1], secondary structure elements[2], and residues. Additionally, the tablet interface has controls for changing the appearance of the currently selected atoms.

To jumpstart the development of this selection and appearance application, we used the Molecular Biology Toolkit [14]. This Java-based toolkit from the San Diego Supercomputing Center provides a set of classes for loading, parsing, and manipulating molecular CIF [8] files.

To aid selection, we built a hierarchical selection widget that was placed on the left side of the screen. This widget displays the chains, secondary structural elements, and individual residues from the loaded file. Using the stylus, a user can select an element from any level, and selections are communicated over the network to the tabletop machine. Figure 5 shows how the selection of an entire chain made on the tablet is reflected on the tabletop. The majority of the tablet application contains controls for altering the appearance of the selected structures. These controls send a corresponding RasMol script to the tabletop machine that effects the visual presentation of the current selection.

In the 'Atoms' quadrant, there are several controls for changing the visual size of the atoms in the current selection and for hiding them completely. In the 'Styles' quadrant, there are six buttons that change the appearance of the current selection to one of six popular visualization schemes – Space-filling, Ball-and-Stick, Stick, Wireframe, Cartoon, and Trace. In the 'Surface' quadrant, there are controls for visualizing the surface of the molecular structure and for making this surface transparent or opaque.

In the 'Color' quadrant, there are controls for changing the color of the atoms in the current selection, as well as controls to color these atoms either by their element or by their residue.

Finally, at the top of the application, there are controls for selecting all of the atoms in the file, selecting none of the atoms in the file, inverting the current selection, and for toggling the highlighting of the current selection on the tabletop.

While our prototype workspace included a single TabletPC, there is no reason that each group member could not have their own tablet that allowed them to make their own selections.

7. EXAMPLE SCENARIO

Professor Ligand is interested in the molecular interactions between a solved enzyme (or protein) found in pigs and a naturally occurring ligand (or small molecule) that is known to bind with and inhibit the function of this enzyme. It is his hope that an understanding of this interaction will help with the development of drugs for a related family of enzymes in humans. Until recently, only the unbound form of the human and the porcine enzymes have been known. Today, Dr. Ligand and his team have managed to successfully solve (via x-ray crystallography) the structure of the porcine enzyme bound to the known small molecule inhibitor.

Dr. Ligand calls a meeting of the research team. The team members arrive carrying their laptop computers and sit down around the tabletop display. At the start of the meeting, one of Dr. Ligand's graduate students loads the recently solved molecular structure onto the tabletop.

While the complex compound initially appears as a cloud of white dots, another of Dr. Ligand's students quickly modifies the appearance of the structure to highlight the bound molecule and the interaction site. This representation is then sent to one of the large wall displays.

For comparison, the team then loads the unbound porcine form of the enzyme, and applies a similar color scheme to this molecule. It is sent to the second wall display for easy viewing by the team.

Side-by-side, the difference between the bound and unbound forms of this protein is obvious even to a non-expert: the presence of the ligand has induced a hinge-like motion to close part of the protein's binding site. The team centers the view on and rotates the view around one end of the bound ligand. Indeed, the "head" of the ligand is held in place by a trio of Lysine residues all of which have swung into place secondary to the motion of the hinge. By selecting and highlighting these residues in the unbound protein, the team is immediately able to see that they are close, but not neighboring in the unbound form of the porcine enzyme. By next displaying the unbound human enzyme, the team identifies a similar geometric arrangement of the binding site residues and explores the possibility that a similar hinge closure could be induced with the right small molecule. With this information gained, the team discusses a plan to identify such a small molecule inhibitor.

[1] A 'chain' is a single connected molecular component (i.e. if a graph is defined in which each atom is a vertex and each bond is an edge between the vertices corresponding to the atoms on the bond, then a chain refers to a single connected component). Some proteins and protein complexes consist of several chains.

[2] 'Secondary structures' are the common structural fragments of proteins. There are 3 main types - helix, strand, and coil.

8. CONCLUSION

This paper has presented the modification of the single-user single-display application Jmol for use in a multi-user multi-display workspace. This adaptation was informed by the needs of groups working collaboratively and by previous research in human perception and visualization. The workspace presented in this paper is, today, atypical; however, as display costs fall and new display form factors become commonplace, multi-display table-centric workspaces will become the norm. Our hope is that through the adaptation of existing tools to work in these spaces, we will gain valuable information that can inform the design of new tools built with these multi-user workspaces in mind.

9. REFERENCES

1. Ball, R. and North, C. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (Portland, OR, USA, April 02 - 07, 2005), 1196-1199.
2. CAVE (CAVE Automatic Virtual Environment). University of Illinois. http://inkido.indiana.edu/a100/handouts/cave_out.html
3. Czerwinski, M., Smith, G., Regan, T., Meyers, B., Robertson, G. and Starkweather, G., Toward Characterizing the Productivity Benefits of Very Large Displays. in *Proceedings of Human-Computer Interaction – INTERACT 2003,* (Zürich, Switzerland, 2003), 9-16.
4. Dietz, P. and Leigh, D. 2001. DiamondTouch: a multi-user touch technology. In *Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology* (Orlando, Florida, November 11 - 14, 2001). UIST '01. ACM Press, New York, NY, 219-226.
5. Forlines, C., Esenther, A., Shen, C., Wigdor, D., and Ryall, K. 2006. Multi-user, multi-display interaction with a single-user, single-display geospatial application. In Proceedings of the 19th Annual ACM Symposium on User interface Software and Technology (Montreux, Switzerland, October 15 - 18, 2006). UIST '06. ACM, New York, NY, 273-276.
6. Greenberg, S., Sharing views and interactions with single-user applications. in *Proceedings of the ACM.IEEE Conference on Office Information Systems*, (Cambridge, Massachusetts, USA), 227-237.
7. Grudin, J., Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use. in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (Seattle, Washington, USA, 2001), 458-465.
8. Hall SR, Allen FH, Brown ID (1991). The Crystallographic Information File (CIF): a new standard archive file for crystallography. *Acta Cryst* A47: 655-685.
9. Hutchings, D., Smith, G., Meyers, B., Czerwinski, M. and Robertson, G., Display space usage and window management operation comparisons between single monitor and multiple monitor users. in *Proceedings of the working conference on Advanced visual interfaces*, (Gallipoli, Italy, 2004), ACM Press, 32-39.
10. Javidi, B. and Okano, F. eds., Three-Dimensional Television, Video, and Display Technologies, Springer Verlag, 2002.
11. Jmol. http://jmol.sourceforge.net/
12. Jolicoeur, P. The time to name disoriented natural objects. *Memory & Cognition*, *13*. 289-303.
13. Levinthal, Cyrus (1966). Molecular Model-Building by Computer. Scientific American 214(6):42-52.
14. Molecular Biology Toolkit. http://mbt.sdsc.edu/
15. North, C. and Shneiderman, B., Snap-together visualization: a user interface for coordinating visualizations via relational schemata. in *Proceedings of the working conference on Advanced visual interfaces*, (Palermo, Italy, 2000), ACM Press, 128-135.
16. Okoshi, T. 1976. Three-Dimensional Imaging Techniques. Academic Press.
17. The RCSB PDB. http://www.pdb.org/
18. RasMol. http://www.umass.edu/microbio/rasmol/index2.htm.
19. Roberts, J.C., On Encouraging Multiple Views for Visualization. in *Proceedings of IEEE Symposium on Information Visualization InfoVis'98*, (Research Triangle Park, NC, USA, 1998), 8-13.
20. Streitz, N., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R.., i-LAND: An interactive Landscape for Creativity and Innovation. in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, (Pittsburgh, Pennsylvania, USA, 1999), 120-127.
21. Tan, D. and Czerwinski, M., Effects of Visual Separation and Physical Discontinuities when Distributing Information across Multiple Displays. in *Proceedings of OZCHI 2003 Conference for the Computer-Human Interaction Special Interest Group of the Ergonomics Society of Australia*, (Brisbane, Australia, 2003), 184-191.
22. Tate, J. *Structural bioinformatics - Chapter 7*. Bourne, P. and Weissig, H. eds. John Wiley & Sons, Inc. 2003.
23. Tate, J., Moreland, J., and Bourne, P. (2001) *Journal of Molecular Graphics 19*, p280-287. Design and Implementation of a Collaborative Molecular Graphics Environment.
24. Wigdor, D., Shen, C., Forlines, C., and Balakrishnan, R. 2006. Table-centric interactive spaces for real-time collaboration. In *Proceedings of the Working Conference on Advanced Visual interfaces* (Venezia, Italy, May 23 - 26, 2006). AVI '06. ACM Press, New York, NY, 103-107.

As Time Goes by – Integrated Visualization and Analysis of Dynamic Networks

Mathias Pohl
Dept. for Software Engineering University of Trier
pohlm@uni-trier.de

Florian Reitz
Dept. for Software Engineering University of Trier
reit4701@uni-trier.de

Peter Birke
Dept. for Databases and Information Systems University of Trier
birke@uni-trier.de

ABSTRACT

The dynamics of networks have become more and more important in all research fields that depend on network analysis. Standard network visualization and analysis tools usual do not offer a suitable interface to network dynamics. These tools do not incorporate specialized visualization algorithms for dynamic networks but only algorithms for static networks. This results in layouts that bother the user with too many layout changes which makes it very hard to work with them.

To handle dynamic networks the DGD-tool was implemented. It does not only provide several layout algorithms that were designed for dynamic networks but also different instruments for statistical network analysis. Network visualization and statistics are combined in a multiple view interface that allows visual comparison of several network layouts and several network metrics at the same time. Furthermore the time-dependent behaviour of structural changes becomes visible and facilitates the analysis of network dynamics.

Categories and Subject Descriptors

H.5.2 [**Information Systems Applications**]: User Interfaces—*Graphical user interfaces*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms

Dynamics of Networks, Human-Centered Visual Analytics

Keywords

Multiple and Integrated Views, Dynamic Network Visualization

1. INTRODUCTION

1.1 Motivation

In the recent years network theory has moved into the focus of social scientists. Relations between people, institutions, or other types of actors within a specified environment are considered to be more than just results of behaviors. These structures may also influence the actors' descisions in the future. To facilitate the analysis of such social networks researchers implemented visualization and analysis tools such as Pajek [1], UCInet (together with NetDraw) [2] or Visone [3]. However, these tools do not provide a suitable interface to cope with network dynamics. As this dynamics is another interesting and expressive dimension of information there was a need for a specifically designed visualization and analysis tool. This tool should be suitable to present a visual interface to time-varying networks on standard hardware (especially on standard displays). To achieve that goal we implemented the DGD system – a visualization and analysis tool for dynamic networks. It makes use of the multiple view paradigm: Several views are semantically combined by brushing and linking effects, e.g. selecting an object in one view highlights it in a different view. However, the goal of DGD to use it on standard size displays requires a less space-consuming seperation into multiple views (one could refer to the term "enhanced view" [4]).

1.2 Technical Preliminaries

When thinking of dynamic networks as a sequence of static networks their visualization is done by specifically crafted graph layout algorithms. These algorithms try to preserve the user's *mental map* of the network [5] – a fact that is cruicial for the understanding of structural changes. More technically such algorithms try to reduce the so-called *mental distance* between two consecutive layouts of a sequence [6]. One algorithm for this task is *Foresighted Graph Layout (*FLT*)* [7]. This layout strategy has already been examined regarding to its usefulness for dynamic graphs [8]. Furthermore the generic design of FLT allows the use of different layout paradigms [9].

However, the analysis of dynamic networks does not only require a sophisticated visualization technique. In many cases it is useful to have more information about the considered network. Such information can be obtained by computation of centrality measures for each actor as well as network properties. Obviously these information is not a single value but a sequence of different values. While dynamic network analysis tools like SIENA [10] compute detail statistical analyses about the evolution of a network DGD provides such results in its main window. This way the user is not only able to read exact values but also can visually compare values of different actors at different positions in the sequence.

2. COMPONENTS OF DGD

The main window of DGD consists of several views. After startup it presents the user an empty network visualization component that can be filled either by loading a file from disk or by modelling a net-

AVI'08 28-30 May, 2008, Napoli, Italy

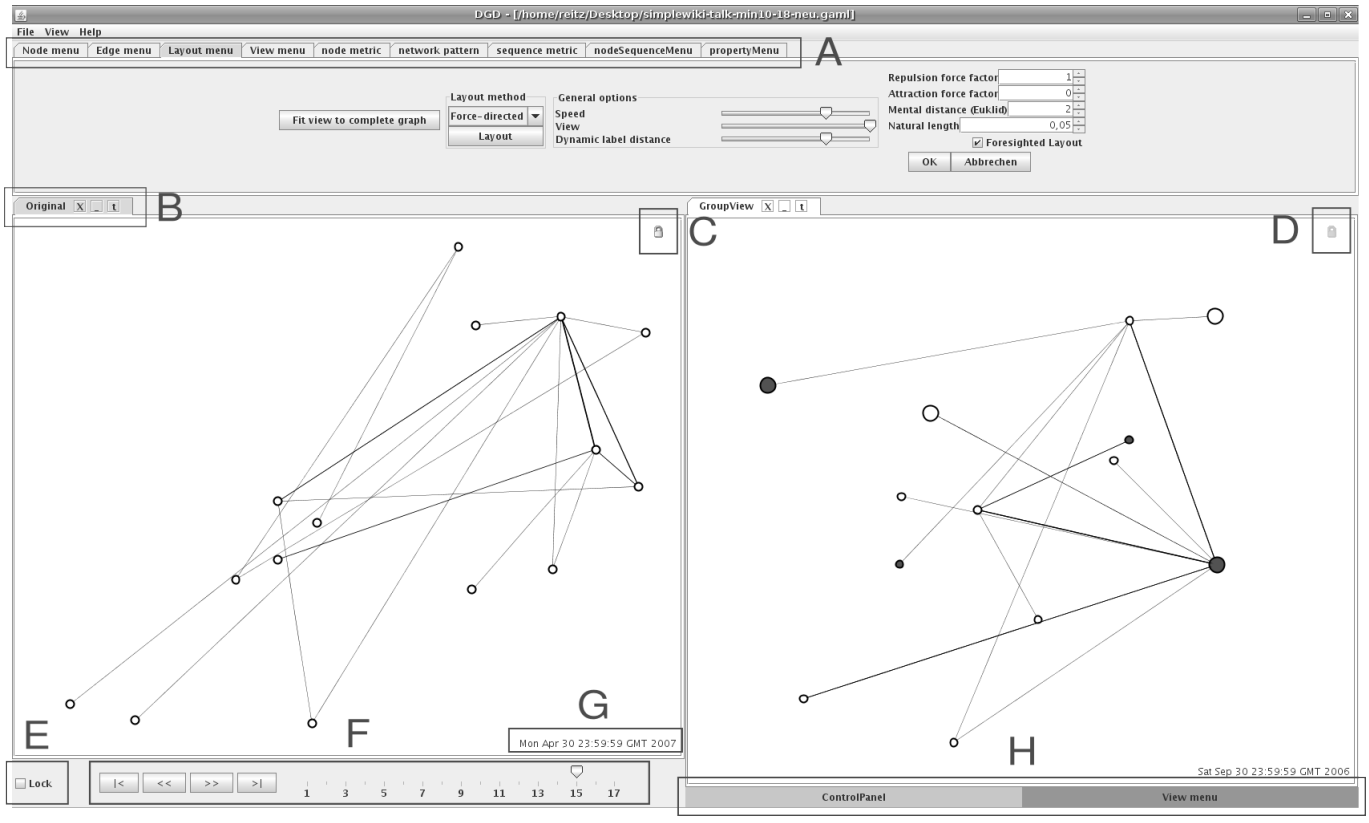


Figure 1: DGD at work: The menu panel (A) is visible at the top, the visualization components are arranged beneath the menu. Every visualization component has a title (B) and contains the sequence navigation controls (F). These controls become visible after moving the mouse on the "Control Panel"-area in the dynamic control section (H). After the mouse pointer exits the visualization component the controls disappear unless they are explicitly locked (E). The currently visible network is labelled at the bottom (G) and a manipulation lock (C) indicates whether the user is allowed to alter the network's structure.

work using the mouse. This component also provides the controls to navigate through the network sequence. In order to have several views onto the network DGD can display more than just one of this visualization components (see Fig. 1). This way it is possible to compare networks with different layouts or at different positions in the sequence.

Besides the network panel there is a menu panel that contains elements to modify the currently selected network component. Properties of nodes and edges can be changed as well as the used layout algorithm and its parameters. Furthermore this panel provides access to the network metrics.

The traditional menu bar of the main window is used for data import and export. DGD comes along with a XML-based file format that has been derived from GRAPHML [11] that provides no explicit support for dynamic graphs.

3. LAYOUT ALGORITHMS

The most important part in creating a view on a dynamic network, is the layout algorithm. In this context a dynamic network is considered to be an ordered sequence of static networks. It has to strike a balance between a high aesthetic quality and a good layout stability (this is usually referred to as *preservation of the mental map*). As mentioned before, DGD uses an implementation of the *Foresighted Graph Layout with Tolerance (*FLT*)* [7, 9]. The main idea behind FLT is to generate a global layout template for the sequence from the layout of the so-called *backbone*. Starting from this template FLT recommends to optimize the static layouts within tolerance limits by adjusting the induced layout.

The backbone contains only representatives of the semantically important nodes in the network sequence. More formally, for a given network sequence $g_1 = (V_1, E_1), \ldots, g_n = (V_n, E_n)$, a mapping $importance : V \rightarrow \mathbb{N}$ and a threshold δ the backbone contains the nodes $v \in \bigcup_{i=1}^{n} V_i$ with $importance(v) \geq \delta$ and the induced edges. The mapping importance defines a semantic on the set of nodes. By using the backbone DGD makes an individual definition of importance possible. By doing so users can modify the focus of layout stability.

The second phase of FLT concerns the iterative adjustment of the global layout template in order to obtain the final positioning in the layout of the static networks of the sequence. FLT bonds the degree of freedom for changes during this phase to a given threshold. FLT therefore needs *mental distances*, i.e. metrics for the preservation of the mental map [5] that are provided by DGD. Alltogether DGD currently contains implementations of three different layout paradigms. Each of them was redesigned to fit into the FLT context and hence fulfills the requirements for dynamic network visualization.

The **Force Directed Layouter** in DGD uses an implementation of the force-directed placement method by Fruchterman and Reingold [12]. The implementation uses a spring embedder with grav-

ity and simulated annealing. That way the algorithm distributes the nodes of a static network by computing pulling forces between connected nodes and pushing forces between every pair of nodes. The control of the different forces via the option's menu enables a visualization which highlights the clusters of the dynamic network. Thus DGD facilitates the identification of strongly connected components by a user.

The **Layer Based Layouter** extends the approach of Sugiyama [13]. A partition of the nodes will be evenly adjusted on different horizontal layers. The adjustment minimizes the number of backward edges, i.e. of edges $e = (v, w)$ with v on a higher and w a lower layer. Therefore, this layouter performs well for graphs with inherent hierarchic structure. Nevertheless this approach also can be used to layout arbitrary graphs so that users possibly detect an unknown hierarchic structure in the dynamic network. Since Sugiyama's approach contains two consecutive stages (layer assignment and layer ordering) that do not work iteratively FLT has been applied to these two stages. This results in the fact that the layer based layouter has controls for two mental distances. Depending on the user's interest in increased stability in layer assignment or in layer ordering he can select different values.

The **Orthogonal Layouter** is an extension of the approach of Brandes et al. [14] which is based on an orthogonal placement method by Fößmeier and Kaufmann [15], a so-called network-based approach. The implementation computes an orthogonal layout, i.e. edges are drawn as a sequence of hierarchical and vertical line segments. The discrete character of an orthogonal layout gives a good global overview of the layout of the static graphs. Evaluations show, that this approach allows a good layout for planar networks and allows users a fast detection of paths between connected nodes. Moreover, the recognition of clusters and shortest path will unfortunately be complicated.

4. VIEWS AND INTERACTION

In order to analyze networks it usually takes more than nice drawings. The possibility of filtering and manipulation and visible node centrality indicators are required. Furthermore characteristics of a network should be computable and visible within an analysis tool.

4.1 Available Views

To work with a dynamic network DGD provides the possibility to have several views to the same network. This can be useful when comparing two different layouts of the same snapshot or when comparing two different snapshots of the dynamic network. Besides the possibility of filtering data in each view seperately (see below) these views also support statistical network analysis by integrated pie diagrams (see Fig. 2).

This type of annotation has the advantage to present exact values to the user which can be crucial in some applications. However, this approach increases visual clutter and is not suitable for the visual exploration of a network. Thus, DGD also supports an integrated view of these metrics. The small pie diagrams show the node's normalized value concerning the selected metric. This approach makes it easier to find outliers in the network. This is an important feature especially in large networks where structures may appear similar although they are quite different.

4.2 Comparative Function Plot

While the integrated pie diagram allows for visual comparison of metrics for different nodes it is also desirable to have an impression of the values' time-dependent behavior. Therefore DGD provides a function plot view that shows the computed values for different

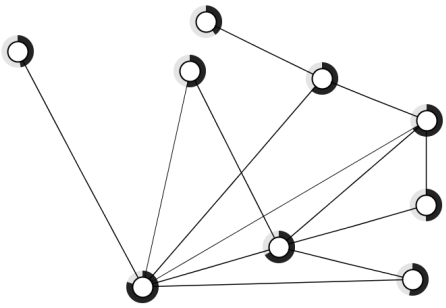


Figure 2: Integrated display of metric values using pie-charts.

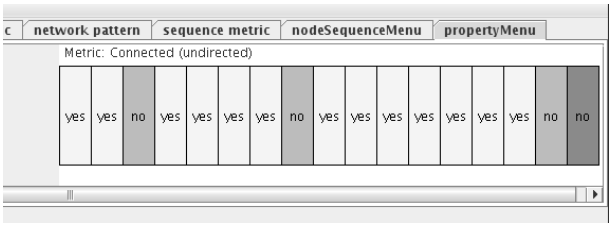


Figure 3: The view for binary properties in DGD. For every network the metric returns either true or false and the result is depicted in two different colors. The currently visible network is indicated by a darker color.

nodes over time (see Fig. 4). This view shows how selected nodes changed their kind and strength of relational integration. Furthermore nodes can be visually compared easily with each other and thus facilitate the identfication of dynamic roles in the network.

4.3 Binary Property View

The metrics and views presented so far work for any scalar value. However there are network properties that can either be *true* or *false* as whether the network is connected or is bipartite. For this binary metrics we implemented an additional view that contains boxes for each network and indicates negative answers with a red box and positive answers with a green box.

4.4 Filtering and Manipulation

DGD provides several mechanisms to work with the inspected data. Using the **GroupView** interface it is possible to alter visual objects. This can be done for each object seperately or by selecting groups of objects. Besides changing the color or the shape of an object it can also be hidden. Through the **ModeChange** interface a network can be semantically transformed. For instance it allows the conversion of a so-called two-mode network into a one-mode network.

5. RELATED WORK

The design of the DGD system was partially inspired by VISONE developped by Brandes and Wagner [3]. It offers many methods for network analysis and makes heavy use of an integrated visualization where computed values can be mapped on any visual property of nodes and edges. Due to its broad variety of available controls it tends to overload users. Other tools like PAJEK [1] and UCINET/NETDRAW [2] are primarily designed for computation of statistics.

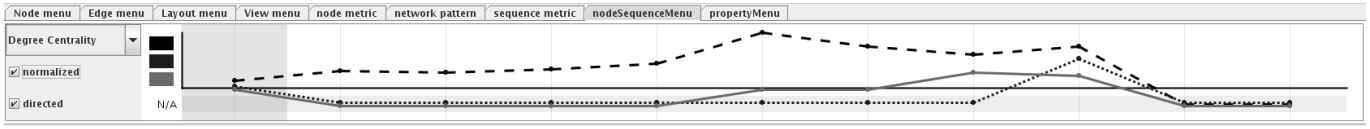


Figure 4: The (normalized) degree centrality values for the three selected nodes as a view in the menu panel. The plots are connected to the network view by using the same color. Authors 1 and 2 are not participating in the first months of the period. Their degree centrality is hence not available and indicated by values in the marked "N/A" area.

6. CONCLUSION

The dynamics of networks is another dimension of information that can be accessed using the DGD system. It incorporates several views onto the examined network as well as to the derived values from the network's structure according to specific metrics. The network visualization is done using a layout algorithm that is specifically crafted for dynamic networks as it pays attention to the user's mental map. The centrality measures can be integrated pie diagrams to facilitate the identification of interesting nodes or they can be displayed in a function plot view to enable visual comparison. The components of DGD are arranged such that it can be used on a usual-size display. This way DGD is a system for network analysis of dynamic networks that is easy to use and that works with standard hardware.

Acknowledgements

Thomas Söhngen implemented centrality measures. Mathias Pohl is partially supported by the Deutsche Forschungsgemeinschaft, grant no. DI 728/6-2.

7. REFERENCES

[1] Batagelj, V., Mrvar, A.: PAJEK – Program for Large Network Analysis. Connections **21** (1998) 47–57

[2] Borgatti, S., Everett, M.G., Freeman, L.C.: UCINet: Software for Social Network Analysis. Harvard MA: Analytic Technologies (2002)

[3] Brandes, U., Wagner, D.: Visone – Analysis and Visualization of Social Networks. In Jünger, M., Mutzel, P., eds.: Graph Drawing Software. Springer-Verlag (2003) 321–340

[4] Görg, C., Pohl, M., Qeli, E., Xu, K.: Visual Representations. In Kerren, A., Ebert, A., Meyer, J., eds.: Human-Centered Visualization Environments. Volume 4417 of Lecture Notes in Computer Science., Springer (2007) 163–230

[5] Misue, K., Eades, P., Lai, W., Sugiyama, K.: Layout Adjustment and the Mental Map. Journal of Visual Languages & Computing **6**(2) (1995) 183–210

[6] Bridgeman, S.S., Tamassia, R.: Difference Metrics for Interactive Orthogonal Graph Drawing Algorithms. In: Proc. of 6th Int. Symp. on Graph Drawing, GD. Volume 1547 of LNCS., Springer (1998) 57–71

[7] Diehl, S., Görg, C.: Graphs, They Are Changing. In: Proc. of 10th Int. Symp. on Graphdrawing, GD. Volume 2528 of LNCS., Springer (2002) 23–30

[8] Purchase, H.C., Hoggan, E., Görg, C.: How Important is the Mental Map. In: Proc. of 14th Int. Symp. on Graph Drawing, GD. Volume 4372 of LNCS., Springer (2006)

[9] Görg, C., Pohl, M., Birke, P., Diehl, S.: Dynamic Graph Drawing of Sequences of Orthogonal and Hierarchical Graphs. In: Proc. of 12th Int. Symp. on Graphdrawing, GD. Volume 3383 of LNCS., Springer (2004) 228–238

[10] Steglich, C., Snijders, T.A.B., West, P.: Applying SIENA. Methodology **2**(1) (2006) 48–56

[11] The GraphML File Format. `http://graphml.graphdrawing.org`, last visited Dec 20, 2007

[12] Fruchterman, T.M.J., Reingold, E.M.: Graph Drawing by Force-directed Placement. Softw., Pract. Exper. **21**(11) (1991) 1129–1164

[13] Sugiyama, K., Tagawa, S., Toda, M.: Methods for Visual Understanding of Hierarchical Systems. IEEE Transactions on System, Man and Cybernetics, SMC **11**(2) (1981) 109–125

[14] Brandes, U., Eiglsperger, M., Kaufmann, M., Wagner, D.: Sketch-Driven Orthogonal Graph Drawing. In: 10th Int. Symp. on Graph Drawing. Volume 2528 of Lecture Notes in Computer Science., Springer (2002) 1–11

[15] Fößmeier, U., Kaufmann, M.: Drawing high degree graphs with low bend numbers. In: 3rd Int. Symp. on Graph Drawing. Volume 1027 of Lecture Notes in Computer Science., Springer (1996) 254–266

Revealing Uncertainty for Information Visualization

Meredith Skeels[1], Bongshin Lee[2], Greg Smith[2], and George Robertson[2]

[1]Biomedical and Health Informatics
University of Washington
Seattle, WA 98195
mskeels@u.washington.edu

[2]Microsoft Research
One Microsoft Way,
Redmond, WA 98052
{bongshin, gregsmi, ggr}@microsoft.com

ABSTRACT
Uncertainty in data occurs in domains ranging from natural science to medicine to computer science. By developing ways to include uncertainty in our information visualizations we can provide more accurate visual depictions of critical datasets. One hindrance to visualizing uncertainty is that we must first understand what uncertainty is and how it is expressed by users. We reviewed existing work from several domains on uncertainty and conducted qualitative interviews with 18 people from diverse domains who self-identified as working with uncertainty. We created a classification of uncertainty representing commonalities in uncertainty across domains and that will be useful for developing appropriate visualizations of uncertainty.

Categories and Subject Descriptors
H.5.m [**Information Interfaces and Presentation (e.g., HCI)**]: Miscellaneous.

General Terms
Experimentation, Standardization.

Keywords
Information visualization, uncertainty visualization, qualitative research, user-centered design.

1. INTRODUCTION
When information is shown in a computer interface, it often appears absolute. The native machine or language data types used to store numerical data employ a very high level of precision. There is no sense of the level of certainty in that data or the degree to which the data is only possibly true. However, in reality data is rarely absolutely certain. By developing ways to make the uncertainty associated with data more visible, we can help users better understand, use, and communicate their data.

There has been a significant amount of research on uncertainty in fields such as information theory [5] and probabilistic reasoning [12]. However, these fields focused on how to compute uncertainty by developing a formal mathematical method. In our study it became clear that uncertainty is a complex concept that occurs in various domains and does not always appear as a quantifiable probability.

Work on uncertainty within domains can inform the design of visualizations, but unfortunately uncertainty is referred to inconsistently within and among domains. Just within the domain of geography, for example, a previous review of models of information uncertainty resulted in an outline of challenges for future research [8]. These challenges include, "understanding the components of uncertainty and their relationships to domains, users, and information needs," "developing methods for depicting multiple kinds of uncertainty," and "developing methods and tools for interacting with uncertainty depictions." Within the amorphous concept of uncertainty there are many types of uncertainty that may warrant different visualization techniques. Before we begin to design these visualizations we need a better understanding of how users view uncertainty and how it is currently represented. To that end, we have reviewed existing work on uncertainty within a number of domains, created an initial classification of uncertainty, and empirically evaluated and improved upon the classification.

AVI'08, 28-30 May , 2008, Napoli, Italy

2. RELATED WORK
Much of the previous work on both the visualization of uncertainty and the classification of uncertainty occurs within isolated domains. The predominant consensus among papers on uncertainty appears to be that uncertainty has been defined many ways and is referred to inconsistently in a variety of fields. MacEachren *et al.* state, "Information uncertainty is a complex concept with many interpretations across knowledge domains and application contexts" [8].

2.1 Visualization of Uncertainty
Most research on visualizing uncertainty is found in geographic visualization, geographic information science, and scientific visualization. The main techniques developed include adding glyphs [14], adding geometry, modifying geometry [4], modifying attributes, animation [2], and sonification [7]. These techniques have been applied to a variety of applications such as fluid flow, surface interpolants, and volumetric rendering.

CandidTree shows two types of structural uncertainty based on the differences between two tree structures [6]. Olston and Mackinlay introduced visualizations to address two forms of uncertainty: error bars for showing statistical uncertainty and ambiguation for showing bounded uncertainty [10].

Unfortunately, most uncertainty visualizations are isolated efforts designated for a specific purpose. To move forward with the challenges of visualizing uncertainty and creating interfaces for interacting with uncertainty in data, we need a model of uncertainty that covers the needs of users in multiple domains.

2.2 Classification of Uncertainty
We began our research by examining the definitions and classifications of uncertainty developed within several domains for common themes and overlap. Within the domain of weather modeling, Pang and his colleagues have worked extensively on

visualizing uncertainty in weather models [11]. Their model of uncertainty describes how uncertainty can be introduced at "acquisition," including issues with measurement or statistical variation; at "transformation," including any manipulation of data; or at "visualization." Within the domain of intelligence information analysts, Thomson *et al.* propose a typology of categories of uncertainty focusing on different types of uncertainty instead of sources of uncertainty [13]. Their categories include: accuracy/error, precision, completeness, lineage, currency/timing, credibility, subjectivity, and interrelatedness. One attempt to describe uncertainty outside any specific domain is a taxonomy of imperfect information, which includes "corrupt data/info," "imperfect presentation," "uncertainty," "info too complicated," "inconsistency," and "incomplete info" [3]. The taxonomy differentiates between uncertainty and concepts (e.g., incomplete info) that others (e.g., Thomson *et al.*) include within uncertainty.

In the domain of decision support and policy making, Walker *et al.* describe a way to convey uncertainty in a model to decision makers [14]. Their three dimensions of uncertainty include: location (context, model, or input), level (from deterministic to "total ignorance"), and nature (epistemic, meaning it could be clarified with more research, or variability, meaning uncertainty due to "inherent variability"). However, Norton *et al.* argue against this model by asserting that instead of seeing uncertainty as something additive that can be simplified to a level of uncertainty, we should view all the aspects of uncertainty associated with any decision [9]. This disagreement is an example of how epistemological differences between fields add to the difficulty of creating a unified classification of uncertainty.

Previous models of uncertainty have not been empirically evaluated and it is not obvious how to select between the models or integrate them into a single model.

3. CLASSIFICATION DEVELOPMENT

Based on our review of the literature on uncertainty from specific domains, we created our own preliminary classification of uncertainty spanning domains for the purpose of information visualization. We then refined our classification based on interviews we conducted with people from several different fields who encountered uncertainty in their own data.

3.1 Initial Classification

In the literature, we identified five common types of uncertainty discussed using different language in disparate domains. **Approximation** is often necessary in science and other domains, but it leads to uncertainty. Various techniques are used to attempt to measure or describe a phenomenon even when it cannot be measured or described with perfect precision. **Predictions** can be projections of future events, which may or may not happen. Prediction also is similar to developing an explanation of something that has already happened when the true explanation is not known. Model building is an example of a way to do prediction about the past or future. **Disagreement or Inconsistency** between experts in a field or across datasets is an indication of uncertainty. **Incompleteness** in datasets including missing data or data known to be erroneous also causes uncertainty. Lastly, **credibility** of data or of the source of data is another type of uncertainty described in the literature.

3.2 Qualitative Study of Uncertainty

To get a deeper understanding of uncertainty across domains, we conducted a formative interview-based study. We were particularly interested in gathering examples of uncertainty and learning how people currently represent and handle uncertainty. We then used the data to improve our classification.

3.2.1 Participants

We recruited 18 participants in the Greater Puget Sound area who self-identified as having aspects of uncertainty in their work. They came from both academic and industry settings including students, established researchers, and practitioners. Several participants worked in computer science with specialties including robotics, machine learning, databases, visualization, perceptual computing, and computer graphics. One participant was a former radiologist and other participants were from psychology, journalism, biology, bioinformatics, intelligence, bioengineering, and ecology.

3.2.2 Interview Methods

We conducted a 30-60 minute interview with each of the 18 participants individually. We took extensive field notes as well as audio recording. Some participants also provided screenshots or pointers to examples of uncertainty in their work. Interviews followed an interview guide, but were open-ended and exploratory. We began with open-ended questions about the uncertainty they encounter in their work. As they described uncertainty we asked for specific examples and asked them how they dealt with uncertainty. Towards the end of the interview we asked each person if they encountered disagreement, credibility issues, or incomplete data (if those issues had not previously been covered). We also asked participants to define uncertainty, asked them how they represented uncertainty or had seen it represented, and asked if they had seen any visualizations of uncertainty.

3.2.3 Analysis Methods

Within our team, we used affinity diagramming to collaboratively analyze our data [1]. This process began with individual thoughts and examples from the interviews broken out onto pieces of paper. The aim of this study was to evaluate and improve our classification. As we went through the pages, we tried to classify the thoughts and examples into our initial categories of uncertainty. When we discovered examples that did not fit our scheme we placed them in a new stack or adjacent to the stack with the closest fit. When we had multiple examples that did not fit into one of our existing classifications we attempted to redefine and iterate on our classification to accommodate the new type of uncertainty. About two thirds of the way through the data our classification stopped changing and the remaining examples fit into the new classification.

4. RESULTS AND DISCUSSION

We present our results in the form of an improved classification with descriptions of how the interview data guided the classification. One important concept introduced by our participants is the idea of levels of uncertainty. Visualizations showing levels of uncertainty could provide ways to show multiple types of uncertainty within a single dataset. We also present participants' definitions of uncertainty, how participants currently represent uncertainty, and what they do with uncertainty.

4.1 Definition of Uncertainty

Most participants had some difficulty providing a definition of uncertainty when asked, but there seemed to be agreement that uncertainty often happens in situations without complete knowledge. Participants used phrases like "imperfect knowledge," "inadequate information," and "lack of absolute knowledge" to describe uncertainty. Some participants saw uncertainty as a time when the probability of something is not 1.0 while others described it with more qualitative labels.

4.2 Classification of Uncertainty

One of the most important concepts resulting from our interviews is that multiple types of uncertainty are often associated with a single dataset and can be thought of as levels or layers of uncertainty. For example, Participant 5 described uncertainty about the measurements he got from scientists and then said that on top of that there was also "inference uncertainty" about the inference methods he chose. A few participants explicitly referred to "levels" of uncertainty. Participant 13 worked on computational photography and described the type of inference he used to try to remove blurring from images. He then distinguished uncertainty in the probabilistic inference from "another level of uncertainty" caused by noise in the sensor and lens variables. The sense of multiple kinds of certainty and different levels of uncertainty in a dataset or process are captured in our classification (Figure 1).

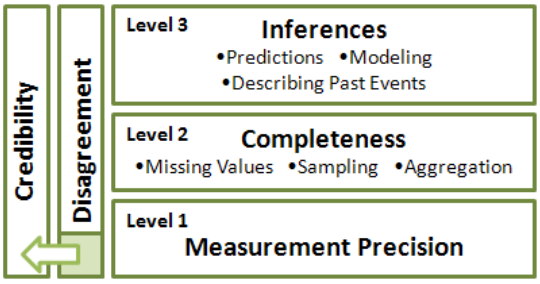


Figure 1. Improved classification showing layers.

4.2.1 Measurement Precision – Lowest Level

Uncertainty due to **imprecise measurements** came up frequently in our interview data and spanned domains. This category of uncertainty covers any variation, imperfection, or theoretical precision limitations in measurement techniques that produce data. Sometimes this imprecision is represented explicitly by a range that the true value is probably in (e.g., confidence interval). However, measurement precision uncertainty is often simply a data point that is known to be potentially flawed. In the example Participant 13 discussed above, there was measurement precision uncertainty from camera lens variability that was not constant enough to be modeled and adjusted for. He did not have a representation of certainty; instead, he had data points known to be somewhat uncertain.

4.2.2 Completeness – Middle Level

Completeness was an issue across domains as well. Some participants described **sampling** as a strategy for representing the values of some population. **Missing values** also represent incompleteness uncertainty. **Aggregating** or summarizing data in an irreversible way can also be a cause of uncertainty since once data has been summarized, information is lost and the data is no longer complete.

An important concept within completeness, that spanned domains, is **unknown unknowns**. Participant 18 distinguished the information you know (*known knowns*) from the information you know exists, but do not have (*known unknowns*) from the information you do not even know you are missing (*unknown unknowns*). The participants who discussed this distinction agreed that the *unknown unknowns* are the worst kind of missing information. When you do not know you are missing important information you are more certain than you should be.

4.2.3 Inference – Highest Level

Inference is a fairly broad category, spanning all types of modeling, prediction, and extrapolation. Inference has a tight relationship with decision-making: it is how data is infused with meaning and transformed into decisions. **Modeling** of any kind, ranging from probabilistic modeling to hypothesis-testing to diagnosis, falls in this category. For example, Participant 16 described the need to take a set of medical symptoms, either as a care provider or health consumer, and fit them into a model of illness. **Prediction** involves inferring future events by creating an abstraction of the causal relationship between current or past data and future occurrences. **Extrapolation** into the past, a complement to prediction, involves using data to try to recreate or make inferences about past events. For example, Participant 1 was interested in locations and paths of devices and people. He could use path data (inferred from location data) to try to identify where someone was in the past.

4.2.4 Disagreement – Spans Levels

Disagreement leads to uncertainty and spans the three levels. At the measurement precision level, disagreement happens when the same thing is measured multiple times or by different sources and the measurements are not the same. At the completeness level, disagreement comes from overlapping but not identical datasets. At the inference level, disagreement comes from two (or more) different conclusions being drawn from the same data. This could be two (or more) experts looking at a dataset and coming to different conclusions, or it could be applying two different mathematical models to a dataset to do inference. Participant 5 described an instance of disagreement at the inference level. Part of his work involved using multiple mathematical models of evolutions to predict the phylogeny of a virus. Each model produced a slightly different phylogeny and thus disagreement. Disagreement and credibility are often associated because as soon as disagreement occurs credibility is often called into question.

4.2.5 Credibility – Spans Levels

Credibility is a type of uncertainty that spans the three levels and is often difficult to measure. An information source that produces data that conflict with other data, has produced unreliable data in the past, or is otherwise suspect for some reason leads to uncertainty. Individuals may have different judgments about what constitutes a credible source. Participant 18, an ecologist, discussed building relationships with people and organizations over time and assigning different levels of credibility based on their level of expertise and on his experiences with them.

4.3 Levels of Uncertainty

As we classified examples of uncertainty into different kinds of uncertainty, we began to see a pattern in the way uncertainty compounds or stacks in datasets. Participants were not describing just one type of uncertainty, but instead were discussing uncertainty about multiple aspects of their work and occasionally used the word "level" to describe a higher or lower level form of uncertainty. After exploring this concept in the data, we assigned **Measurement Precision** to the lowest level type of uncertainty, **Completeness** to the middle level, and **Inference** to the highest level (Figure 1). **Credibility** and **Disagreement** are types of uncertainty that occurred along with, or on top of, each of the other types of uncertainty so they span the three levels. This does not mean that every dataset or project will involve every level of uncertainty, but many projects involved more than one level of uncertainty. One reason levels of uncertainty are so crucial and problematic in our participants' experiences is that uncertainty within one level, even if well-quantified at that level, rarely can be adequately transformed or accounted for at another level when the decision-making process requires a transition between levels.

4.4 Dealing with Uncertainty

The degree to which uncertainty in a dataset impacts an eventual outcome is hard to quantify. Participants described several strategies for dealing with uncertainty, but the predominant feeling seemed to be that the uncertainty was complex and difficult to describe, let alone deal with. Part of the problem might be that it is difficult to transform measurable uncertainty at one level into meaningful information at another level. It is also difficult to clearly convey the complexity of multiple levels of uncertainty to others. At some point, participants had to choose to do one of two things: live with the uncertainty or try to become more certain. Participants made this decision based on the potential impact of being wrong and based on how successful they felt they would be in improving their certainty.

4.5 Representations of Uncertainty

One of the challenges for visualizing uncertainty is that it is often not expressed in a standard quantification. We asked participants how they convey uncertainty and how they represent uncertainty.

4.5.1 Formats of Uncertainty

Some participants had quantifications of uncertainty they routinely used. In computer science, participants tended to define uncertainty in terms of probabilities representing a belief that something is true. The other quantification of uncertainty we saw was a range (e.g., confidence interval, error bound).

Many participants had uncertainty they did not quantify. Instead they used looser qualitative labels in communicating with others, but these labels were rarely stored with the data. Participant 8 described it in terms of t-shirt size: "small, medium, large, and XL." These were not standardized definitions, but were constructs created and used within a group. Participants also used words such as "likely" and "probably" to convey their own belief in an assertion or value.

4.5.2 Visualization of Uncertainty

By far the most commonly mentioned visualization was error bars. Some participants expanded the idea of an error bar to apply to location as well, describing a point with a circle around it. One participant described a sphere surrounded by a buffer zone (or error bar). Other visualizations of uncertainty included showing distributions with box plots and using data plots with quartiles. Participant 5, who dealt with evolutionary trees, mentioned tree alignment, described color coding branches, and adding icons (often asterisks) to branches to indicate certainty. Several participants expressed frustration with the difficulty of communicating certainty to others in a useful way.

5. FUTURE WORK AND CONCLUSION

Our motivation for categorizing uncertainty across domains was to eventually create useful visualizations that provide a more accurate depiction of the data. Our next step will be to identify ways to visualize different types of uncertainty and find ways to convey the layers of uncertainty that exist within a dataset.

The classification of uncertainty we have proposed spans domains and will be useful for incorporating indicators of certainty into visualizations of data. Our classification is based on a review of literature from several domains and on interviews with 18 people working with uncertainty in several fields. We found that participant were aware of uncertainty at many levels in their data and expressed discomfort at their inability to be transparent about showing their uncertainty. Our classification better describes the broad range of uncertainty across domains, provides a structure for visualizing uncertainty, and will ultimately help develop visualizations that make uncertainty visible.

6. ACKNOWLEDGMENTS

We thank our participants for their time, patience, and insights. We also thank John Stasko for his thoughtful comments.

7. REFERENCES

[1] Beyer, H. and Holtzblatt, K. 1998. Contextual design: defining customer-centered systems. Morgan Kaufmann Publishers, San Francisco, CA.

[2] Gershon, N.D. 1992. Visualization of Fuzzy Data using Generalized Animation, *IEEE Symposium on Visualization 1992*. IEEE Computer Society Press: Chicago, 268-273.

[3] Gershon, N.D. 1998. Visualization of an Imperfect World. *IEEE Computer Graphics and Applications*, 18, 4, 43-45.

[4] Grigoryan, G. and Rheingans, P. 2004. Point-based probabilistic surfaces to show surface uncertainty, *IEEE Transactions on Visualization and Computer Graphics 2004*, 10, 5, 564-573.

[5] Klir, G.J.2005. *Uncertainty and Information: Foundations of Generalized Information Theory*. Wiley-IEEE Press.

[6] Lee, B., Robertson, G.G., Czerwinski, M., and Parr, C.S. 2007. CandidTree: Visualizing Structural Uncertainty in Similar Hierarchies. *Proc. Interact 2007*. 250-263.

[7] Lodha, S.K., Wilson, C.M., and Sheehan, R.E. 1996. LISTEN: sounding uncertainty visualization, *Proc. Visualization 1996*. IEEE Computer Society, 189-195.

[8] MacEachren, A.M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., and Hetzler, E. 2005. Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know. *Cartography and Geographic Information Science*, 32, 2, 139-160.

[9] Norton, J.P., Brown, J.D., and Mysiak, J. 2006. To what extent, and how, might uncertainty be defined? Comments engendered by "Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support" (Walker, 2003). *Integrated Assessment*, 6, 1, 83-88.

[10] Olston, C. and Mackinlay, J.D. 2002. Visualizing Data with Bounded Uncertainty, *Proc. of InfoVis 2002*, IEEE Computer Society, 37-40.

[11] Pang, A.T., Wittenbrink, C.M., and Lodha, S.K. 1997. Approaches to uncertainty visualization. *The Visual Computer*, 13, 370-390.

[12] Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann.

[13] Thomson, J., Hetzler, B., MacEachren, A., Gahegan, M., and Pavel, M. 2005. A Typology for Visualizing Uncertainty. *Proc. SPIE-VDA 2005*, SPIE/IS&T.

[14] Walker, W.E., Harremoes, P., Rotmans, J., van der Sluijs, J.P., van Asselt, M.B.A., Janssen, P., and Krayer von Krauss, M.P. 2003. Defining Uncertainty: A conceptual basis for uncertainty management in model-based decision support. *Integrated Assessment*, 4, 1, 5-17.

[15] Wittenbrink, C.M, Pang, A.T., and Lodha, S.K. 1996. Glyphs for Visualizing Uncertainty in Vector Fields, *IEEE Trans. on Visualization and Computer Graphics*, 2, 3, 266-279.

Perceptual Usability: Predicting changes in visual interfaces & designs due to visual acuity differences

Mike Bennett
Imaging, Visualisation & Graphics Lab
Systems Research Group
School of Computer Science & Informatics
University College Dublin, Ireland
mike.bennett@ucd.ie

Aaron Quigley
Imaging, Visualisation & Graphics Lab
Systems Research Group
School of Computer Science & Informatics
University College Dublin, Ireland
aaron.quigley@ucd.ie

ABSTRACT

When designing interfaces and visualizations how does a human or automatic visual interface designer know how easy or hard it will be for viewers to see the interface? In this paper we present a perceptual usability measure of how easy or hard visual designs are to see when viewed over different distances. The measure predicts the relative perceivability of sub-parts of a visual design by using simulations of human visual acuity coupled with an information theoretic measure. We present results of the perceptual measure predicting the perceivability of optometrists eye charts, a webpage and a small network graph.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Evaluation/methodology, Screen design, Theory and methods

1. INTRODUCTION

Interactive dynamic visual displays are becoming increasingly pervasive [6, 2]. As display research and materials technology advances we can expect clothes, floors, tabletops, buildings, human skin and many other physical surfaces to continue getting turned into realtime visual displays. The implications of this are that an increasingly important facet of visual interface design will be catering to viewing interfaces and visualisations in a range of changing environments [14]. An advantage of turning previously static signage and materials into displays is that it offers the ability to improve the usability of visual interfaces and designs. Visual designs can be adapted in realtime to a viewer's perceptual abilities. Before a visual design is adapted we need measures of its current perceptual usability

AVI '08, 28-30 May , 2008, Napoli, Italy.

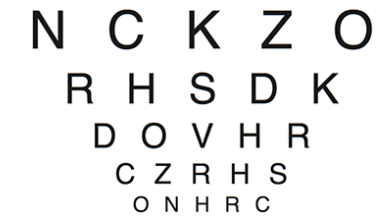


Figure 1: ETDRS chart for measuring visual acuity.

In this paper we present a perceptual usability measure of how easy or hard visual designs are to see when viewed at different distances due to blur. As viewers move away from a display their ability to perceive the display content decreases as a function of distance [7]. Distance is equated with a perceivers ability to perceive visual detail. The measure predicts the relative perceivability of sub-parts of a visual design by using simulations of human visual acuity coupled with an information theoretic measure.

2. RELATED WORK

2.1 Human Visual System Modeling

Human-Computer Interaction researchers have applied specific human vision system (HVS) models to evaluating and developing interfaces [12]. For example Rosenholtz et al's clutter measurement [11] for evaluating information density in interfaces and visualisations. In order to establish how visual designs and interfaces would appear to perceivers we require methods for modeling optical aberrations in the HVS [4, 10]. An optical aberrations model is needed to transform the visual designs into visual patterns people could see. The optical aberration we simulated was optical blur - since we hypothesized it would have a large effect on the perceivability of a visual design.

2.2 Visual Acuity

Spatial visual acuity is the smallest spatial detail that can be visually detected, discriminated, or identified [8]. Studies have experimentally demonstrated that there is a correspondence between a person's visual acuity and their ability to perform everday tasks [13].

Optometrists commonly measure visual acuity by taking psychophysical measures of a person's ability to identify and discriminate optotypes (Figure 1). Optotypes, such as letters, are presented on eye charts, such as the Snellen, Lan-

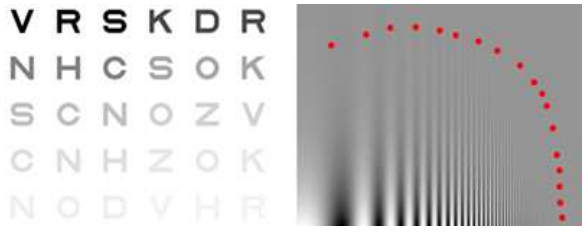


Figure 2: Pelli-Robson (left) and Campbell-Robson (right) Contrast Sensitivity charts. Dots plot Contrast Sensitivity Function.

dolt C, Bailey-Lovie and ETDRS charts [3].

2.3 Contrast Sensitivity

Measurements of spatial contrast sensitivity are another important predictor of a person's ability to see visual detail. Over the years it has been demonstrated that contrast sensitivity plays a very important part in people's abilities to resolve visual detail and carry out everyday tasks [9, 13, 5].

Contrast sensitivity can be measured by using a Pelli-Robson chart (Figure 2, left) or a Campbell-Robson Contrast Sensitivity chart (Figure 2, right). In a Pelli-Robson chart the contrast between the optotypes and the background decreases as a function of optotype distance from the top left of the chart. For further details on measuring Contrast Sensitivity Functions (CSF) please consult Norton et al [8].

3. MEASURING PERCEPTUAL CHANGES

3.1 Algorithm Overview

Our method for predicting the perceptual usability of a visual interface due to a viewer's position works as follows (Figure 3). A static image of the interface undergoes repeat transforms (convolutions). Each transform incrementally blurs the image. At each incremental blur information theoretic measures of the blurred image are calculated. Graphs of the information theoretic measures are then created. Users of the perceptual usability measure can then examine the graphs. The rate of change of the graphed information theoretic measures tells them how different parts of an image change due to the increasing blur. Changes in blur are treated as a function of distance [7]. As distance increases, image blur increases and as distance decreases, image blur decreases.

3.2 Modeling Blur With PSF

To simulate the drop in visual detail due to increasing distance from a display the display contents are incrementally blurred. We use a standard 2D Gaussian function (Eqn. 1) to generate multiple Point Spread Functions (PSF) for use as image convolution filters. The PSF simulates the ambiguity in the path a point of light takes through an aberrant optical system.

$$h_g(x,y) = e^{-(x^2+y^2)/(2\sigma^2)}$$

$$h(x,y) = \frac{h_g(x,y)}{\sum_x \sum_y h_g} \quad (1)$$

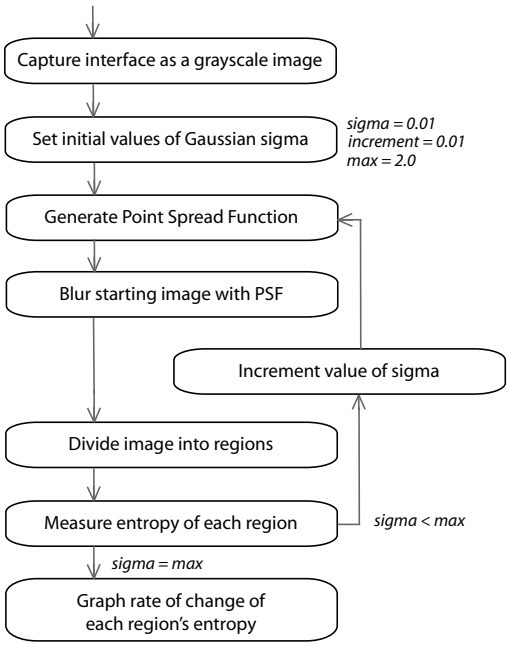


Figure 3: Algorithm measuring blur rate of change.

Slowly increasing the value of the Gaussian filter's σ sigma means the amount of image blur slowly increases. The higher the sigma the more image blur. The rate of change of sigma controls the sensitivity of the perceptual test. Similarly the range of sigma controls how much blur, and indirectly over what distances, we test.

3.3 Measuring Change With Entropy

When considering the eye and HVS as a communication system we hypothesized that elements of Shannon's Information Theory could be a potential measure. Information theory for sensory coding has been researched and applied to vision modeling and statistical image analysis [1]. Initially we used the rate of change of Shannon's entropy over multiple blurs (Eqn. 2) for our analysis, where each unique pixel colour counted as a discrete symbol x_i .

$$\frac{d(-\sum_{i=1}^{n} p(x_i)\log_2 p(x_i))}{d(Blur)} \quad (2)$$

Using Eqn. 2 to analyse the effects of blur in natural images gave what initially seemed meaningful predictions. Unfortunately when using it to analyse images of interfaces there was a problem. The problem arose due to the general structure of the images. Natural images are complex, while interfaces are sparse images. With complex images the entropy tended to decrease due to increasing blur. With sparse images the entropy increased as the blur increased, and eventually the entropy decreased but the point at which it decreased depended on the starting image.

After examining why the entropy in sparse images was increasing we found that it was because there was an increase in the number (n) of colours (x_i). That is the entropy was changing due to a change in the number of unique colours (n) in an image as well as a change in the distribution $p(x_i)$ of the colours. By subjecting entropy to the rate of change of blur we were effectively measuring the entropy of a series of

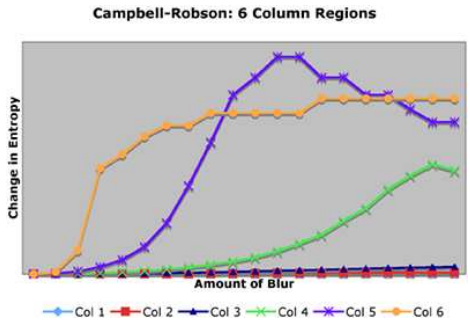


Figure 4: Results Campbell-Robson Contrast Sensitivity Chart. Divide into 6 equal column regions with column 1 starting on the left.

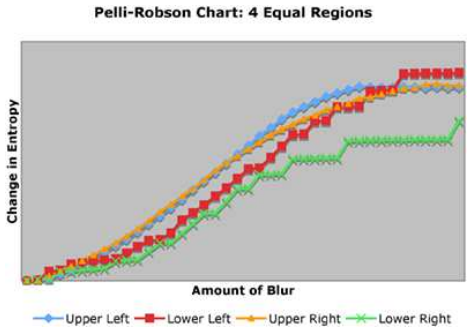


Figure 5: Results Pelli-Robson Contrast Sensitivity Chart. Divided into 4 equal regions.

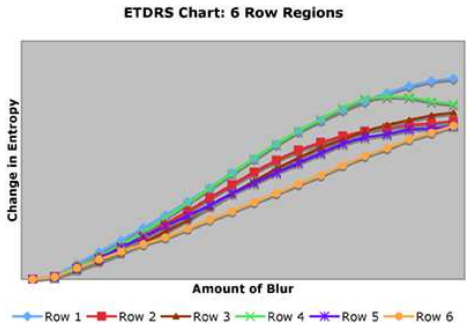


Figure 6: Results ETDRS Chart. Divided into 6 equal row regions with row 1 starting at the top.

Figure 7: Small network graph (left) and webpage (right) that were analysed.

unique communication channels, each of which has its own set of symbols. To eliminate the change in entropy due to the increase and decrease of symbols between blurred images (communication channels) we normalised Shannon's entropy equation:

$$NormEntropy = \frac{-\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)}{n}$$

$$ChangeMeasure = \frac{d(NormEntropy)}{d(Blur)} \quad (3)$$

By dividing Shannon's entropy measure by n, we sought to control the change in the number of colours between images while still allowing for the change in the distribution of colours.

4. EVALUATION

Interfaces and visualisations can visually vary significantly so we sought to establish a ground truth for the performance of our measure. We hypothesized that if our perceptual measure worked it would make predictions consistent with very well established human performance on eye charts [8].

4.1 Results

4.1.1 Campbell-Robson Contrast Sensitivity Chart

Figure 4 shows the results where we tested the Campbell-Robson Contrast Sensitivity Chart by dividing it into 6 column regions of equal width. As you can see from the graph the perceptual measure predicted Col 6 would change the most initially, then Col 5, Col 4, and so on. This result conforms to how people see CSF charts. You can also see that Col 6 stopped changing but the other columns continued changing due to blur. The entropy of Col 6 does begin to decrease, this may be due to normalization working imperfectly (Section 3.3) though it may also be due to how the gray scale colours are quantized into symbols.

4.1.2 Pelli-Robson Chart

We tested the Pelli-Robson Chart (Figure 2, left) by dividing it into four regions of equal size. Figure 5 shows the results. In this case the results are not as clear as with the Campbell-Robson Contrast Sensitivity Chart. Early on in the blur we see that the lower left (red line) and right (green line) regions change fastest, that is they become harder to see quicker. Though the rate of change of the top of the chart quickly surpasses the bottom of the chart. The letters at the top of the chart continued changing for longer at a faster rate because they can undergo a greater amount of blur before becoming indistinguishable blobs. The larger letters are more perceptually robust, while the rate of change of the smaller letters slowed down because the letters had lost so much detail relative to their size.

4.1.3 ETDRS Chart

Figure 6 graphs the results of the perceptual measure evaluating an ETDRS chart (Figure 1). The results are as we would expect, row 1 with the largest letters is the most robust and can change longest while row 6 changes at a slow rate because it has less detail to lose. Of concern is row 4, which appears to perceptually robust - this is an artifact of white space and how the chart was segmented into regions. In future work a smart region segmentation approach will be taken to help avoid such issues.

4.1.4 Small Network Graph

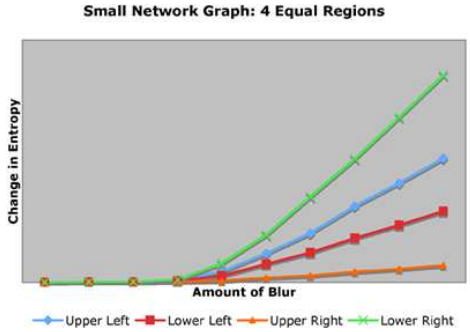


Figure 8: Results of the small network graph analysed with the perceptual measure.

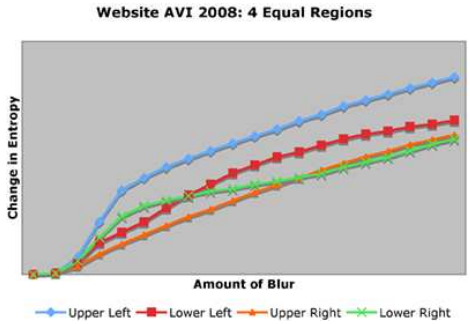


Figure 9: Results of the perceptual measure evaluating the AVI 2008 website.

The small network graph (Figure 7, left) was divided into four equal regions and the perceptual measure was applied. For this analysis the increment value of sigma was set low. As can be seen the results (Figure 8) were as expected. The lower right hand region (green line) changed the most due to blur, and the mostly empty upper right region (orange line) changed the least.

4.1.5 Webpage

Shown in Figure 9 are the results of the perceptual measure evaluating the AVI 2008 front webpage (Figure 7, right). The results are also as expected, the rate at which the lower right hand region (green line) changes quickly decreases as the amount of blur increases. The upper left hand region (blue line) initially looses detail fastest because it has the most detail to loose due to the text and the logo. For a brief while near the midpoint of the blurring the amount of detail lost is faster in the lower left hand region (red line).

5. CONCLUSIONS & FUTURE WORK

In this paper we have presented a first approach to quantifying the perceptibility of a visual design when viewed over different distances. We implemented the perceptual measure and evaluated its performance on a range of eye charts. The results showed the perceptual measure does predict the perceivability of visual designs and, with further research, the accuracy of the perceptual predictions are open to improvement. We also demonstrated the measure predicting the perceivability of a visual design commonly found in graph visualisations, while also providing the results of it analyzing a web page. Further experimental analysis of the perceptual measure outlined in this paper is ongoing. Especially with regards to testing it on a wide range of interfaces.

6. ACKNOWLEDGMENTS

Thanks to the ongoing support from the School of Computer Science and Informatics, University College Dublin, Ireland.

7. REFERENCES

[1] J. J. Atick. Could information theory provide an ecological theory of sensory processing? *Journal of Network: Computation in Neural Systems*, 3:213–251, 1992.

[2] M. Czerwinski, G. Smith, T. Regan, B. Meyers, G. Robertson, and G. Starkweather. Toward characterizing the productivity benefits of very large displays. In *Human-Computer Interaction – INTERACT 2003*, pages 9–16. IOS Press, 2003.

[3] F. Ferris, A. Kassoff, G. Bresnick, and I. Bailey. New visual acuity charts for clinical research. *American Journal of Ophthalmology*, 94:91–96, 1982.

[4] D. D. Garcia. *CWhatUC: Software Tools for Predicting, Visualizing and Simulating Corneal Visual Acuity*. PhD thesis, University of California at Berkeley, 2000.

[5] A. P. Ginsburg, J. Easterly, and D. W. Evans. Contrast sensitivity predicts target detection field performance of pilots. In *Proceedings of Human Factors Society*, pages 269–273, October 1983.

[6] F. Guimbretière, M. Stone, and T. Winograd. Fluid interaction with high-resolution wall-size displays. In *ACM Symposium on User Interface Software and Technology (UIST)*, pages 21–30, 2001.

[7] G. Heron, H. P. Furby, R. J. Walker, C. S. Lane, and O. J. E. Judge. Relationship between visual acuity and observation distance. *Journal of Ophthalmic and Physiological Optics*, 15(1):23–30, 1995.

[8] T. T. Norton, D. A. Corliss, and J. E. Bailey. *The Psychophysical Measurement of Visual Function*. Butterworth Heinemann, 2002.

[9] C. Owsley and M. E. Sloane. Contrast sensitivity, acuity, and the perception of 'real-world' targets. *British Journal of Ophthalmology*, 71:791–796, 1987.

[10] E. Peli. Test of a model of foveal vision by using simulations. *Journal of the Optical Society of America A*, 13(6):1131–1138, June 1996.

[11] R. Rosenholtz, Y. Li, J. Mansfield, and Z. Jin. Feature congestion: A measure of display clutter. In *Proceedings of SIGCHI conference on human factors in computing systems*, pages 761–770, 2005.

[12] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2nd edition, 2004.

[13] S. K. West, G. S. Rubin, A. T. Broman, B. Muñoz, K. Bandeen-Roche, and K. Turano. How does visual impairment affect performance on tasks of everyday life? *Archives of Ophthalmology*, 120:774–780, June 2002.

[14] D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan. Perception of elementary graphical elements in tabletop and multi-surface environments. In *Proceedings of the 2007 SIGCHI conference on human factors in computing systems*, pages 473–482, 2007.

Illustrative Halos in Information Visualization

Martin Luboschik
Institute for Computer Science
University of Rostock
18051 Rostock, Germany
luboschik@informatik.uni-rostock.de

Heidrun Schumann
Institute for Computer Science
University of Rostock
18051 Rostock, Germany
schumann@informatik.uni-rostock.de

ABSTRACT

In many interactive scenarios, the fast recognition and localization of crucial information is very important to effectively perform a task. However, in information visualization the visualization of permanently growing large data volumes often leads to a simultaneously growing amount of presented graphical primitives. Besides the fundamental problem of limited screen space, the effective localization of single or multiple items of interest by a user becomes more and more difficult. Therefore, different approaches have been developed to emphasize those items – mainly by manipulating the items size, by suppressing the whole context or by adding supplemental visual elements (e.g., contours, arrows). This paper introduces the well known illustrative technique of *haloing* to information visualization to address the localization problem. Applying halos emphasizes items without a manipulation of size or an introduction of additional visual elements and reduces the context suppression to a locally defined region. This paper also presents the results of a first user-study to get an impression of the usefulness of halos for a faster recognition.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*evaluation*; H.5.0 [**Information Interfaces and Presentation**]: General

General Terms

Design, experimentation, human factors, verification.

Keywords

Illustrative rendering, illustrative visualization, halos, information visualization.

1. MOTIVATION

An important problem in interactive scenarios is the fast localization of relevant information to effectively perform a task. Especially in cases where large amounts of information are scattered loosely on the screen, the search time may increase immensely if no localization supporting system is provided.

In information visualization – where interaction is a principal component – the visualization of permanently growing data volumes is a general aim. In many cases, this visualization results in an increased amount of visual primitives. An effective localization of interesting items becomes hard to achieve without any support. Therefore, different methods have been developed to emphasize interesting items. Accentuation techniques like the magnification of the interesting item [3] or indicative contours or arrows are explicitly emphasizing the wanted information. Suppression techniques like desaturation, transparency or hiding of unimportant items [5] reduce the visual presence of distracting visual primitives and therewith implicitly emphasize the remaining. In [11] a brief overview over several emphasizing approaches in medical visualizations is given. However, the existing strategies bear constraints: Both, accentuation and suppression, are endangered to occlude, distort or simply hide surrounding context information – that may be crucial for the interpretation of the interesting information – if applied inconsiderately.

Although the illustrative technique of haloing is a wide spread tool in graphical applications (Sect. 2), it is rarely considered in information visualization. Therefore, this paper analyzes the usefulness of halos to accelerate the localization in interactive scenarios like information visualization scenarios. We use a simple halo-generator (Sect. 3) to produce haloed images that are also used in a user study. The result and the design of this study are presented in Section 4. The results of our work are discussed in Section 5.

2. RELATED WORK / BACKGROUND

Graphical halos originate from artistic drawings to highlight important persons and to enhance the visual separation of objects from the background. This is done by drawing the surrounding background locally with brighter or darker colors. In that way, the contrast is locally adapted and therewith halos support the human viewing process [10]. The spectrum of halo effects ranges from thin gaps around the highlighted object through thick opaque outlines up to nearly undistinguishable darkening or brightening effects.

Today halos are used in different domains. For example in medical illustrations halos are used to enhance the perception of depth and the distinction of different structures. Moreover, many approaches have been developed in the field

AVI'08 , 28-30 May, 2008, Napoli, Italy

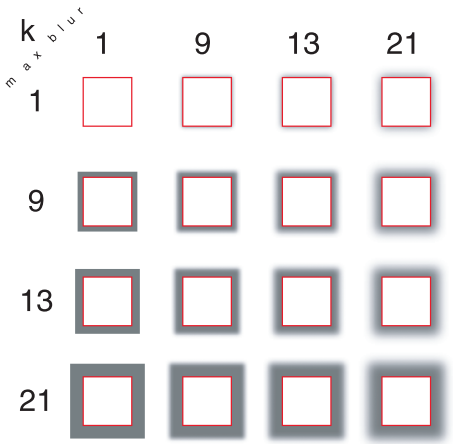


Figure 1: Black halos with different k_{max}, k_{blur} and a transparency of 50%.

of 3D visualization and 3D rendering to achieve an enhanced depth perception. Very early a first method has been developed to improve the perception of overlapping lines in 3D [1]. Halos are also a wide spread approach to distinguish different structures in the field of flow and volume visualization. For example in [7][8] they are used to visually separate different stream lines within a 3D flow visualization. Approaches from volume visualization [4][13][2] and 3D rendering [9] often result in interactive GPU-based methods. The use of halo profile functions in [2] additionally allows an effective definition and manipulation of halos.

The fast access to import information in complex information spaces is a main aspect in many efficient work scenarios. Therefore, several techniques have been successfully developed to emphasize screen elements. However, these techniques (see Sect. 1) manipulate the presentation of visual attributes and therewith may hinder an effective interpretation of data values. Other visual clues may occlude important information. Moreover, some of the existing techniques are simply not applicable in every scenario. Contour lines for example cannot be applied if the highlighted object itself is a thin color encoded line due to misinterpretation risks. Applied to pixel oriented visualization techniques they would occlude information.

The examples from flow and volume visualization show that halos are widely used for a better distinction of *all* different structures [7][8][4]. In some cases, they are already used for *accentuation of single objects* [2]. However, up to now halos are not sufficiently considered in information visualization. A first approach that uses halos in the field of *2D information visualization* is presented in [14]: Thin outlines are used to enhance the perception of line based visualizations (parallel coordinates, network visualization). For this reason, this paper analyzes the benefit in the field of 2D information visualization – especially for localization tasks. The presented study verifies the usefulness of halos.

3. HALO GENERATION

A common approach to generate halos is the use of a seed image combined with a low-pass filtering that blurs the edges. The result is a so called halo field image (see [2]) that

Figure 2: The halo generation (from left to right): Seed image, enlargement, blurring and compositing.

is used in the final halo rendering. The naive application of a low-pass convolution to the seed image may blur out halos of small but maybe important visual primitives. In [2] this problem is solved by a multiple use of low-pass filters followed by a recombination with the original seed image.

To generate a wide range of halo effects with low efforts, we use a simple but flexible approach that extends the common one. It uses an additional maximum filtering step to prevent disappearing halos around small primitives and to provide a big variety of halos. The overall halo generation is summarized as follows (see Fig. 2):

1. The object to be haloed is rendered offscreen as a seed image whereas its color values represent the values of the later halo field.
2. A 2-dimensional maximum convolution of free kernel size (k_{max}) is applied to the seed image. A quadratic filter kernel with $k_{max} \geq 2$ enlarges the seed image.
3. A low-pass filter of free kernel size (k_{blur}) is applied to the result of step 2 to get a soft-edged halo. We use quadratic filter kernels with Gaussian weights.
4. The gained halo field is used to render a halo that is combined with the remaining visualization.

The free combination of k_{max} and k_{blur} allows the generation of outlines ($k_{max} \geq 2$,$k_{blur} = 1$), the prevention of disappearing halos ($k_{max} = k_{blur}$) or a simple blurring of the seed image ($k_{max} = 1$,$k_{blur} \geq 2$). Figure 1 shows the influence of different k_{max} and k_{blur} to the halo appearance.

The first three steps result in a halo field image containing values that may be interpreted in different ways. A common way is to interpret the halo field as alpha values of a colored halo. More complex halo profile functions are also possible (see [2]). The transparency (values of the seed image) and the color of the halo become additional parameters which can be used during application. These parameters control the conspicuousness of the generated halo as they influence the appearance (color) and contrast to the background (color and transparency).

The wide range of different halo effects described here, enables the support of different localization tasks. *Directed search* for example, means the fast localization of one or a small set of visual primitives as the result of a concrete query. In many cases of information visualization this is important for efficient task performance. For example one or more items of interest shall be compared to the whole information space and should therefore be located fast. For example color coding is used in Figure 3 to visualize health data of ≈230 districts of northern Germany in January 2000. The number of respiratory infections recorded by a health insurance is visualized. Since color coding is good for perceiving the global value distribution and maximum values,

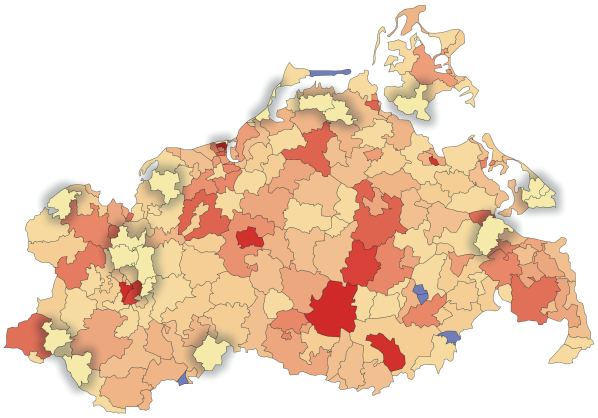


Figure 3: Color coded respiratory infections in northern Germany. Highlighting halos are applied to districts with minimum values. These areas are easy to perceive, although color is more prominent.

it is hard to distinguish lower values. Therefore halos are used to highlight minima. Although the red areas are more prominent, the districts with lowest values are easy to perceive.

Additionally, halos may also support the *undirected search*. In this case, a data set is explored without any concrete query. Thus, halos should not be used preeminent to highlight single items of interest but in a subliminal way to support the general exploration. For example, statistical features concerning the data may be mapped onto halo parameters to control the items visual prominence.

4. USER STUDY

Since halo outlines are nearly equal to contours (e.g., see [12] [14]), this paper considers only halos with a blurred edge ($k_{blur} > 1$). Moreover, we concentrate on semitransparent monochrome halos to investigate the subliminal use that is often found in artistic drawings. This kind of halo can be seen as a hybrid of accentuation and suppression as it suppresses locally the background (by the transparency property) and accentuates the haloed object (colored opaque property). To verify and quantify the usefulness of such halos in localization tasks, we developed and realized a small uncontrolled user study. The aim of our study was to get a first impression of halo parametrization for an acceleration of the *directed search* on different cluttered backgrounds.

Currently eight female and twelve male participants took part in our study. Their age was 20 to 36 and the education varies from secondary school up to PhD students. Only three persons were educated in computer science.

Design The task that had to be performed by the different participants was a simple search for one given icon within a larger group of 49 shown icons. These were randomly chosen from one of two different sets of 120 icons (color icons and polygon icons). For each set we used three different backgrounds (no, medium, hard cluttered) and four different halos (none, small, medium, large). Colors (in *Lab* color space, see [6]) and icon sizes (const.) were chosen carefully to achieve nearly constant visual stimuli.

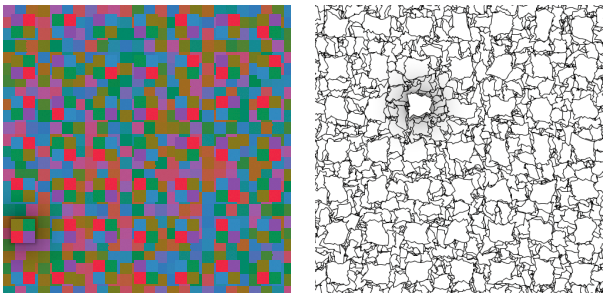


Figure 4: Two example screens of our study: left) a color icon scenario, right) a polygon icon scenario. In both cases, the most difficult background and largest halos were applied.

Procedure. This study has been implemented using a web page (*http:// www.informatik.uni-rostock.de/∼malub/ study.html*) to simply address a larger group of participants. To prevent the development of a fixed search strategy, every presented screen has been generated completely randomly (equal distributed pseudo random): the used icon set, the selection of 49 icons, the used background, the used background orientation and the used halo. Participants of the study were briefed by a small text (German) and a monitor calibration image. Afterwards they were exposed to 100 screens containing the randomly generated image and one icon to be found. Clicking the located item loads the next screen and records localization time. Image generation parameters are the only additional recorded values.

Analysis and Results To get a first impression of halo parameterizations and the influence of different backgrounds, a straight forward analysis has been performed: Wrong answers were disregarded and extreme outliers (3∗IQR) were eliminated per participant to get a meaningful average. The overall average suggests that halos are generally able to accelerate the localization process.

Figure 5 shows the average answer times with both kinds of icons on different backgrounds with different halos applied. In the color icon scenario (see Fig. 4 left) we expected the most difficult background to provoke the highest answering times in the unsupported scenario. Instead the medium cluttered background did. This and the seeming faster feedback concerning the medium halo on the medium background compared to the largest halo, may be ascribable to the fact of few participants and the uncontrolled environment. However, the diagram shows that the perception of a halo seems to depend on the halos size and the background: Whereas every halo is hard to perceive at a white background, they seem to be well recognizable at the noisy backgrounds. This may be explained by the halo-introduced increased contrast on a colored background resulting in a better separation. Meanwhile, the halos decrease the contrast on a white background, due to their grey appearance. Although the results of the polygon icons are similar to the color icons, some differences can be found. The most important is the background dependance: The localization task becomes more and more difficult with an increasingly cluttered background (see Fig. 4 right) – even with applied halos. Whereas the contrast is increased in the color icon scenario, now the contrast decreases due to the overlapping of black background lines and a black halo. Therewith the

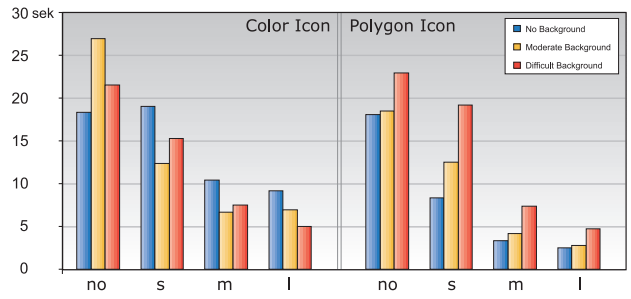


Figure 5: Average localization times achieved in a directed search for a single icon. Different halo sizes (no,s,m,l) were applied on different backgrounds.

halos disappear. However, the applied halos reduce the localization time corresponding to every background.

Due to the low number of participants and their strongly varying localization times, no heavy analysis has been executed on acquired data. Besides the decreasing average answering times, a continuous reduction of standard deviation with increasing halo sizes has been observed. The bigger fluctuation found in the color icon scenario may be due to unknown perception malfunctions of the participants and different lighting conditions.

5. CONCLUSION

Our approach of using halos in the field of 2D information visualizations describes an additional way to emphasize items of interest. Many of the several existing accentuation methods to are restricted to tight constraints and therewith hinder an overlapping use of one approach in several visualization techniques. The wide range of parametrization possibilities makes halos very flexible and so they may be adapted to different tasks and visualization techniques. Thereby, the halo effect can be modified continuously from a hard edged opaque outline to nearly invisible soft halos.

The major benefit of halos in information visualization is the naturally provided parameter of transparency. Therewith, halos generally do not occlude context but modify it locally to enhance the perception of selected items. In contrast to known suppression techniques, the global overview is preserved. In Figure 3 for example, the minimum values are perceived at first glance without any occlusion or graphical distortion. Hence, halos support the exploration of large data volumes and represent an non-opaque alternative to distortion and information hiding techniques. Especially the fast localization, that is often needed in efficient work scenarios, is supported. Moreover, even data values may be mapped on halo parameters.

Since maximum filter and Gaussian blur are separable filters, they can be used each as a combination of two 1D filters. Thus, convolution complexity changes from $O(n^2)$ to $O(2n)$ enabling time critical applications.

Although the implemented study showed that halos are helpful in localization tasks, it also showed that the efficiency of halos strongly depends on the given background, color and transparency. Therewith, similar problems appear that are known from accentuation techniques like contours. Both, color and transparency have to be selected carefully to guarantee an accentuation affect. Different alternatives like a local desaturation of the background using the halo field may cope with that problem.

Therefore, further investigations will examine the use of this compositing alternatives to achieve high contrasts independently of the background. Moreover, further user studies are reasonable to get a more quantitative assessment of halo sizes and to analyze the contribution of transparencies and colors. Different visualization tasks should be investigated. Finally, a direct comparison to contours and other techniques may reveal the limits of our approach.

6. REFERENCES

[1] A. Appel, F. J. Rohlf, and A. J. Stein. The haloed line effect for hidden line elimination. In *Proceedings of ACM SIGGRAPH'79*, 1979.

[2] S. Bruckner and E. Gröller. Enhancing depth-perception with flexible volumetric halos. In *Proceedings of IEEE Visualization (Vis'07)*, 2007.

[3] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Distortion viewing techniques for 3-dimensional data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'96)*, 1996.

[4] D. Ebert and P. Rheingans. Volume illustration: Non-photorealistic rendering of volume models. In *Proceedings of IEEE Visualization (VisŠ00)*, 2000.

[5] G. W. Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'86)*, 1986.

[6] H. Hagh-Shenas, S. Kim, V. Interrante, and C. Healey. Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. In *Proceedings of IEEE Information Visualization (InfoVis'07)*, 2007.

[7] V. Interrante and C. Grosch. Strategies for effectively visualizing 3d flow with volume lic. In *Proceedings of IEEE Visualization (Vis'97)*, 1997.

[8] L. Li and H.-W. Shen. Image-based streamline generation and rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):630–640, 2007.

[9] J. Loviscach. Stylized haloed outlines on the gpu. ACM SIGGRAPH '04 Poster, 2004.

[10] T. Luft, C. Colditz, and O. Deussen. Image enhancement by unsharp masking the depth buffer. In *Proceeding of ACM SIGGRAPH'06*, 2006.

[11] B. Preim, C. Tietjen, and C. Doerge. Npr, focussing and emphasis in medical visualizations. In *Proceedings of Simulation und Visualisierung*, 2005.

[12] C. Stoll, S. Gumhold, and H.-P. Seidel. Visualization with stylized line primitives. In *Proceedings of IEEE Visualization (Vis'05)*, 2005.

[13] N. A. Svakhine and D. S. Ebert. Interactive volume illustration and feature halos. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications (PG'03)*, 2003.

[14] C. Waters and T. J. Jankun-Kelly. Illustrative rendering for information visualization. IEEE InfoVis'06 Poster, Oct. 2006.

Shadow Tracking on Multi-Touch Tables

Florian Echtler Manuel Huber Gudrun Klinker, PhD
{echtler,huberma,klinker}@in.tum.de

Technische Universität München - Institut für Informatik
Boltzmannstr. 3, D-85747 Garching, Germany

ABSTRACT

Multi-touch interfaces have been a focus of research in recent years, resulting in development of various innovative UI concepts. Support for existing WIMP interfaces, however, should not be overlooked. Although several approaches exist, there is still room for improvement, particularly regarding implementation of the "hover" state, commonly used in mouse-based interfaces.
In this paper, we present a multi-touch system which is designed to address this problem. A multi-touch table based on FTIR (frustrated total internal reflection) is extended with a ceiling-mounted light source to create shadows of hands and arms. By tracking these shadows with the rear-mounted camera which is already present in the FTIR setup, users can control multiple cursors without touching the table and trigger a "click" event by tapping the surface with any finger of the corresponding hand.
An informal evaluation with 15 subjects found an improvement in accuracy when compared to an unaugmented touch screen.

Categories and Subject Descriptors

H.5.2 [**Information interfaces and presentation**]: User Interfaces—*Input devices and strategies (e.g., mouse, touchscreen)*

Keywords

direct-touch, mouse emulation, tabletop interfaces, shadow tracking, multi-touch, FTIR

1. INTRODUCTION

In recent years, multi-touch capable input systems have increasingly been a focus of research in the UI community. Most of these systems can be categorized as direct-touch input devices, allowing the user to manipulate data objects directly on the display. In contrast, however, a large majority of input devices in everyday use are still indirect-touch devices like laptop touchpads. Moreover, like the plain computer mouse from which they have evolved, they support only a single point of interaction. Owing to this omnipresence of mice and mouse-like devices, a great percentage of existing and emerging software is based on established UI paradigms like, e.g., WIMP interfaces.
Ideally, all software that is to be used on a multi-touch system should natively provide support for these new input possibilities. However, any current setup will likely run a mix of multitouch and legacy WIMP applications, e.g. a web browser. To provide easy access to this large existing software base, a multi-touch system should offer backwards compatibility to mouse-based applications, while at the same time providing the full range of input data to multitouch-enabled software.

AVI '08, 28-30 May , 2008, Napoli, Italy.

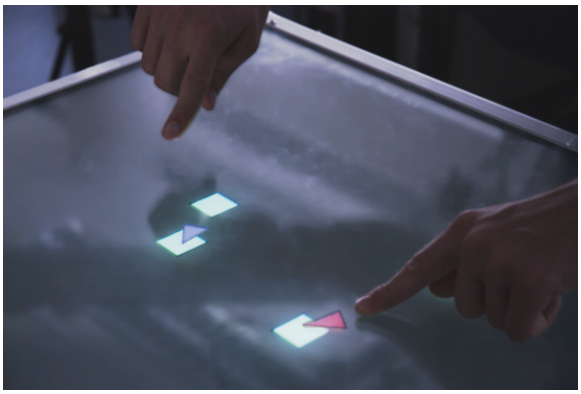


Figure 1: User controlling two cursors that are independent in position and orientation. The right hand's cursor is in the "hover" state (blue), while the left hand's cursor was switched to "click" state (red) by touching the surface.

One of the challenges posed by this goal is that most current multi-touch systems provide less data on certain aspects of the users' actions than a mouse does. These systems usually report only one kind of interaction, a touch of the surface, which is generally interpreted to have the same function as a button click. With a mouse, on the other hand, it is possible to interact by only moving the pointer on top of an object without clicking, a technique known as hovering. Another problem is inherent to the concept of direct touch. When touching the surface, the finger itself typically occludes dozens of pixels. Particularly when aiming for small targets like, e.g., window handles, this greatly reduces the accuracy with which such an interface can be operated.

In this paper, we present a multi-touch system which is designed to address these problems. A multi-touch table based on FTIR (frustrated total internal reflection) is extended with an additional infrared light source mounted at the ceiling. This light source causes hands and arms to cast clearly defined shadows on the table surface. The rear-mounted infrared camera that is already available in the FTIR system can be used to track these shadows to provide proximity and orientation data for each hand. Using this data, the system can provide an independent pointer for each user's hand that can be moved without touching the table. Subsequently, a "click"

event can be triggered by touching the surface with any finger of the corresponding hand (see Figure 1). We have conducted an informal user evaluation with 15 subjects regarding the accuracy of our system and found it improved over an unaugmented touch screen.

2. RELATED WORK

The subject of how to accurately control a pointer on a direct-touch screen has already been investigated. In this section, we will look at existing approaches to enable direct-touch support for mouse-based applications, especially with respect to support for the fundamental UI states ("tracking/hover" and "dragging/click") noted by Buxton et al. [8].

Esenther et al. [9] present a solution which is based on the DiamondTouch [3] interaction surface and takes advantage of the multi-touch capability. Their *Fluid DTMouse* requires the user to touch the surface with two fingers simultaneously to control the mouse cursor, which is placed in the middle between the two contact points. Tapping with a third finger triggers a click. This technique allows to distinguish between hovering and clicking and also solves the occlusion problem. However, while effective, these methods might not be as intuitive to the first-time user as a simple "point-and-tap" interface. For any kind of interaction, they require the users to touch the surface while moving their fingers, which may grow tiresome over time.

In a similar approach, Benko et al. [11] present several different dual-finger interaction techniques that allow the user to control the cursor with one hand while slowing down or freezing it with the second hand to enable accurate positioning. They also present a technique called *SimPress* which distinguishes between "hover" and "touch" states by analyzing the shape of the finger contact area. In an extensive user study, they found significant advantages over a plain touchscreen.

A different solution is offered by the SMARTBoard [1]. Their system provides a dedicated "hover" button to switch into a state in which the user can move the pointer without triggering further interaction. Frequent mode switches are likely to be time-consuming, however, as they require the user to direct her or his attention to a completely different part of the UI.

Three other direct-touch systems which should be mentioned here are Wilson's TouchLight [10] and PlayAnywhere [6] as well as Rekimoto's SmartSkin [12]. Although all three systems provide proximity data about users' hands, this feature has seen little use regarding pointer control. A possible reason is that they do not provide unambiguous differentiation between "touch" and "hover" states but instead have to rely on a threshold value. Users are therefore likely to trigger a click event before they actually touch the surface, which may be confusing.

Malik et al. [5] present a system which does not provide direct interaction, but offers a gesture-sensitive touchpad instead. Although it cannot deal with multiple users, the system offers two-handed input for a single user. It relies on a calibrated stereo camera, however, thereby significantly increasing the complexity of the system.

Han has presented an FTIR-based [7] direct-touch system which has contributed to the large research interest in multi-touch due to its easy construction from common off-the-shelf components, low price and back-projection support.

As our system is based on FTIR, we will include a more detailed description here. FTIR works by guiding infrared light through an acrylic glass sheet placed in front of a projection surface. The light rays are subject to total reflection at the air-material interface and are transported through the plate similar to an optical fiber. If a soft, dense material like skin touches the surface, however, total reflection is interrupted and the light illuminates the contacting object, which is now visible on the back side as a bright spot (see Figure 2). As mentioned by Han, this setup also does not provide proximity data.

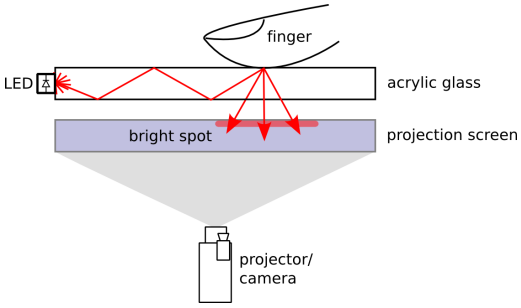


Figure 2: FTIR principle

3. TISCH[1] SYSTEM DESCRIPTION

We share the opinion that a multi-touch system can greatly benefit from supporting common mouse-based software. Our solution to the problems mentioned above is an extension to an FTIR-based interactive table that allows the user(s) to control an independent pointer with each hand. It is not necessary to keep contact with the surface for this kind of interaction, moving the hand above the table is sufficient. The surface only has to be touched if a click event is meant to be triggered.

The central element of our system is a multi-touch table (TISCH[1]) that provides room for 4 to 6 concurrent users.

A frosted glass plate of about 1.10 x 0.7 m is used as a back-projection surface. It is mounted on a robust aluminium frame which contains a projector, an infrared camera and a computer. An acrylic glass sheet placed on top of the projection area has 70 infrared LEDs attached around its rim to provide multi-touch input to the computer via an IR camera.

To gather proximity information, our goal was to create distinct shadows of objects on and above the surface. We have therefore mounted an additional infrared light source at the ceiling above the table. While this increases the complexity of the system and reduces its mobility, interactive tables tend to be stationary equipment which could be integrated into existing conference room tables or placed in a public area as an information booth. An additional overhead light should therefore be easy to add to such a setup.

As the illuminated areas of the surface should not interfere with the bright spots from the FTIR system, the top light source and the side-mounted LEDs are switched on alternatingly for odd- and even-numbered camera frames, thereby providing two consecutive images which will be referred to as shadow and contact image (see Figure 4). As this reduces the effective frame rate by a factor of two, a fast camera of at least 60 FPS should be used to provide a smooth user experience. To ensure accurate synchronization with the camera, a small circuit based on a PIC18 microcontroller activates the two light sources in turn and supplies them with a pulsed control current to increase total light output.

Building the top light source presented some unexpected challenges. To create hard shadows, a point light source is the ideal choice. We therefore evaluated a point light source first, consisting of a cluster of 16 infrared LEDs at different angles to be suspended over the center of the table. Unfortunately, this setup failed to illuminate more than a small fraction of the surface directly below the light source.

The reason for this effect is that light from above has to pass a total of four material-air interfaces (see also Figure 2). If each layer re-

[1]**T**angible **I**nteraction **S**urface for **C**ollaboration between **H**umans

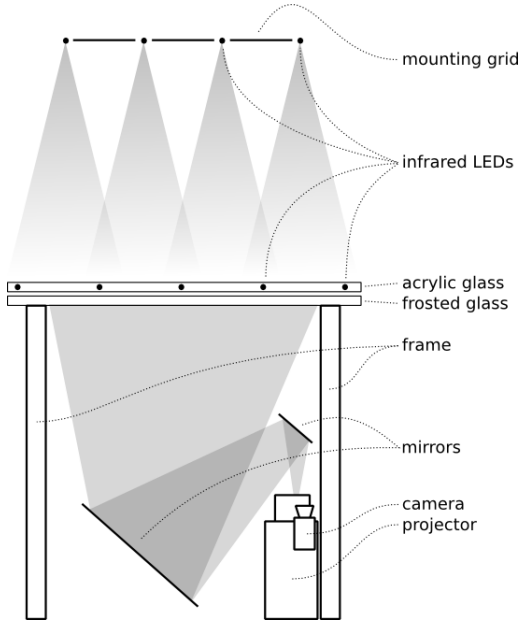


Figure 3: hardware setup

flects 15% of incoming light (a conservative assumption), the total intensity arriving at the camera already drops to $(1-0.15)^4 \approx 52\%$ of emitted light. The reflected percentage increases with decreasing angle of incidence according to Fresnel's equations. Below the critical angle of about 41°, the light transmission even drops to zero because total reflection takes effect and all light is captured in the topmost plate.

As a single point light source proved insufficient to illuminate the entire surface, we switched to a regular grid of 28 LEDs. Each LED is oriented straight downwards and mounted at a distance of 25 cm to the others. Considering the LEDs' beam width of about 20°, their intensity falloff and a distance of 1.50 m between surface and light source, the overlapping spotlights provide a sufficiently uniform illumination of the table with easily discernible shadows. For a schematic of the entire hardware setup, see Figure 3.

On the software side, we aimed for a clean separation between the input system and the end user application.

The raw camera images are acquired and analyzed in a background process. After a shadow/contact image pair has been read from the camera, both images are segmented into disjoint blobs by background subtraction, thresholding and erosion for noise removal. Size (pixel count), centroid, major and minor axes and outermost points along the major axis are now calculated for each remaining blob which is larger than the minimum size.

In a second step, illustrated in Figure 4, every touch point from the contact image is associated with its nearest shadow. Ambiguities between closely spaced touch points can now be resolved. After all data for the different blobs has been calculated, the positions are transformed by a homography [4] in a final step to compensate for the projective distortion between projector and camera image. This homography is calculated separately with a calibration tool using four point correspondences that are gathered by tapping four crosshairs in the screen corners. The transformed data for each blob is finally sent to the application(s) as a UDP packet for each processed image pair.

As a proof-of-concept and as a basis for the informal evaluation described in the next section, we use an application that displays an

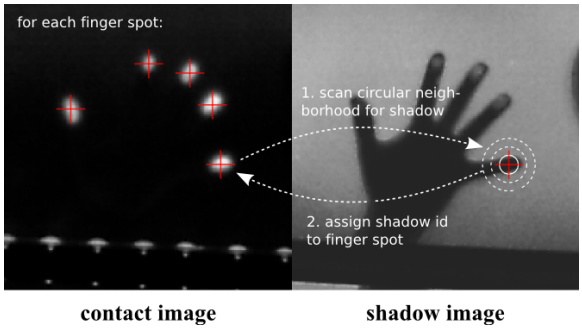


Figure 4: shadow processing

arrow-shaped cursor for every detected shadow (see also Figure 1). This cursor is located near the peak of each shadow, but is shifted by an additional offset along the shadow's major axis. This prevents the user's hand from occluding the cursor. As long as the hand is not in contact with the table, the cursor is in the "hover" state. If the surface is touched with any finger of the associated hand, a "click" event is triggered at the location of the cursor tip. As an additional feedback to the user, the cursor changes color from blue to red. This state is maintained as long as one or more fingers from this specific hand touch the surface. When all fingers have been lifted off again, the cursor reverts to the "hover" state. For a short video demonstrating use of our system, see [2].

4. PRELIMINARY EVALUATION

For a first evaluation of our setup, we conducted an informal targeting test. Our goal was to verify whether our system, despite its prototype state, would be able to provide increased targeting accuracy through pointer feedback as noted in other publications, e.g. by Benko et al. [11].

The targeting application requires users to activate a single randomly positioned square target on the screen. After activation, the target is replaced by a new one in a different randomly chosen location and the task is repeated. For each of the 20 repetitions, the time between two successful target activations and the distance from the target center to the touch point is recorded. The target has a size of 30 x 30 pixels, which is equivalent to a physical size of about 1 square inch on our screen.

Two different test modalities were used. In the first one, no cursors were available and the target had to be directly touched with any finger to activate it. The centroid of the contact spot was used as touch point.

In the second test, each hand held over the table surface was augmented with a cursor as described above. In this case, the touch point was at the cursor tip.

Our 15 computer-literate test subjects (4 women, 11 men, average age 26 years) had little prior experience with touch-screens. All users were told to hit the targets as fast and accurately (that is, close to the center) as possible, and in the second test, to use the hand-controlled cursor as they would use a mouse cursor.

The results of our test confirmed our expectations. Despite noticeable jitter in the cursor position, the accuracy increased by about 4 pixels: the average distance from the target center was 11.7 pixels for the first test, compared with 7.5 pixels for the second test. However, the test also exposed a drawback of our system: the time which users took to hit a target increased by a factor of two from 1.3 seconds in the first test to 2.6 seconds in the second test. Again,

this can be attributed to cursor jitter which caused users to hesitate before tapping the target.
A notable observation during our tests was that users intuitively took advantage of the cursor's variable orientation, particularly when reaching for targets close to the table edge. In this case, most subjects oriented their hand parallel to the table edge so that the cursor now pointed perpendicular to the subjects' viewing direction instead of outwards, thereby preventing occlusion.

5. CONCLUSION AND FUTURE WORK

Our system provides a useful addition to existing FTIR-based multi-touch setups in order to provide intuitive mouse emulation, including support for the "hover" state and precise targeting. Although the required hardware slightly increases in complexity (additional ceiling-mounted infrared light and control circuit), still only a single camera and calibration is required.

Although our evaluation was not rigidly controlled, some conclusions can be drawn. Our technique is able to provide a noticeable increase in pointing accuracy at the expense of targeting speed. Accuracy as well as usage speed are likely to increase further when the cursor motion jitter is reduced. To this end, a Kalman filter can be employed to make cursor position and orientation less sensitive to camera noise.

Moreover, the system is highly intuitive. Most users understood the system immediately after first holding their hand over the table surface and observing the associated cursor. This is an advantage that should not be underestimated, as interactive surfaces are often deployed in public or semi-public scenarios where little or no prior instruction is available to users.
As the overhead light source currently requires a (semi-)permanent installation, an application like a public information booth is one of the best suited scenarios for our system. Here, a mix of legacy software (e.g. a web browser) and multi-touch applications (e.g. casual games) is likely to be used. While conventional applications can then be controlled with one or more pointers as usual, multi-touch software can be used with direct touch interaction.
However, if a mobile solution is absolutely mandatory, the light source could be attached directly to the table like a canopy. While the unavoidable poles might hinder users, a different solution which relies entirely on environment light could be envisioned. In this scenario, an existing ceiling lamp could provide the necessary illumination.

Another aspect of this system is that it provides support for tangible user interfaces [13]. Objects on the surface create shadows, but no contact spots (with the exception of some very soft plastics). These shadows could be classified according to their size and ratio between major and minor axis, thereby providing, e.g., physical handles for widgets.

Finally, we have not yet explored the possibilities which are offered by our system's ability to assign contact spots to a certain hand. Recent laptop touchpads like those installed in MacBooks allow the user to tap with one, two or three fingers simultaneously to perform a left click, right click or scrolling operation. While other multi-touch surfaces would not be able to distinguish such a gesture from closely spaced gestures with two hands, our setup can easily be extended to support such interactions. This could also improve intuitive usability, as laptop users are probably already well accustomed to these gestures.

6. REFERENCES

[1] Smart Technologies. SMART Board. `http://www.smarttech.com/SmartBoard`.

[2] F. Echtler. Shadow tracking demonstration video. `http://campar.in.tum.de/personal/echtler/avi08-shadowtrack.avi`.

[3] P. Dietz and D. Leigh. DiamondTouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226, New York, NY, USA, 2001. ACM Press.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[5] S. Malik and J. Laszlo. Visual touchpad: a two-handed gestural input device. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 289–296, New York, NY, USA, 2004. ACM Press.

[6] A. Wilson. PlayAnywhere: a compact interactive tabletop projection-vision system. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 83–92, 2005.

[7] J. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, New York, NY, USA, 2005. ACM Press.

[8] W. Buxton, R. Hill, and P. Rowley. Issues and techniques in touch-sensitive tablet input. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 215–224, New York, NY, USA, 1985. ACM.

[9] A. Esenther and K. Ryall. Fluid DTMouse: better mouse support for touch-based interactions. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 112–115, New York, NY, USA, 2006. ACM Press.

[10] A. Wilson. TouchLight: an imaging touch screen and display for gesture-based interaction. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76, New York, NY, USA, 2004. ACM Press.

[11] H. Benko, A. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272, New York, NY, USA, 2006. ACM Press.

[12] J. Rekimoto. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120, New York, NY, USA, 2002. ACM Press.

[13] H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the Conference on Human Factors in Computing Systems*, pages 234–241, 1997.

LocaweRoute: an advanced route history visualization for mobile devices

Taina M. Lehtimäki[1], Timo Partala[1], Mika Luimula[2], Pertti Verronen[2]

[1] University of Oulu, Oulu Southern institute
RFMedia Laboratory
Vierimaantie 5
FIN-84100 Ylivieska, Finland
{taina.lehtimaki, timo.partala}@oulu.fi

[2] CENTRIA Research and Development
RFMedia Laboratory
Vierimaantie 5
FIN-84100 Ylivieska, Finland
{mika.luimula, pertti.verronen}@centria.fi

ABSTRACT
In this research, we addressed the problem of visualizing route histories on a mobile device. We developed a solution, which combines the visualization of three route history parameters: speed, direction, and location. The visualization was tested in a laboratory evaluation with 12 subjects. The results showed that by using the visualization the subjects were able to estimate actual driving speeds accurately. The subjects also evaluated that the visualization supported their knowledge of the speed, location, and direction quite well. The results suggest that the presented visualization is an improvement over currently used route history visualizations.

Author Keywords
Visualization, route history, mobile device

ACM Classification Keywords
H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

INTRODUCTION
The use of different tracking and navigation devices is growing rapidly. Different personal navigators have already become very popular and they are increasingly used especially in car navigation. In addition, professional solutions based on location data are becoming more and more typical in various fields from global logistics solutions to health care.

Current location-based systems can rarely be used to display the user's route history effectively - if at all. Typically, they only show the user's route history with a simple line or sequences of dots and neglect most other information (e.g. speed and direction of movement), which could be useful for the user.

AVI'08, 28-30 May , 2008, Napoli, Italy

Existing developments on the visualization of tracking data include, for example, the mPATH framework, which can be used for visualizing human behavior history [1] and the event visualization tool GeoTime, which allows combined temporal and spatial visualizations [2]. However, neither of these tools addresses the challenge with visualizing route history on a mobile device typically using a small low-resolution display. Navigation and tracking systems are used typically on mobile devices. However, existing route history visualization software has been developed mostly for desktop usage, while in many cases it would be beneficial for the user to review traveled routes on a mobile device.

Different visualization options to display scalable data include using varying color (e.g. in geographical maps for altitude), symbols (e.g. in weather maps in meteorology), and different geometrical shapes. It is also possible to use different brightness levels visualizing, for example, speed variation, and using spikes to show speed against distance [3]. The current vehicle navigation and tracking systems typically display speeds and distances in numbers and do not fully utilize these or other possibilities for visualization. The visualizations used in some current systems include, for example, aerial views of the route, lines with varying color together with a scale guide for visualizing variations in speed, arrow symbols for visualizing direction. However, using color variations to display varying speed does not provide the user with information about direction of movement. This becomes a problem especially when the route overlaps with itself – color ageing is one possible solution to this problem [4]. It has been found that users can extract information, which is not explicitly available even from complex map based visualizations [5]. The development of advanced visualizations provides important possibilities for efficient route visualization especially for mobile devices.

A more advanced mobile route visualization would be useful in various professional real-world applications. For example, it could provide information to support planning the routes of professional vehicles (e.g. trucks, delivery lorries, emergency vehicles). Current route planning systems are typically based on displaying average route parameter values (e.g. average speed, distance traveled). An advanced visualization could also be useful in large-scale traffic planning in towns and

cities. Another class of potential applications is professional sports applications (e.g. rally, cycling, skiing, snowmobiling), in which it is important to visualize overviews of speed variations on routes. Advanced route history visualizations could also be of great benefit in personal applications for the general consumer, for example, in planning and analyzing driving or sailing routes. They could be especially useful in off-road applications, especially when using topographical maps, which do not provide enough information about typical speeds, when traveling outside the road network.

Considering the popularity of navigation systems in real use it is surprising that there is currently very little research on visualization of route history. This study aims at filling this gap by introducing an advanced route history visualization for mobile devices. We describe and evaluate LocaweRoute, a visualization for displaying speed, direction, and location information efficiently on a mobile device.

METHOD

Subjects

12 subjects (8 male, 4 female) attended the experiments. The mean age was 38 years (range 22-53 years). In the pilot experiments, there were six participants and the mean age was 39 years (range 33-53 years).

Equipment

A HP iPAQ PDA device was used for both data collection (the visualized route) and for displaying the visualizations for the subjects. A Holux GPSlim236 receiver was used in data collection. The display resolution was 240x320 pixels. The software used for both collecting and visualizing the route information was built on Locawe, which is a platform for mobile location-aware systems. We have used Locawe before in studying location-based systems, car navigation, and mobile learning [6,7,8].

Route visualization

The LocaweRoute visualization is based on arrows (see Figure 1). The arrows were drawn as follows: the system selected a measurement point from the GPS data every 100 ms from the start of the data. The center of the base of each arrow was drawn at this measurement point. The direction was set so that the base was perpendicular to a (non-visualized) line between the base center point and the next measurement point. The arrow length correlated linearly with the vehicle speed at the starting point (e.g. 2 pixels per 10 km/h). Overlapping arrows were not drawn. A sample view of the visualization and a guide for the different arrow length used in the experiment is shown in Figure 1.

One route was used in all visualizations throughout the pilot experiments and the actual experiment. The evaluated route was driven specifically for this experiment in a rural area in Oulu South, Finland. The driven speed varied between 0 and 80 km/h and the duration of the route driven was 210s.

Procedure

The purpose of the pilot experiments was to find out the range of feasible visualization arrow lengths and widths. The

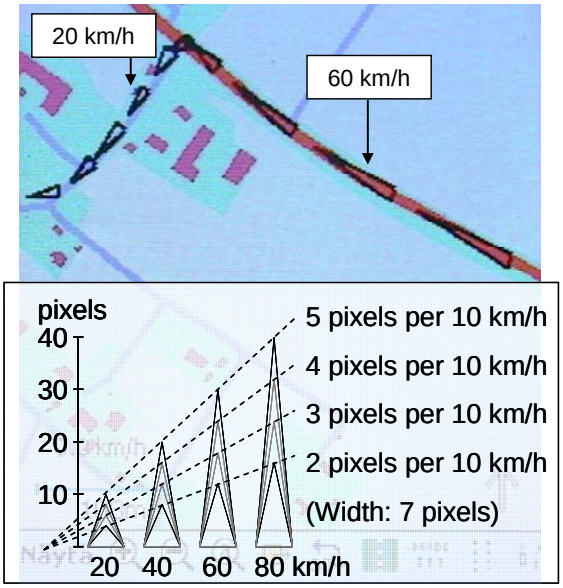


Figure 1. An example of the visualization with the arrow length 5 pixels per 10 km/h and a guide for all different arrow lengths. Points, in which the speed was 20 and 60 km/h are marked.

subjects systematically experimented with and evaluated different arrow lengths and widths using zoom level 2 (described later). As result of the pilot experiments, we chose the alternatives of 5, 6, 7, and 8 pixels in width and 2, 3, 4, and 5 pixels per 10 km/h in length for the actual experiment. The most preferred combination was 7 pixels (width) x 3 pixels per 10 km/h (length).

In tasks 1-4 of the actual experiment, the visualizations were shown to the subjects one at the time in randomized order and the subject was instructed to give an evaluation before moving on to the next visualization. In task 5, the subject was shown all four visualizations for comparison side-by-side using four different PDA devices. In all tasks, the subjects were instructed to pan through the whole visualization before giving an evaluation. Panning was carried out by pressing a stylus on the screen and moving it to the desired direction.

The purpose of task 1 was to find a subjectively preferred arrow width. The arrow length of 3 pixels per 10 km/h and the zoom-level 2 stayed constant and the width changed from 5 to 8 pixels. The subjects were asked to evaluate each arrow width using a 1 to 5 scale (1=too narrow, 3=suitable, 5=too wide).

In task 2 the subjects' task was to find arrow length scaling factors (2, 3, 4, or 5 pixels per 10 km/h), which are the best in expressing speed variation on four different zoom levels. The width of the arrows stayed constant (7 pixels). The zoom levels were chosen based on a previous study on automatic zooming in car navigation [8]. The map area that the different zoom levels displayed were as follows: Z1: 180x200m; Z2: 265x300m; Z3: 400x445m; Z4: 605x675m. The subjects used a 1 to 5 scale (1=too short, 3=suitable, 5=too long) to evaluate how well each length scale factor visualized variation in driving speed at each zoom level.

In task 3, the subjects' task was estimate actual driving speeds at given points of the route. They estimated points with actual driving speeds of 20, 40, 60, and 80 km/h using four different visualizations (arrow length scaling factors 2-5. The width (7 pixels) and the zoom-level (Z2) remained constant. The subjects wrote their estimates of the driving speed on an evaluation form (e.g. '78 km/h'). The evaluation points (different for each visualization) were pointed for the subjects by the researcher using a sharp pen.

In task 4 the subjects were asked to evaluate the four visualizations (arrow length scaling factors 2-5) in terms of the quality of the visualization in visualizing (1) speed, (2) direction, and (3) location of each point on the route. They also gave an overall evaluation for each visualization. The width (7 pixels) and the zoom-level (Z2) were constant. Scale 1 - 9 (1=poor – 9=good) was used.

In task 5, the subjects were asked to rank the four visualizations using arrow scaling factors 2, 3, 4, and 5 from 1 (=best) to 4 (=fourth best) based on their quality on the same four scales as in task 4.

Data analysis

Within-subjects ANOVAs (with Greenhouse-Geisser corrected degrees of freedom) and paired t-tests were used to test the speed estimation data for significant differences. For all other data, their nonparametric equivalents, Friedman's rank tests and Wilcoxon's matched pairs signed ranks tests were used to analyze the data statistically.

RESULTS

The results for the first task confirmed that the visualization arrow length used in the subsequent task (7 pixels) was evaluated as the most suitable width with a mean of 2.9. The ratings for the other alternative widths were: 5 pixels, 1.8, 6 pixels, 2.4, and 8 pixels, 3.6. The results for task two are shown in Figure 2.

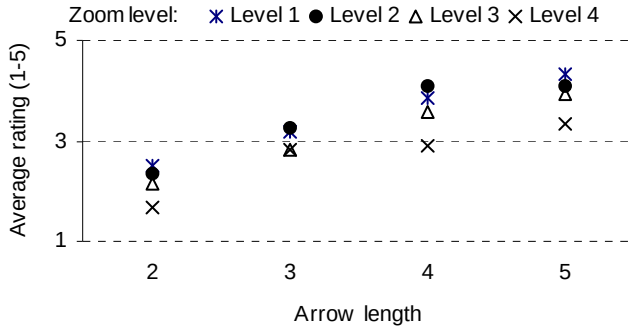


Figure 2. The ratings for the preferences for visualization arrow length on four zoom levels.

Statistical analyses of the preferences for of the arrow length showed there were significant differences in the data $\chi_F^2 = 26.8$, $p < 0.001$. There were four significant pairwise differences: Arrow lengths three, four, and five were all evaluated as significantly different from arrow length two (in all three comparisons: $Z = 3.1$, $p < 0.01$). Arrow length three was also evaluated as significantly different from arrow length five $Z = 2.5$, $p < 0.05$.

The results for task three are presented in Figure 3.

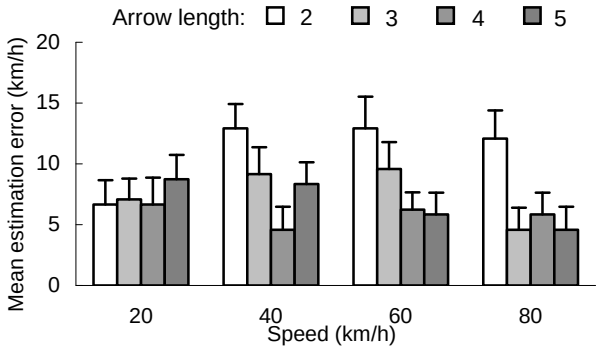


Figure 3. The mean speed estimation errors for each speed and visualization.

The length of the visualization arrow used had a significant effect on the accuracy of the subjects' speed estimations $F(2.5, 28.0) = 4.7$, $p < 0.05$. The related pairwise test showed that using the visualization with arrow length four, the subjects made significantly more accurate speed estimations than using the visualization with arrow length two $t(11) = 3.0$, $p < 0.05$. Similarly, using the visualization with arrow length five, the subjects' speed estimations were more accurate than using the visualization with arrow length two $t(11) = 2.6$, $p < 0.05$.

The results for task four are shown in Figure 4.

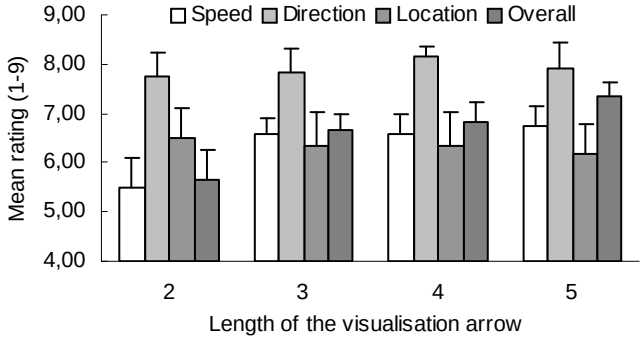


Figure 4. The subjects' evaluations of speed visualization support, direction visualization support, location visualization support and an overall evaluation.

The length of the visualization arrow used had a significant effect on the overall subjective evaluations of the visualizations $\chi_F^2 = 10.6$, $p < 0.05$. The visualization with arrow length five was given significantly better overall evaluations than the visualizations with arrow lengths two $Z = 2.6$, $p < 0.05$ and three $Z = 2.5$, $p < 0.05$. In addition, the visualization with arrow length three was given significantly better overall evaluations than the visualization with arrow lengths two $Z = 2.6$, $p < 0.05$. In contrast, the length of the visualization arrow did not have significant effects on the other evaluation criteria (i.e. speed/location/direction visualization support), when they were evaluated separately.

The results for task five are presented in Table 1.

Table 1. The mean rankings of the visualizations (1=best)

Evaluation / arrow length	*2*	*3*	*4*	*5*
Speed visualization support	3.3	2.2	2.7	1.8
Direction visualization support	3.2	2.5	2.3	2.0
Location visualization support	2.5	2.2	2.5	2.7
Overall	3.2	2.3	2.4	2.2

Statistical analyses showed that the rankings of the different visualizations differed significantly in terms of speed visualization support $\chi_F^2 = 10.6$, $p < 0.05$. The speed visualization support of arrow length five was evaluated as higher than that of both arrow length three $Z = 2.0$, $p < 0.05$ and two $Z = 2.2$, $p < 0.05$. The speed visualization support of arrow lengths three $Z = 2.8$, $p < 0.01$ was also evaluated as higher than that of arrow length two. Differences on the other scales were not significant.

DISCUSSION

Advanced visualizations of route history have been largely missing. The results of this study suggest that the LocaweRoute visualization was successful in visualizing the speed, direction, and location of a vehicle based on route history data on a mobile device. This was evidenced by the speed estimation data, which showed that the subjects were able to estimate actual driving speeds quite accurately, and the subjective evaluations of the speed, direction, and location knowledge support of the visualizations.

We also studied the LocaweRoute visualization with different arrow length scaling factors. These results suggest that a linear mapping of vehicle speed and visualization arrow length works well, but the exact optimization of the scaling factor is not very crucial as long as the visualization is approximately optimally proportioned to the map materials and the display size. This was also evidenced by the results of the evaluations using different zoom levels: there was an implementation of our visualization (using length scaling factor 3), which was evaluated as suitable at all four zoom levels on average. Naturally, further benefits could probably be gained by also scaling the size of the visualization arrows by zoom level, especially in systems with a large range of different zoom levels.

The visualization described in this paper was adjusted especially for visualizing route histories of vehicles on mobile devices. However, it could also be used with small adjustments for route history visualizations on larger displays such as on traditional desktop computers. In visualizing pedestrian routes, higher speed scaling factors should be used. Finally, the visualization could even be used in presenting route instructions or plans. In this case, the speed data could be automatically assigned by the system according to the prevailing speed limits.

The results suggest that the developed visualization is a noteworthy alternative for current route history visualizations. It has potential to be used in professional and personal applications. It could be especially useful in off-road applications, when using topographical maps.

In the future, we plan to improve the visual layout of the developed visualization and further refine the details of how the visualization arrows should be drawn. We also plan to further evaluate this visualization against alternative visualizations. The current results also contribute towards building software, which can visualize the user's route history in an intelligent way. Such software could, for example, scale the route history visualization automatically according to map materials, display size, tracked unit (e.g. pedestrian vs. vehicle), or zoom level. This kind of software would be of benefit to many professional and personal users of location-based systems worldwide.

ACKNOWLEDGMENTS

The authors would like to thank all the test subjects and everybody who has contributed to the development of the Locawe platform. This work was supported by European Regional Development Fund, State Provincial Office of Oulu, Ylivieska Region, Ylivieska Town, and Pohjanmaan Puhelin.

REFERENCES

1. Ito, M., Furuichi, Y., Nakazawa, J., and Tokuda, H. mPATH View: an interactive behavior history viewer for enhancing communication. In Proc. Pervasive Computing 2005, Springer-Verlag (2005), 93-96.
2. Kapler, T., Harper, R., and Wright, W. Correlating Events with tracked movements in space and time: a GeoTime case study. In Proc. IA 2005, (2005).
3. Stopher, P.R., Bullock, P., and Jiang, Q. Visualising trips and travel characteristics from GPS data. Road and transport research 12, 2 (2003), 3-14.
4. Aris, A., Gemmel, J., and Lueder, R. Exploiting location and time for photo search and storytelling in MyLifeBits. Technical report MSR-TR-2004-102, Microsoft Research (2004), 1-8.
5. Trafton, J.G., Marshall, S. Mintz, F., and Trickett S.B. Extracting explicit and implict information from complex visualizations. Diagrams 2002, Springer-Verlag (2002), 206-220.
6. Partala, T., Luimula, M., and Saukko, O. Automatic rotation and zooming in mobile roadmaps, In Proc. MobileHCI'06, ACM Press (2006), 255-258.
7. Haapala, O., Sääskilahti, K., Partala, T., Luimula, M., and Yli-Hemminki, J. Parallel Learning between the Classroom and the Field using Location-Based Communication Techniques, In Proc. ED-MEDIA 2007, AACE (2007), 668-675.
8. Luimula, M., Sääskilahti, K., Partala, T. and Saukko, O. A Field Comparison of Techniques for Location Selection on a Mobile Device, In Proc. WAC 2007, IADIS (2007), 141-146.

The Effect of Animated Transitions in Zooming Interfaces

Maruthappan Shanmugasundaram
Department of Computer Science
University of Manitoba
satish.shanmugasundaram@jri.ca

Pourang Irani
Department of Computer Science
University of Manitoba
irani@cs.umanitoba.ca

ABSTRACT
Zooming interfaces use animated transitions to smoothly shift the users view between different scales of the workspace. Animated transitions assist in preserving the spatial relationships between views. However, they also increase the overall interaction time. To identify whether zooming interfaces should take advantage of animations, we carried out one experiment that explores the effects of smooth transitions on a spatial task. With metro maps, users were asked to identify the number of metro stops between different subway lines with and without animated zoom-in/out transitions. The results of the experiment show that animated transitions can have significant benefits on user performance - participants in the animation conditions were twice as fast and overall made fewer errors than in the non-animated conditions. In addition, short animations were found to be as effective as long ones, suggesting that some of the costs of animations can be avoided. Users also preferred interacting with animated transitions than without. Our study gives empirical evidence on the benefits of animated transitions in zooming interfaces.

Keywords
Animation, zooming interfaces, information visualization.

1. INTRODUCTION
Visualization systems commonly employ animated transitions to shift between different views of a workspace. Animations appear in transformations that result from navigation, rotation, hiding and revealing structure, zooming in and out of the space, or switching between detail view and overview. Designers include animations between view transitions to help a user maintain a sense of the true nature of the information when visual changes occur during view transformations. Intuitively, designers believe that smooth transitions will result in reduced time and effort as users mentally reorient themselves to the structures visible at the completion of the transformation.

While animated transitions are a common element in many interfaces, very little empirical evidence supports the effectiveness of such a feature. On one hand, intuition suggests that animated transitions may reduce the cognitive load required by the user to maintain a mental map of changes occurring in the system. However, evidence also suggests that the time delays caused by animations can be disruptive, reduce efficiency and lead to frustrations [7]. Therefore, it is important to understand whether the use of animated transitions in visual interfaces is effective.

We report the results of on an on-going project that aims at identifying instances in which animated interfaces are effective. In this paper we evaluate the effectiveness of animated transitions in zooming interfaces. In one experiment, users performed a spatial task on subway maps. Our results suggest that while animations introduce time delays, users are faster in performing certain tasks with the animation than without. Furthermore, we found that animated transition speeds can be lowered from the commonly suggested values to create more efficient animated interfaces.

AVI '08, May 28-30, 2008, Napoli, Italy

2. RELATED WORK
We review the results that have inspired our study and contrast these against some of the drawbacks of animated transitions.

2.1 The potential of animated transitions
A number of studies have investigated the potential of animated transitions. Klein and Bederson [5] demonstrate that animating the movement of the document during the scrolling operation can improve target search tasks by up to 5.3% for text targets and 24% for graphical targets. Although animation can enhance scrolling performance, Andersen [1] suggests limiting the scrolling rate to the maximum rate a target can be perceived at during animation.

Bederson and Boltman [3] examined the effects of animated viewpoint changes on a user's ability to build a mental map of the information space. The authors compared two presentation types, animated and non-animated to test the effectiveness of animation for forming spatial structures. The participants were presented with a family tree containing images of different family members. Participants were asked to assemble the structure of the family tree based on the contents of the nodes they had seen previously. In this task, subjects performed better with smooth transitions than without. However, their results showed an ordering effect, i.e., if smooth transitions were shown first, then they performed significantly better than if they were shown last.

A study by Shanmugasundaram et al [8], explored whether animated transitions facilitate perceptual constancy in node-link diagrams. In their experiments, participants were required to identify entire tree structures by inspecting parts of the hierarchy that shifted in/out of view. Their results showed that users were capable of formulating structural relationships more efficiently with animated transitions than without. Surprisingly, participants took less time with animations to complete the task, than without.

Several techniques have used smooth transitions for gradually revealing information content. Continuous semantic zooming (CSZ) developed by Schaffer et al [6] employs animations to increase content visibility. This technique is characterized by two distinct but interrelated components: continuous zooming and presentations of semantic content at various stages of the zoom operation. When a region of interest becomes the focus, the user applies the continuous zoom to "open up" successive layers of the display. At each level of the operation the technique enhances continuity through animations between views, and thereby reduces the user's sense of spatial disorientation.

Continuous semantic zooming has been applied to information structures other than topological graphs. DateLens [4] employs CSZ to reveal varying degrees of content in tabular structures in a smooth and continuous manner. An evaluation comparing DateLens to common calendar-based interactions reveals that continuous semantic zooming enhances content browsing in tabular structures. Another distortion-based interactive technique was designed by Shi et al [9] for inspecting data in nodes of a TreeMap. The distortions are smooth transitions that gradually

expand the space allotted to a node. This enables users to see elements at leaf nodes without drilling-down through various layers of the hierarchy. In a study, Shi et al [9] showed that participants were able to identify content quicker and maintain context of the space better with smooth distortions.

2.2 Drawbacks of animated transitions

In spite of its advantages, animated transitions have numerous drawbacks. The most notable drawback is that animated transitions take considerable amount of time to complete a viewpoint transformation, thereby increasing system response time [3]. This additional time may not benefit users who are familiar with the task or when the task is not complex. Additionally, if animated transitions are not designed carefully, they can disrupt user performance and lead to distractions [2]. Bartram et al [2] evaluated the effectiveness of simple motion as a method of drawing the user's attention to an area of the display. Their results show that simple motion is significantly more disruptive than color or texture cues. From a design standpoint, implementing animations also requires more development effort. Additional algorithmic complexity is necessary to adequately interpolate between initial and final views of the animation. Furthermore, designers need to consider details such as the display's refresh rate or the user's hardware capacity. These constraints put an additional overhead in the development effort required for building an animated system.

In light of these drawbacks it is even more important for designers to be informed about the benefits that animations may provide. If there is evidence that animations provide significant benefits then designers may use these to outweigh the drawbacks of animated systems.

3. EXPERIMENT

The purpose of this experiment was to assess whether animated transitions are useful in zooming interfaces. Animation is applied at various steps in the zooming process thereby giving a smooth transition from zoom-in to zoom-out views, and vice versa. In an effort to create a canonical task with some ecological validity, we created a zooming interface for navigating through a spatial workspace, represented by subway maps of major cities. Subway maps have a close resemblance to a network or a node-link diagram where the subway lines appear as links and the subway stations act as nodes. The basic task was to navigate through a particular subway line and find the number of transferable intersections between two given points on that line, using zoom-in and zoom-out operations. Based on prior work we predicted the following outcomes:

Hypothesis 1: users will be more accurate when animated transitions are applied to viewpoint changes.
Hypothesis 2: completion times will be lower when animated transitions are used as in comparison to the no transition case.
Hypothesis 3: processing times (completion times - navigation time) will be the highest for the no transition case.

3.1 Method

3.1.1 Subjects

Sixteen subjects participated in this experiment (all male). All subjects were undergraduate students in computer science. The participants were regular users of mouse- and windows-based systems and had 5 to 16 years of experience with animated interfaces. They also had 3 to 8 years of experience using zooming interfaces primarily through computer gaming and map browsing applications such as Google™ and Yahoo ™ maps.

3.1.2 Materials

We used subway maps of four large cities for this experiment - Bangkok, Madrid, London and Paris. The maps were scaled to a maximum resolution of 2250x1500 pixels. We split the maps into two categories: Small (Bangkok and Madrid) and Large (London and Paris). Small maps had 6 and 8 railway lines while the large maps had more than 12 railway lines. All the railway lines were marked by a unique color.

The experimental setup was developed using .NET running on a P4 Windows XP PC system. The display was a 17" monitor set to 1280×1024 resolution. Two types of views were employed for this experiment: zoomed-out view and zoomed-in view. The system toggled between these two views through mouse clicks using either animated or no transitions. The system always started in the zoomed-out view showing the entire tube map through a viewport. Moving the mouse over the viewport would draw a small rectangular viewfinder (99x66 pixels) around the mouse pointer. Clicking the mouse button would expand the map to its maximum size and also shift the map in such a way that the region under the viewfinder would fill the entire viewport (zoomed-in view). Clicking the mouse again in the viewport would result in the zoomed-out view thereby scaling down the entire map.

3.1.3 Task

The subjects were shown one of the four subway maps in the viewport at the beginning of each trial in the zoomed-out view. Every map that was shown consisted of two highlighted points, marked in red, on a particular subway/railway line. The task was to enumerate and answer a question based on the number of transferable intersections between the two highlighted points. A transferable intersection is an intersection of two or more subway lines, where a commuter can transfer from one line to another. On the map, these transferable intersections are either shown as a single small white circle or more than two small white circles connected at the intersection of two or more subway lines. Figure 1 shows the zoomed-out and zoomed-in views respectively.

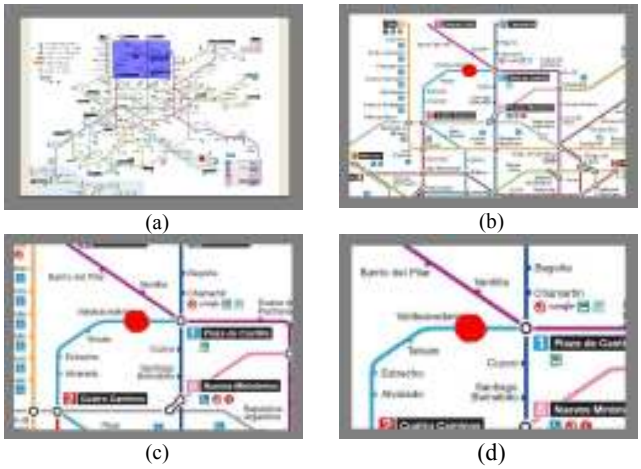


Figure 1 - Zoom-out to zoom-in, over multiple transitions.

When smooth transitions are employed the subject was able to see a number of intermediate views thereby giving a smooth transition effect between the zoomed-out and zoomed-in views. Figures 1.b and 1.c show the viewport during transition. In contrast, when no

transitions are employed the subjects would not see the map scale gradually and the net effect is that the users see the views in Figures 1.a and 1.d only. Clicking the mouse in the viewport, in the zoomed-in view, would make the system transit back to the zoomed-out view either using smooth or no transitions. The users were free to zoom-in and zoom-out as many times as they wanted to count the number of transferable intersections between the two highlighted points and answer a question. The question was always displayed below the viewport and it asked the user if the number of transferable intersections between the red dots was greater or less than a certain number. The user answered this question by clicking on the YES or NO buttons that were provided. The following data was collected for each task: Error rate, Task time and the Number of Zoom-in and Zoom-out operations. Error rate is directly related to whether users gave the right answer to the question, and the Task time is the time from the start of the task till the user clicks on the YES/NO button.

3.1.4 Design

The minimum size of the maps was 450 x 300 pixels (in the Zoomed-out view) and they expanded to a maximum size of 2250 x 1500 pixels (in the Zoom-in view). The experiment was setup using a 4x2 within-participants factorial design. The factors are:

Transition style: Slow-Transition, Medium-Transition, Fast-Transition and No-Transition

- Slow-Transition: this style zoomed-in or out in 1 second.
- Medium-Transition: zoomed-in or out in 0.5 seconds.
- Fast-Transition: this style zoomed-in or out in 0.25 seconds.
- No-Transition: this style zoomed-in or out in 1 millisecond.

Map Size: Small (6 to 8 subway/railway lines), Large (more than 12 subway/railway lines)

Transition style was fully counterbalanced using a Latin square design. The other factor was always presented in increasing order (i.e., from smaller to larger maps). Within each condition, participants carried out 4 trials. With 16 participants, 4 transition styles, 2 map sizes and 4 trials per condition, the system recorded a total of 512 trials. The system collected the total number of zoom-in and zoom-out operations, the errors and the total task time. Participants also filled out a brief questionnaire on their preferences at the end of the experiment.

3.1.5 Procedure

Participants were randomly assigned to one of the four groups obtained by counterbalancing the transition styles. Prior to starting the experiment, participants were given a small practice session which involved 2 trials per condition. After completing the practice trials, all participants indicated that they were comfortable with the four transition styles and the two types of maps being used. The participants then completed 32 trials without any breaks. At the end of the trials, the participants were asked to indicate the transition style that was easiest and the style for which they felt they performed the fastest.

3.2 Results and Discussion

We measured subjects' performance on the given task with respect to errors, task completion time and task processing time.

3.2.1 Error rate

The average error rate is summarized in figure 2 below. Average error rates were not consistent with the normality assumptions. The analysis was therefore performed on the log transform of the recorded error rates. The error rate was analyzed by means of a 4x2 (Transition Style x Map Size) one-way analysis of variance (ANOVA), with both Transition Style (Slow-Transition, Medium-Transition, Fast-Transition, No-Transition) and Map Size (Small, Large) serving as repeated measures (alpha=.05). The main effect of Transition Style was not found to be statistically significant at the 0.05 level ($F(3, 45) = 0.705$, $p = 0.554$). However the effect of Map Size was found to be significant ($F(1, 15) = 7.975$, $p = 0.013$) with the small size map mean error rate (3.9%) being smaller than the large size map mean error rate (11.3%). Finally there was no significant interaction effect between Transition Style and Map Size ($F(3, 45) = 0.442$, $p = 0.724$).

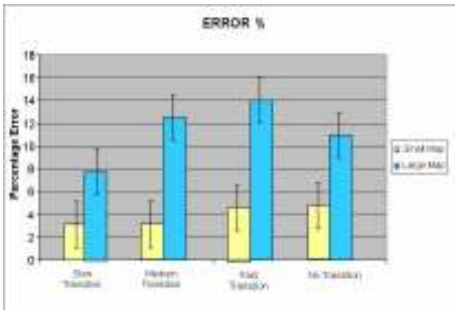


Figure 2 – Average error rates for each transition style.

Pair-wise comparisons reveal that the error rate is not significantly lower between the following transition styles: Slow-transition and Medium-transition ($p = 0.188$), Slow-transition and Fast-transition ($p = 0.173$), Slow-transition and No-transition ($p = 0.423$), Medium-transition and Fast-transition ($p = 0.609$), Medium-transition and No-transition ($p = 1.000$), Fast-transition and No-transition ($p = 0.580$). This rejects hypothesis-1 which states that users will be more accurate with smooth transitions. However pair-wise comparisons on Map size show that the smaller map error rate is significantly lower than the error rate on larger maps ($p = 0.013$).

3.2.2 Task Completion Time

The average task completion time is summarized in Figure 3. Task completion time is the amount of time (in seconds) a participant took from the moment a map was shown, till the participant gave a response by clicking on the YES/NO buttons. The completion time was analyzed by means of a 4x2 (Transition Style x Map Size) one-way analysis of variance (ANOVA), with both Transition style and Tree size serving as repeated measures. An alpha level of .05 was used for all statistical tests.

The main effect of Transition Style was found to be significant ($F(3, 45) = 7.424$, $p < 0.001$) with the average task completion time for No-transition (50.688 secs) being considerably higher than Fast-transition (35.617 secs), Medium-transition (36.453 secs), and Slow-transition (36.898 secs). The effect of Map Size was also statistically significant ($F(1, 15) = 42.685$, $p < 0.001$) with the small map average completion time (30.84 secs) being considerably lower than the completion time for larger maps (48.988 secs). We found a significant interaction effect between Transition Style and Map Size ($F(3, 45) = 3.652$, $p = 0.019$).

Pair-wise comparisons reveal that completion time for Slow-transition is not significantly lower than that of Medium-transition ($p = 0.863$) and Fast-transition ($p = 0.637$). Also, the completion time for Medium–transition is not significantly lower than the completion time for Fast-transition ($p = 0.737$), thereby

suggesting that performance based on task completion times are independent of the type of smooth transitions being employed. But the completion time for No-transition is significantly higher than the completion times for Slow-transition ($p = 0.024$), Medium-transition ($p = 0.008$) and Fast-transition ($p = 0.001$). This result supports hypothesis-2, suggesting that completion times are lower when smooth transitions are used. This strongly justifies the necessity of animation in zooming based applications.

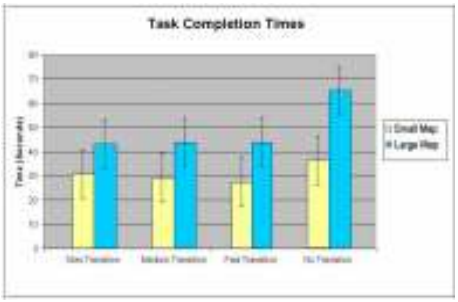


Figure 3 – Average task completion times per transition style.

3.2.3 Task Processing Time

The average processing time is summarized in Figure 4. Processing time is derived from the task completion time and the number of zoom-in and zoom-out operations. Task completion time is the time from the moment the participant starts the task to the time he/she responds by clicking the YES/NO buttons. During this time, the participant navigates the map through multiple zoom-in/-out operations, using either smooth transitions or no transition. Processing time is the task completion time minus the transition time, which is calculated from the number of zoom-in and zoom-out operations. We present our results with respect to task processing time, as it is a good measure to analyze the effect of transition style on cognitive processing ability.

The processing time was analyzed by means of a 4x2 (Transition Style x Map Size) one-way analysis of variance (ANOVA), with both Transition Style and Map Size serving as repeated measures (alpha=.05). The main effect for Transition Style was found to be statistically significant ($F(3, 45) = 18.806$, $p < 0.001$) with the participants requiring more processing time with No-transition (50.688 secs) as compared to the Slow-transition (26.563 secs), Medium-transition (30.105 secs) and Fast-transition (32.531 secs) conditions. The main effect of Map Size was also statistically significant ($F(1, 15) = 42.524$, $p < 0.001$) with the small map processing time (26.735 secs) being substantially lower than large map processing time (43.208 secs). However a significant interaction effect was found between transition style and map size ($F(3, 45) = 5.146$, $p = 0.004$).

Pair-wise comparisons show that there is no significant difference between Slow-transition and Medium-transition ($p = 0.109$) and no significant difference between Medium-transition and Fast-transition ($p = 0.271$). We found significance between Slow-transition and Fast-transition ($p = 0.025$) suggesting that Slow-transitions are better than Fast-transitions in terms of processing times. The most important point is that there is significant difference between No-transition and Slow-transition ($p < 0.001$), No-transition and Medium-transition ($p < 0.001$) and, No-transition and Fast-transition ($p < 0.001$). This result strongly supports hypothesis-3 stating that the processing times are the highest for the No-transition case.

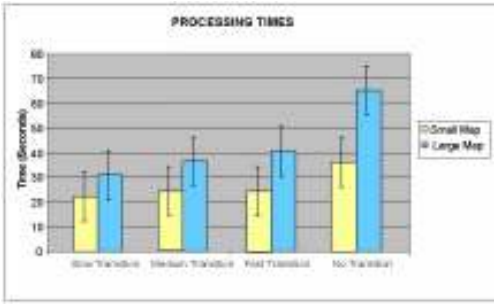


Figure 4 – Average processing times for each transition style.

3.2.4 User Preference

Participants answered two questions (Q1 and Q2) at the end of the experiment. Q1 asked them to indicate the animation style they thought was easiest and Q2 asked them to suggest the animation style that helped them complete the task faster. Fifteen out of sixteen participants rated one of the three animations as faster and easier while only a very few preferred the no-transition style.

4. Discussion and Conclusion

Overall, our analyses suggest that while participants are as accurate with animated transitions as without (reject hypothesis 1), they are approximately twice as fast with animations (support for hypothesis 2), and require much less processing with animations (support for hypothesis 3). Interestingly, we did not find any significant differences between different animation styles. This may suggest that for certain tasks, animation speeds could be reduced to ¼ of a second. This result is important as it can guide designers in integrating animated transitions in visual systems. In future work, we intend on quantifying more precisely the effects of smooth transitions with zooming or other interactive tasks, determining the correlation between transition speed and task complexity, and investigating the effects of different transition styles, such as slow-in/slow-out or variable transitions speeds on task performance.

5. REFERENCES

[1] T. H. Andersen. A simple movement time model for scrolling. *In CHI'05 Extended Abstracts*, 1180-1183, 2005.

[2] L. Bartram, C. Ware, W.T. Calvert. Moticons: detection, distraction and task. *Int. J. Hum.-Comput. Stud*. 58(5): 515-545, 2003.

[3] B. B. Bederson, A. Boltman. Does animation help users build mental maps of spatial information?, *IEEE InfoVis*, 28 – 35, 1999.

[4] B. B. Bederson, A. Clamage, M. Czerwinski, G. G. Robertson. DateLens: A fisheye calendar interface for PDAs, *ACM TOCHI*, 11(1):90-119, 2004.

[5] C. Klein and B. B. Bederson. Benefits of animated scrolling. *In CHI '05 Extended Abstracts*, 1965-1968, 2005.

[6] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, M. Roseman. Navigating Hierarchically Clustered Networks Through Fisheye and Full-Zoom Methods. *ACM TOCHI* 3(2):162-188, 1996.

[7] A. Sears, J. A. Jacko, & M. S. Borella, Internet Delay Effects: How Users Perceive Quality, Organization, and Ease of Use of Information, *In Extended Abstracts of CHI*, 353-354, 1997.

[8] M. Shanmugasundaram, P. Irani, C. Gutwin. Can smooth view transitions facilitate perceptual constancy in node-link diagrams?, *Graphics Interface*, 71-78, 2007.

[9] K. Shi, P. Irani, B. Li. An Evaluation of Content Browsing Techniques for Hierarchical Space-Filling Visualizations. *IEEE InfoVis*, pp. 81-88, 2005.

Visual Design of Service Deployment in Complex Physical Environments

Augusto Celentano and Fabio Pittarello
Dipartimento di Informatica, Università Ca' Foscari Venezia
Via Torino 155, I-30172 Mestre (VE), Italia
{auce,pitt}@dsi.unive.it

ABSTRACT

In this paper we discuss the problem of deploying appliances for interactive services in complex physical environments using a knowledge based approach to define the relations between the environment and the services, and a visual interface to check the associated constraints, in order to design a solution satisfactory for the user.

General Terms

Navigation, virtual and augmented reality, visual interaction, X3D

1. INTRODUCTION

Design of pervasive applications is an iterative process involving changes both in the application services and in the environment in which they are deployed; it is the result of a cooperation among a team of designers with different experiences and skills, examining the problem from different perspectives, and with different evaluation parameters in mind.

The team should include experts from many disciplines: architecture, information technology, HCI experts, etc., leading to a coordinated design of the physical environment and of the application. Coordinate design faces, in a unified process, the application services, their deployment in the environment, the user actions and the assistance to be provided for overcoming the difficulties due to user and environment limitations.

If a complex distributed application must be deployed in an existing physical environment, designers might not have sufficient degrees of freedom to optimally deploy the appliances for service access. The physical environment (a building, a set of rooms, an area) might discourage or prevent at all the implementation of some services and user assistance supports because incompatible with the environment structure. Nevertheless, the designers can collaborate to select a "good" deployment of interactive services within the constraints of the physical environment.

The interaction designer plays the most critical role in the team, identifying the activities executed in the environment, associating them to the environment locations and appliances and proposing to the environment designer and to the information engineer a suitable service deployment scheme compatible with the environment. The association between activities and locations might lead the designer to discover points of weakness in the environment spatial configuration, which in critical cases could require the environment designer, i.e., the architect, to conceive a change in the environment physical structure to comply with the application to be deployed. Unfortunately, in most real cases, the change of the physical environment cannot be done and the application must be changed to fit the environment constraints, often lowering the quality of interaction.

The approach proposed in this work tries to limit such drawbacks, enabling the designers to propose alternative solutions that can be checked for consistency by the system.

AVI'08, May 28-30, 2008, Napoli, Italy

2. INTERACTION AMBIENTS

Human activities take place in environments that we define *interaction ambients*, focusing on the availability of embedded interactive appliances and devices accessed by users in order to get information and services.

According to the *activity theory*, human activities are performed through the execution of *tasks* corresponding to specific user goals, according to some plan. Tasks are composed of *actions*, which are the basic level of operation on some device. Actions are executed by using *tools* which, in the information technology domain, include also service providers [2, 5].

Complex activities often require the user to move from place to place in the environment; hence, besides activity related tasks, also navigational tasks are important in interactive ambients. An interaction ambient is therefore best modeled as a set of connected locations, populated by objects and artefacts. Each location is defined by a physical or visual partition in the space (e.g., by walls or by pieces of furniture), morphologically meaningful for the user and suitable for the development of classified user activities. People moving through the locations can access services available in such locations by executing the task actions, which are therefore bound each to a single location.

From the information technology perspective, applications can be viewed as composed of *services*, works executed by a provider for a consumer; services produce both physical and abstract results, i.e., they may deliver concrete objects, such as a ticket, or produce changes in an information system, such as a reservation record. We distinguish between *field services* and *local services*: field services allow the user to access the service in a wide area, for example through a Wi-fi access point or through a large public display, while local services are bound to a device requiring user proximity or contact, for example a kiosk or an ATM. At design time, field services require the analysis of features such as range, covering and shields, while local services require the analysis of the physical placement.

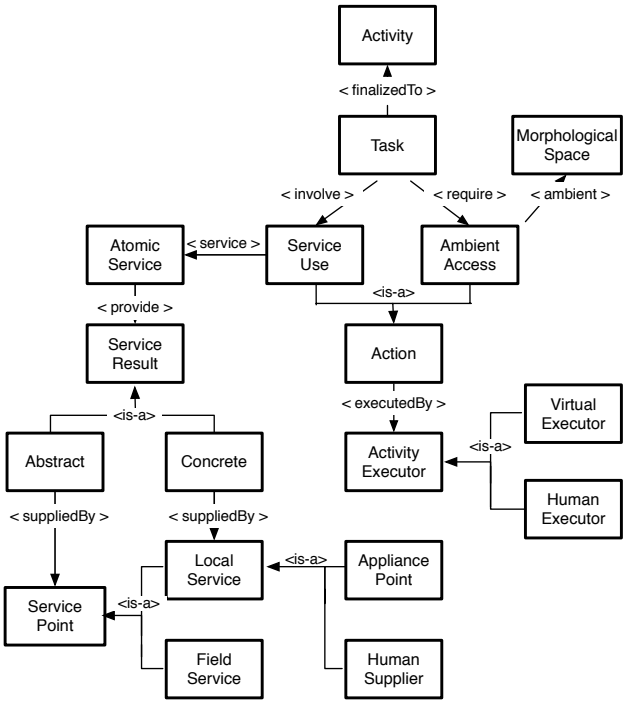


Figure 1: The activity and service ontology

3. PERVASIVE SERVICE DEPLOYMENT

In this section we present the functional architecture of a system supporting the designer in deploying pervasive services in complex interaction ambients. The system operates on a semantic description of the ambient, of the activities performed in it and of the available services, provided by a set of ontologies. An *ambient ontology* describes the physical environment and the objects in it, which are the interface between the user and the application services. A *service ontology* describes the services available and the concrete or abstract results delivered. An *activity ontology* relates services to ambients, describing how to execute in an effective way complex activities. Ontologies are defined at three levels: a meta-level describes the general structure of an ambient, service and activity, independent from the specific domain; a domain dependent level describes ambient, activity and service types relevant for an application domain; an instance level related to a case describes a specific physical environment, a defined set of activities and the set of services provided by that case.

Figure 1 shows a simplified view of the relations between the service and the activity meta-ontology components, which refer to the activity theory concepts, as illustrated in Section 2. An activity is composed by a set of tasks; to execute a task, a user needs to use services by accessing (hence navigating) the ambient in which they are available. Such elementary actions can be executed through a virtual executor (e.g., a program) or a human executor (e.g., an employee at a desk or the user him/herself).

Atomic services deliver concrete (objects) and abstract (information) results through field and local services. Local services can be supplied by appliance points or by human suppliers. Due to space limits, the reader is referred to [3] for details.

Figure 2 illustrates the functional architecture of a system able to support the interaction designer to deploy services in a way consistent with the environment constraints. The designer uses a visual interface based on a virtual reality paradigm for displaying a navigable 3D model of the physical environment. The interface displays also the list of activities and tasks, allowing the designer to trigger the system components devoted to deploying the service appliances in selected locations.

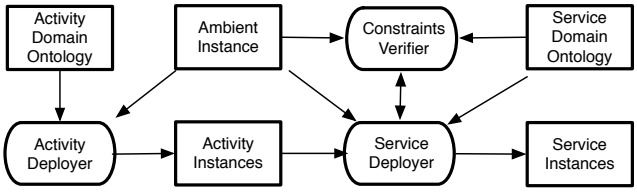


Figure 2: A functional architecture for service deployment

The *activity deployer* takes as input the description of the activities at the domain level provided by the activity domain ontology, processes them together with the description of the physical environment, and produces as output the description of the activities deployed in that specific ambient. The deployment is based on the identification of the relevant functions of the environment locations and of the objects contained; the designer can examine the system proposal and modify it, e.g., by suggesting alternative locations or different paths to follow to execute the activity.

The *service deployer* places local service appliances and field service access points, defined by the service domain ontology, in selected locations, according to the designer choices. The technical features of the services concerning placement, access, use, safety, efficiency, etc., are part of the service ontology, and are expressed as constraints. Some constraints must be obeyed by all the services of a given type: for example, all the appliances for local services must be visible and reachable by users, and must have enough space around them to avoid obstructions. Other constraints are referred to specific appliances: for example, a telematic kiosk must be placed near to electric and network connections, next to a light source and in a safe place.

Table 1 shows a sample of constraints: they correspond to specific needs stemming from user, communication, safety and wiring requirements. User requirements are related to the ability to identify the presence of a service, to physically access the appliance if it is a local service, and to the ease of use. Communication requirements are related to the availability of a field service inside a given area, and are focused on avoiding discontinuities due to bad placement and occluding elements.

Table 1: Sample constraints for service appliances

appliance	*req.*	*type**	*description*
kiosk	user	m	place next to the user's path
kiosk	user	p	place next to the user's path start
kiosk	user	m	place in the user's sight
kiosk	user	p	don't place in narrow connection spaces
kiosk	user	p	place against an opaque surface
kiosk	user	m	place next to a light source
kiosk	wiring	m	minimize distance from existing wiring
kiosk	wiring	p	place next to existing partitions
kiosk	safety	p	place against a robust surface
kiosk	safety	m	place in controlled locations
wi-fi	comm.	m	minimize occlusion
wi-fi	comm.	m	place on the ceiling or on top of vertical partitions
wi-fi	wiring	m	minimize distance from existing wiring
wi-fi	wiring	p	place next to existing partitions
wi-fi	safety	m	place in controlled locations

* m=mandatory, p=preferred

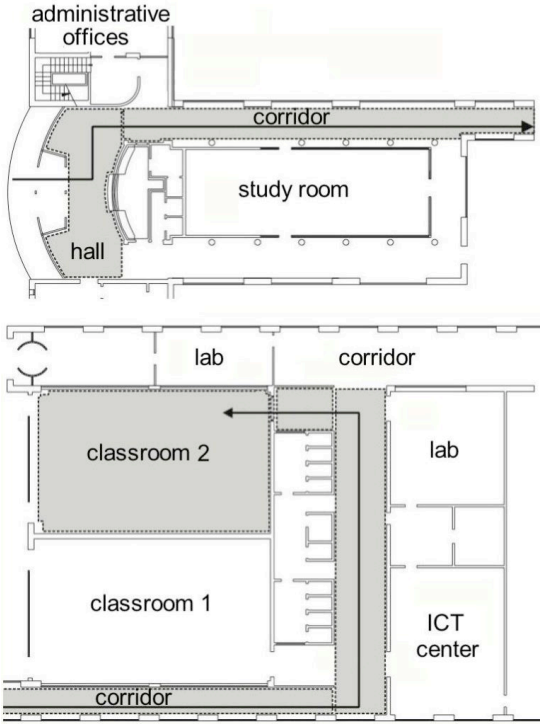


Figure 3: A map of two Computer Science Department areas

Safety requirements are related to the safety both of the appliance and of the user, preserving them from intentional or accidental physical damage. Wiring requirements are related to the infrastructure needed to connect the service appliance to the power supply and to the communication network.

An important component of the architecture is the *constraint verifier*, that allows the designer to check the compatibility of the local and field services deployed with the constraints defined in the service domain ontology, with respect to the ambient instance features.

4. A CASE STUDY

The building hosting the Department of Computer Science of Ca' Foscari University is divided into three areas connected by corridors and halls: an administrative area hosts the reception and administrative offices; an educational area hosts classrooms, undergraduate laboratories and the ICT centre; a research area hosts the teachers' and researchers' offices, the library and the graduate and post-doc labs. Study rooms and other services are distributed in the three areas, which therefore lack a strong separation of functions. Figure 3 shows a map of the first two areas.

Different classes of users populate the building for executing different types of activities. Because of the lack of a clear separation of functions in the building, casual visitors unfamiliar with the department physical structure may experience difficulties in finding the right place for performing their activity. While daily users need to be supported mainly for their core activity (such as attending lessons, consulting the library, etc.), occasional users, need assistance also for moving inside the building to locate the relevant rooms and services; navigation in the building becomes therefore an important task.

To exemplify the service deployment process, we consider an occasional user who has to attend a half-day seminar in classroom 2. Figure 3 shows the navigational path that the user should follow to attend the seminar, superimposed on the department map. While being only a part of the user activity, the task of going to the right place requires more support than the task of attending the seminar because of the user unfamiliarity with the building structure.

The designer effort is thus focused on placing services for helping the user to identify the correct path leading to classroom 2. The designer chooses to deploy a display with an interactive map of the building, showing the services, the current activities and their location, next to the building entrance, which is the beginning of any path for users' activities. Such local service will give assistance to the casual users even when the front desk personnel, usually working at the reception next to the entrance, is temporarily unavailable.

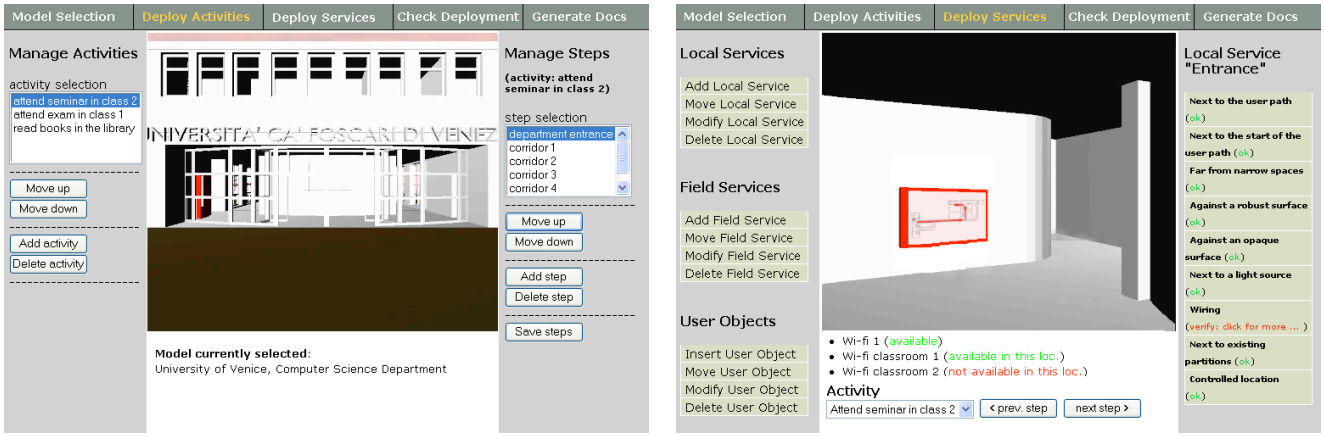
As a local service, the display placement must obey a number of constraints, similar to the ones listed in Table 1 for kiosks. The constraint verifier can be used to check the features of the *interactive display* appliance and of the location selected by the designer, showing the relevant constraints. Constraints that are violated by the placement, such as placement in a narrow passage, or in an unsafe location, or occluded from the entrance door, can be detected and corrected.

5. A VISUAL DEPLOYMENT INTERFACE

Figure 4 shows the proposal for the interactive deployment system interface. The upper menu shows the main functions that enable the user to choose the scenario to work with (item *Model Selection*), to deploy the activities (item *Deploy Activities*), to deploy the services (item *Deploy Services*) and to trigger the deployment checking system (item *Check Deployment*). The last item of the menu (item *Generate Docs*) generates the technical documentation produced by the system as part of the service deployment solution.

The central part of the screen is devoted to the 3D model of the physical environment. The ambient description is based on a proposal, described in [4], that takes into account not only the geometry of the objects represented, but also their high-level semantics: for example, an information panel is described both in terms of geometric primitives and with labels identifying its function. The approach is based on web standards: semantic web layers are used for defining the classes of semantic objects and their relations. The X3D standard [1] is used for describing the geometry of the 3D environment and its association with the higher-level semantic objects. While the semantic web layers define a scene-independent ambient ontology, the geometric description qualifies the specific ambient instance.

During the initial interaction phase, the interaction designer selects the X3D model to work with, which is visualized in the 3D window; Then, he/she defines the activities associated to the environment (see Figure 4.a). The activity deployer component supports the designer proposing, based on the knowledge of the environment semantics, a navigational path leading the user from some initial position of the activity (e.g., the building entrance) to the place where the activity is done or completed. In literature several navigation algorithms are discussed, which can be applied for accomplishing this task, that will not be detailed here. The interaction designer can modify the single steps of the path suggested by the system, using the menu on the right of the screen. Alternatively, he/she can design the user path from scratch, navigating the 3D environment and defining, as single steps, a set of views corresponding to subsequent current positions on the scene. The interaction designer can also choose to add new activities using the menu on the left side of the screen. At the end of the process, a set of paths associated to different activities, corresponding to the user movement in the environment for executing the activities, is available for the following design phases.



(a) (b)

Figure 4: The visual interface: (a) deploying activities, (b) deploying services

Figure 4.b displays a snapshot of the service deployment phase. The designer selects one of the activities defined in the previous phase, using the menu displayed on the bottom of the screen. This choice determines the initial view on the 3D window, according to the designed activity deployment. The designer moves along the path associated to the chosen activity using the buttons placed next to the menu, and can also decide to depart from the predefined path and to move freely in the environment. The designer, in such way, can simulate both the behavior of an expert user inside the real environment and of a user who doesn't know which services are available and is only able to identify them when they are visible.

While local services are made perceivable by sight, the field services associated to the environment are listed on the bottom of the interface because they can correspond to visible or hidden service points. Their availability in relation to the current user location is notified as a label next to the service item.

The column on the left enables the designer to add and remove local services and to move them to reach a specific location. The designer may also choose to add field services and define their properties, such as the coverage area.

The column on the right displays the list of constraints associated to the currently selected local service, in order to allow the designer to check the consistency of the deployment step-by-step, individually analyzing each service.

The example displayed shows the *Entrance* location where a local service with the same name is placed: an interactive panel delivering information about classroom location and current use. The list of the constraints associated to the category of devices classified as *kiosk* is displayed on the right. For each constraint the constraint verifier checks the consistency of the solution. If any problem is found (e.g., the placement of the local service next to the entrance is not compatible with the building wiring), a warning notifies the designer to manually check the item in order to find an alternative design.

If the result of the checking activity doesn't evidence any problem, the designer can update all the documents related to the new solution (i.e., the list of the service instances on the 2D building map) by clicking the button *Generate Docs*.

In many situations a given environment can be temporarily populated with additional services that are targeted to specific situations, such as an exhibition or a scientific conference, which are not part of the environment purpose, and need not to be accessed permanently. In such cases, without performing the complete design process, the designer can place and move new service appliances (if they are know to the system) inside the scene. During the validation phase that will follow the modifications operated by the user the system will consider also the presence of such new appliances and their additional constraints.

6. CONCLUSION

Deploying application services in complex environment requires architects, interior designers, application engineers and interaction designers to coordinate their effort to select proper service placements according to the environment functions and to the users requirements. Good appliance deployment is one of the main goals in the architectural domain, often achieved mainly thanks to the designer experience. Design systems based on formal grounds could provide a set of guidelines easier to apply and verifiable.

The use of knowledge based descriptions of activities, services and ambients constitutes a further step towards the design of interactive systems in pervasive environments. Existing environments could be insufficient to fulfill the trend of pervasive applications. The visual approach proposed in this paper can assist the designer to simulate environments and to move inside them with a VR-based metaphor to find nearly optimal placement of interactive appliances.

7. REFERENCES

[1] Extensible 3D (X3D) ISO/IEC 19775:2004. http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification, 2004.

[2] O. Bertelsen and S. Bodker. Activity theory. In J. Carroll, editor, *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*, pages 291–324, San Francisco, 2003. Morgan Kaufmann.

[3] A. Celentano, A. Okroglic, and F. Pittarello. An ontology based approach to interaction ambient design. In *Proc. Workshop on Mobile Services-oriented Architectures and Ontologies (MoSO)*. IEEE Computer Society, 2007.

[4] F. Pittarello and A. D. Faveri. A semantic description of 3d environments: a proposal based on web standards. In *Proc. Web3D 2006*, pages 85–95, 2006.

[5] M. Vukovic and P. Robinson. Adaptive, planning-based, web service composition for context awareness. In *Proc. Int. Conf. on Pervasive Computing*, 2004.

Visualizing Program Similarity in the AC Plagiarism Detection System

Manuel Freire
EPS-Universidad Autónoma de Madrid
Av. Tomás y Valiente 11,
ES-28049 Madrid, Spain
manuel.freire@uam.es

ABSTRACT

Programming assignments are easy to plagiarize in such a way as to foil casual reading by graders. Graders can resort to automatic plagiarism detection systems, which can generate a "distance" matrix that covers all possible pairings. Most plagiarism detection programs then present this information as a simple ranked list, losing valuable information in the process.

The AC system uses the whole distance matrix to provide graders with multiple linked visualizations. The graph representation can be used to explore clusters of highly related submissions at different filtering levels. The histogram representation presents compact "individual" histograms for each submission, complementing the graph representation in aiding graders during analysis.

Although AC's visualizations were developed with plagiarism detection in mind, they should also prove effective to visualize distance matrices from other domains, as demonstrated by preliminary experiments.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: [GUI, interaction styles]; K.3.2 [**Computer and Information Science Education**]: [Computer science education]; G.2.2 [**Graph Theory**]: [Graph algorithms]

General Terms

Algorithms, Design

Keywords

Software plagiarism, Visualization

1. INTRODUCTION

Many courses include programming assignments. Depending on the time constraints, honesty, and other factors, some students of these courses may decide to plagiarize from their colleagues instead of coding their own submissions. Cheating students are usually liable to heavy penalties – if discovered. However, students are well aware that manual discovery is only feasible in small groups. To reliably detect plagiarism, a grader would have to compare all possible pairings of student submissions to each other. Given N students, this scales as $O(N^2)$; manual detection is not feasible.

Revealing the use of automated plagiarism-detection systems to the students prior to completion of an assignment proves to be a remarkably strong (though still not absolutely perfect) deterrent [3], because it alters the cost-benefit analysis of potentially dishonest students. However, since computer programs lack the context to make moral judgements on academic dishonesty, the role of these systems should be limited to helping graders to discard the vast majority of non-plagiarized submissions and concentrate on the few were students may have yielded to temptation.

Software plagiarism detection systems such as MOSS [1], SHERLOCK [8] and SIM [7] use different similarity algorithms to fill a distance matrix, which must then be presented to the user. The default presentation mode for all of them is a ranked list, achieving roughly comparable results for precision and recall (see [11]).

Ranked lists discard a huge amount of data from the distance matrix. A ranked list will not allow a grader to determine whether an analysis is truly informative or too noisy to be trusted. Additionally, a single numerical score does not provide any further clues regarding the confidence that the system has in the similarity being due to plagiarism instead of coincidence, even though the distance matrix, taken as whole, may contain the necessary information. Graders faced with a ranked list are expected to perform manual checks in order of decreasing plagiarism-probability score until they locate a series of non-plagiarized pairs. However, a crafty plagiarizing student may introduce enough "noise" (modifications to the source that do not change program semantics) into a plagiarized submission to make it slip under the radar. Additionally, lists cannot display or help to identify closely-related groups of submissions; similarity need not be restricted to pairs. The graph and histogram representations described by this paper seek to address the above problems through a better visualization.

The following section describes visualization in AC, starting with an overview of the AC system, presenting the graph and histogram visualizations, and ending with a brief subsection on the use of AC's visualizations outside the domain of programming assignments. Section 3 summarizes the main results and describes future work.

AVI '08, 28-30 May , 2008, Napoli, Italy.

2. VISUALIZATION IN AC

The AC system[1], described in [6], has been developed in the Escuela Politécnica Superior of the Universidad Autónoma de Madrid, where it is currently being used in several courses. Since its introduction, (detected) plagiarism rates in these courses have dropped dramatically. These results support the observation quoted from [3]: it is enforcement of penalties, not the penalties themselves, which dissuades would-be trespassers.

AC currently supports three visualizations: table, graph and histogram. Once an analysis has generated a distance matrix, users can switch between the different visualizations (using the tabs shown at the top of, for instance, Fig. 1). The table visualization is the simplest of the three; similarity for each submission pair is displayed in a sortable table. The table is therefore equivalent to the ranked lists described in the previous section, and presents the same drawbacks.

2.1 Graph Visualization

The graph visualization provides an overview of the values in the distance matrix, displaying a graph that includes submissions with distances below a threshold. The primary goal of this visualization is to identify group relationships. For instance, if an analysis reports low distances between submissions A and B, and B and C, it may be interesting to know whether A and C are also closely related.

The lower part of the visualization (see Fig. 1) contains a (color-coded) histogram that indicates the relative frequency of each distance in the matrix. In an ideal scenario, the analysis will generate a roughly bell-shaped distribution. In other cases, such as when different course instructors have suggested different approaches, the distribution may look more like a superposition of distinct bell-like curves. In a pathological case, such as a choice of submission files which does not capture enough variability or includes too much noise, the distribution will be highly skewed towards low or high values. The global distance histogram can be therefore used to quickly gauge the relevance of the results provided by an analysis.

Since the left-most edge of the histogram represents lower distances, spikes to the left of the main distribution are suspect of plagiarism. By adjusting the position of the horizontal slider placed on top of the histogram, a grader can select the threshold to be used for graph representation: the area under the slider becomes shaded, and all submission pairs with distances below this threshold will be displayed in the main graph. The graph is redrawn automatically every time the threshold slider is move. The slider+histogram can be seen as an implementation of the scented widget concept [15]. Graph vertices are labeled with the corresponding submission IDs, and edges are color-coded and width-coded to indicate distance: edges representing very low distances are colored red and are several pixels wide, while those that represent larger distances become thinner as they progress through orange, yellow and green. Connected components are rendered inside separate grey boxes.

The general graph only renders a subset of all edges which fall below the threshold. Given a box that encloses a subgraph G', any edge that does not belong to the minimum spanning tree (MST) of this subgraph and is not within

[1]AC stands for "AntiCopias", and is available from `http://tangow.ii.uam.es/ac`

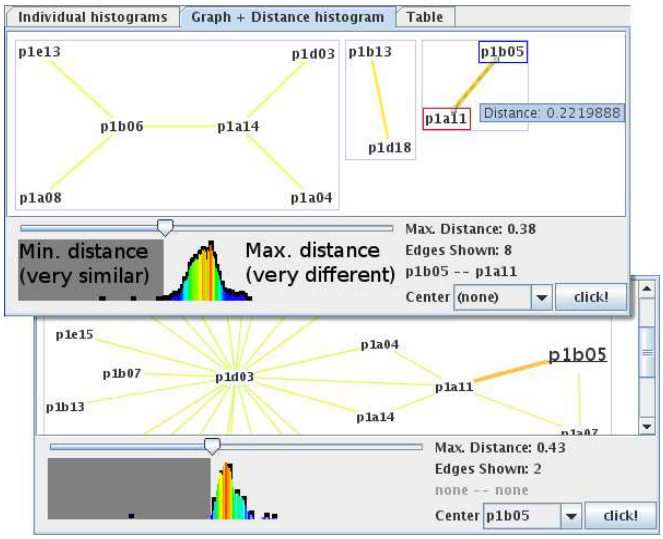


Figure 1: Graph visualization and distance histogram, using a low threshold. Above, without a center. Below, centered around submission *p1b05*.

the shortest |G'| edges is elided. This is a refinement over the edge-removal approach suggested by Whale in [14]; the result is low edge clutter, while low-distance edges corresponding to strong cliques are preserved.

Individual similarity graphs, centered on a particular submission, can also be generated. The "center submission" is selected by using the combo box located near the lower-right corner of the graph window. The center submission is highlighted using a large font, and the global histogram is substituted with the individual distance histogram for the center submission (these histograms will be introduced in the next subsection). Finally, a different criterion is used to select which edges and vertices to display: the goal is to display only those submissions that are highly related to the center one. Individual graphs allow graders to check on a particular submission in a graphical way. The distance threshold is preserved when the center is changed.

Graph drawing in AC relies on the CLOVER [5] library[2]. This library provides automatic layout, zoom and pan functionality, and allows users to manually displace vertices. The use of a force-directed layout (FDL, also termed 'organic' or 'spring') algorithm does not impose significant slowdowns on the interface, as long as the number of vertices displayed is kept below a few hundred.

Besides the expected zoom and pan behavior, hovering the mouse pointer over any graph edge displays the associated numerical distance as a tooltip, and double-clicking an edge displays a side-by-side comparison dialog for the connected submissions. Double-clicking on a single submission displays a small pop-up window with the individual histogram for the corresponding submission.

Similarity graphs are also available for the SHERLOCK system[13]. However, a circular graph layout is used, which is less informative than AC's organic layout. Additionally, there is no individual submission graph mode, and the horizontal slider used to set the edge inclusion threshold does not provide any clues on the actual distance distribution,

[2]Available from `http://tangow.ii.uam.es/clover`

making threshold selection a blind process.

2.2 Histogram Visualization

The histogram visualization displays a series of "individual histograms" stacked vertically. Each histogram displays the distribution of distances from a specific submission to all others; the submission ID for this specific submission is displayed in the left-most edge of the histogram (see Fig. 2). Individual histograms are usually presented in collapsed form. Selecting any row, however, will display it in the traditional, expanded form. The mapping from traditional histogram representation to the compact, color-coded representation is straightforward: the higher the frequency of a certain distance, the "redder" the color. Very frequent distances will be red, and progressively rarer distances will be colored orange, yellow, green and finally blue. The compact representation is also referred to as a "hue histogram", inspired on Kincaid and Lam's Line Graph Explorer [10]. Both expanded and compact representations can be found in Fig. 2.

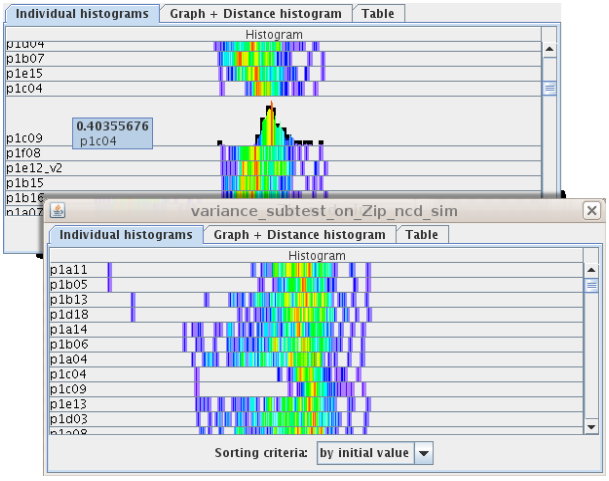


Figure 2: Individual histogram visualization. Each row corresponds to a single submission, and plots the histogram of distances to all other submissions. Selected rows are expanded and shown in traditional form. The lower histogram visualization has been generated after refining matrix values using the left-outlier heuristic.

Constructing histograms for floating-point values, such as similarity distances, involves choosing a number of buckets into which these distances can be aggregated; it is rare to find more than one distinct pair of submissions with exactly the same distance between them. To capture "distance sameness" in the compact histogram representation, two types of coloring are used. Buckets themselves are colored in unsaturated hues. Within each bucket, exact distances are colored in a completely saturated color matching the bucket's hue, at the horizontal position nearest to their actual value.

Hovering the mouse pointer over any point of a histogram displays the IDs of the submissions with distances closest to the current hovered-over value, together with their actual numerical values. Double-clicking on a histogram position will display a comparison screen featuring the two submissions that correspond to those IDs. This behavior is also available in the global histogram that can be found in the graph visualization (see Fig. 1).

2.3 Interpreting Individual Histograms

By default, individual histograms are sorted according to their lowest distances; this is equivalent to the ranked lists found in other systems.

Individual histograms, however, prove to be much more informative than simple numerical values. For instance, in Fig. 2, large gaps can be observed between the first leftmost spike in the histograms for *p1c04* and *p1c09* and the rest of their distance distributions. Not only is the distance between *p1c04* and *p1c09* low - it is also much lower than any distance from *p1c04* or *p1c09* to other submissions.

After manual inspection, these two submissions can be seen to share a substantial amount of source code, and *p1c09*'s authors recognized having plagiarized from *p1c04*. These observations can lead to the following interpretation rule: *a leftmost-outlier within an individual distance distribution is likely to be due to plagiarism.* The rationale is that, if Bob copies from Alice, his submission would be expected to be much more similar to Alice's than to all other, independently developed submissions. An interesting corollary is that the low distance between Bob and Alice's submissions may not be as important as its position within the histogram. If Bob's submission is much more similar to Alice's than to all other submissions, then it probably deserves a manual comparison by the grader - even if the distance itself, as reported by the current analysis, is not particularly low.

Indeed, if Bob wished to avoid detection after plagiarizing Alice, his best bet would be to introduce noise (random cosmetic changes) in the source code. Although this would increment the distance between both submissions, it would also increment the distances to other, unrelated submissions, leading to a right-shifted individual similarity histogram such as the one found in the rows for *p1c04* and *p1c09* of Fig. 2. In a traditional ranked-list representation (such as the upper part of Fig. 2), this would be row 22 – far outside the area where a grader would have looked. Using the histogram visualization it is easy to scan for this pattern, and manually examine each case.

A refinement heuristic has been developed to increase the visibility of such cases. When this heuristic is in use, each distance D_{AB} within the matrix is adjusted to factor in the "degree of outlierness" within the corresponding individual distance distribution (see [6]). In the lower part of Fig. 2, *p1c04* occupies row number 9, and is much more visible.

2.4 Application to Other Domains

Although developed for use in plagiarism detection, AC's visualizations can be applied to any distance matrix. In a recent experiment, a normalized compression distance similarity analysis was used to generate a distance matrix for a corpus of news headlines[3]. Most highly-related headlines were indeed semantically significant, even though the approach had not been optimized at all for natural language processing.

Two other experiments in non-plagiarism domains are currently under way. In the first one, AC is used to locate similar classes within a single large java program; the results

[3]Thanks to Jae-Wook Ahn, from the PAWS group at Pittsburgh University, for providing the corpus.

can be used in refactoring and testing efforts. In the second, Ac is used to analyze amino acid sequences, and the goal is to locate similarities at the sequence level that may prove significant at higher levels.

3. CONCLUSIONS AND FUTURE WORK

Ac's graph and histogram visualization represent improvements in the visualization of distance matrices, specially when compared to simple distance-ranking approaches.

The graph visualization couples the histogram of the distance population and the edge threshold slider, allowing intuitive selection of relevant thresholds. Additionally, the heuristic used to elide edges from connected components preserves the most important parts of the component structure, while simplifying the graph for later layout. The addition of an individual graph mode allows graders to quickly examine the position of a single submission.

The histogram visualization is based on individual histograms. To reduce space requirements, hue-histograms are used instead of bar histograms; greater compactness could be achieved by dropping submission labels, and generally adopting the interface of Card's Table Lens [12]. The interpretation of this visualization provides a wealth of information, and has prompted the development of a novel heuristic to detect cases of plagiarism in the presence of added noise.

Feedback from graders indicates that the tool is effective at detecting and, even better, deterring plagiarism. Performance on artificial benchmarks is very high (see [4]), although experiments in which students are asked to cheat have not been performed. Beyond the graph and histogram visualizations, graders have repeatedly requested support for incremental comparisons, and for the ability to avoid comparing old submissions among themselves.

3.1 Future Work

Although Ac is useful (and used) as-is, the interpretation of histograms and, to a lesser extent, graphs, is certainly not intuitive, and requires a degree of training and familiarity. Work is under way to simplify this task, by using statistical outlier detection to aid graders to quickly locate suspect submission pairs. An over-simplification of the interface, however, would be dangerous, since graders may decide to "let the system do the work" instead of making informed decisions based on the actual submissions.

Reviewers have provided insightful comments and pointers; color-coding could benefit from the ideas contained in [9], and histogram rows could be sorted as described in [2]. Additionally, the experiments described in section 2.4 have prompted the development of a new "dendrogram+graph" visualization, similar to the current "histogram+graph" visualization.

4. ACKNOWLEDGMENTS

This work has been sponsored by the Spanish Ministry of Science with project code TIN2004-03140.

5. REFERENCES

[1] A. Aiken et al. Moss: A system for detecting software plagiarism. *University of California–Berkeley. See www. cs. berkeley. edu/aiken/moss. html*, 2005.

[2] M. Ankerst, S. Berchtold, and D. A. K. Mihael. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings IEEE Symposium on Information Visualization 1998*, pages 52–60. IEEE, 1998.

[3] B. Braumoeller and B. Gaines. Actions do speak louder than words: Deterring plagiarism with the use of plagiarism-detection software. *PS: Political Science and Politics*, 34(04):835–839, 2002.

[4] M. Cebrián, M. Alfonseca, and A. Ortega. Automatic Generation of Benchmarks for Plagiarism Detection Tools using Grammatical Evolution. In *Proceedings of the 9th annual conference on Genetic and Evolutionary Computation*. ACM Press New York, NY, USA, 2007.

[5] M. Freire. *An Approach to the Visualization of Adaptive Hypermedia Structures and other Small-World Networks based on Hierarchically Clustered Graphs*. PhD thesis, Universidad Autónoma de Madrid, 2007.

[6] M. Freire, M. Cebrian, and E. del Rosal. Ac: An integrated source code plagiarism detection environment. Pre-print manuscript, available at `http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0703136`, May 2007.

[7] D. Gitchell and N. Tran. Sim: a utility for detecting similarity in computer programs. In *Proceedings of 13th SIGSCI Technical Symposium on Computer Science Education*, pages 266–270. ACM Press New York, NY, USA, 1999.

[8] M. Joy and M. Luck. Plagiarism in Programming Assignments. *IEEE TRANSACTIONS ON EDUCATION*, 42(2):129, 1999.

[9] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, Jan./Mar. 2000.

[10] R. Kincaid and H. Lam. Line graph explorer: scalable display of line graphs using focus+context. In *Proceedings of AVI 2004*, pages 404–411. ACM Press, 2006.

[11] C. Liu, C. Chen, J. Han, and P. S. Yu. Gplag: detection of software plagiarism by program dependence graph analysis. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 872–881, New York, NY, USA, 2006. ACM.

[12] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proceedings of CHI '94*, pages 318–322, New York, NY, USA, 1994. ACM Press.

[13] University of Aberdeen, CS Department. Student handbook: on plagiarism. `http://www.csd.abdn.ac.uk/teaching/handbook/both/info.php?filename=cheating.txt`. Last visited, Dec. 2007.

[14] G. Whale. Identification of Program Similarity in Large Populations. *The Computer Journal*, 33(2):140, 1990.

[15] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Trans. Vis. Comput. Graph*, 13(6):1129–1136, 2007.

Visual representation of web design patterns for end-users

Paloma Díaz
Laboratorio DEI
Universidad Carlos III de Madrid
Avda. de la Universidad 30, 28911
Leganés (Madrid). Spain
34916249456
pdp@inf.uc3m.es

Ignacio Aedo
Laboratorio DEI
Universidad Carlos III de Madrid
Avda. de la Universidad 30, 28911
Leganés (Madrid). Spain
34916249490
aedo@ia.uc3m.es

Mary Beth Rosson
Information Sciences & Technology
Pennsylvania State University
State College, PA (USA)
814-863-2478
mrosson@psu.edu

ABSTRACT
In this paper, we discuss the use of visual representations of web design patterns to help end-users and casual developers to identify the patterns they can apply in a specific project. The main goal is to promote design knowledge reuse by facilitating the identification of the right patterns, taking into account that these users have little or no knowledge about web design, and certainly not about design patterns, and that each pattern might include some trade-offs users should consider to make more rational decisions.

Categories and Subject Descriptors
D.2 [**Software Engineering**]: Reusable Software - *Reuse models*; D.2 [**Software Engineering**]: Requirements/Specifications - *Elicitation methods*; D.3 [**Programming Languages**]: Language Classifications – Design languages

General Terms
Design, Languages.

Keywords
Design patterns; web design; goal-oriented design.

1. INTRODUCTION

Design patterns document solutions that have been successfully applied to recurrent problems in software development [1]. As a design concept, patterns are rooted in the works of Cristopher Alexander in urban architecture. In Alexander's words, "*a pattern is, in short, at the same time a thing, which happens in the world, and the rule which tells us how to create that thing, and when we must create it. It is both a process and a thing; both a description of a thing which is alive, and a description of the process which will generate that thing*" [2, p. 247]. Compared to guidelines and principles, design patterns are always grounded in a number of real examples of application ("*is a thing which happens in the real world*"); they also include information on multiple and probably competing concerns that help to understand when the pattern should be applied ("*and the rule which tells us how to create that thing, and when we must create it"*). Patterns provide a systematic way to record knowledge experience using a predefined format. Though there are variations in pattern formats, all of them include at least the name, the description of the problem addressed by the pattern, the description of the solution, some examples of application and one or more sections discussing the context where the pattern should be applied and the context resulting from its application as well as its trade-offs. There are design patterns for different software development domains including object-oriented design [1], e-commerce [3], security [4] or web design [5, 6, 7]. This paper focuses on the web design domain.

Design patterns are relatively more structured than guidelines and principles, and they provide much more information that can help a designer decide whether or not to apply a specific pattern. Therefore, patterns may serve as design knowledge repositories that will help designers to make rational decisions. But the problem is that such knowledge is difficult to discover and to reuse for end-users with no experience in design. For instance, some empirical studies have shown that users with previous knowledge about design are much more proficient in the use of patterns than novice users [8], who paradoxically are the target users such design guidance. In this poster paper, we propose a visual representation that can be used to illustrate a problem specification using patterns. The technique is based on soft-goal graphs that have been successfully applied to specify, validate and identify non-functional requirements [9]. Our goal is to assist novice web designers to select the patterns they need.

2. MOTIVATION AND RELATED WORKS

Design patterns are usually organized in catalogues, whether text-based or web-based; the catalogues enable designers to browse the patterns using different criteria (e.g., the design concern or the alphabetical order). A set of design patterns can also form a cohesive pattern language, which makes explicit the relationships existing amongst patterns (basically composition and association). In the specific domain of web design, there are several collections of design patterns, including those published in books like [3, 5, 10] and in web-based catalogues like [6, 7, 11].

But as said before, applying patterns is not easy, particularly for novice designers with no expertise in the domain of application (the web in our case) and in the use of design patterns. Firstly, applying a pattern implies an initial identification of the pattern and understanding of its context, forces and design rationale; but a novice web designer cannot easily determine some rational

AVI'08, May 28-30, 2008, Napoli, Italy

criteria to select a specific pattern from a collection, particularly taking into account that collections are primarily browsed alphabetically or by design concern and that each pattern might have some trade-offs that can undermine its applicability.

Secondly, patterns are described at different levels of abstraction. Some patterns provide solutions accompanied by images and pictures that can be understood by end users, but in other cases the descriptions are abstract and may even use very technical terms whose meaning is not obvious for novice designers.

Thirdly, the representation of the knowledge space underlying the patterns is in all cases quite poor and flat. On the one hand, patterns are accessed using indexes, whether alphabetical or thematic (namely design concerns). The first option is only useful for users who already know the patterns, and the second one can be used for exploratory tasks where users analyze the full catalogue trying to figure out which solutions can be reused. It is important to keep in mind that when designing a web site there are always time constraints and that identifying useful patterns is just one part of the design task. A second perhaps even more complex aspect of the task consists of adapting a given pattern solution to the specific project being addressed. In order to simplify the identification of the right design pattern, Kanmpffmeyer and Zschaler [12] use a ontology to browse 23 object-oriented patterns by their "Intent" part which is the one that includes the description of the problem addressed by the pattern. In this case, the tool has an interface oriented towards experts in ontological engineering. What we propose is to apply visual representations that can be more usable for our target audience, novice web designers.

On the other hand, there is no visual representation of the relationships existing amongst patterns. Each pattern is seen on an individual basis and even though there is a "Related to" section listing all the related patterns, there is no visual distinction amongst the different kinds of relations (for example, negative or positive contribution to the problem being addressed) so the user has to realize by herself which are the interdependencies amongst the set of pattern she wants to apply. We propose that a visual representation of the interdependencies, both positive and negative, could help users to make more rational decisions.

More specifically, we propose to adapt the type of visual representation provided by soft-goal interdependency graphs (SIG hereafter), originally envisaged to depict the problem space in terms of non-functional requirements [9]. In a nutshell, a SIG is a graph where the nodes are goals to be addressed and the edges are interdependency links amongst goals, which can reflect compositions relationships, contributions and trade-offs. This notation has been used to help in the design process of agent-based systems in [13]. In this case for each agent-oriented design pattern a "forces" diagram is built where it can be visually analyzed how that pattern affects to each requirement. In our case, we aim to build a representation where all design patterns are used to build up the problem space, in which the problem or intent pieces of the patterns are used to populate the nodes and edges.

3. VISUAL REPRESENTATION OF PATTERNS

The goal of this work is to assist novice web designers, a category that includes end users and casual developers, to select the patterns they need for a specific design project, being aware of the complexity of the patterns and also of the relationships and trade-offs each has. To meet this objective we will use patterns to provide users with a visual representation of the problem space in the form of a SIG that we call visually enhanced and interactive SIGs (VEISIG). The next few sections describe the pattern language we are using and the visual representation proposed.

3.1 Hyperpatterns, a pattern language for web design

As the pattern language we will use HyperPatterns [14] a pattern language for web design that is based on existing design patterns reported in the literature. In this catalogue, some patterns are slightly modified to shorten the description of the problem in order to make them more readable; some also include new sections like images of application examples, in order to improve their comprehensibility.

Each pattern description includes nine sections: its *identifier*, *name* and *reference* to the original pattern; the *context*, that describes the situation leading to the applications of the pattern; the *intent*, that describes in a very short sentence the problem addressed by the pattern; the *solution* that consists of an image and a description of the proposed solution; the *discussion* that analyzes the implications of applying the pattern; *related patterns* that links to some related patterns; and *references* that links to the original source of the pattern as well as other sources used to improve the original pattern.

We selected this catalogue for two reasons. First, the patterns included in it are taken from existing pattern languages and all of them provide some kind of design rationale. Second, these patterns are formalized using an ontology and integrated within a software tool, AriadneTool, that makes it possible to translate them into design models as described in [15]. By leveraging the existing work on AriadneTool, the visually enhanced SIGs for web design patterns could eventually become a first step in automatic generation of a web software design.

3.2 VEISIG: A visual representation of design pattern languages

In this paper we propose the use of Visually Enhanced and Interactive SIGs (VEISIG) to provide a visual representation of design pattern languages that could be useful for novice web designers. VEISIGs are visually enhanced in that visual clues are used to highlight contributions and trade-offs that appear in the problem space when the user selects a specific goal. VEISIGs are interactive in that users will be able to select goals to expand and thereby to understand the magnitude of the problem space as well as to realize pattern relationships and interdependencies. Each goal is linked with its operationalization, the concrete design pattern that makes it possible to reach this goal. In this way, users are not expected to identify the patterns they need but to select the *goal* they want to meet, while also understanding the cost and complexity of the goal by visualizing the relations amongst goals. Patterns are automatically proposed according to the selected goals. Table 1 summarizes some questions that we assume may arise when a novice designer is trying to understand a pattern and its applicability. In our approach, users are expected to think in terms of goals but finally they are suggested a number to patterns to apply, those tied to the selected goals. Thus, these questions concerning patterns usage are directly addressed in our approach by the way information is organized and presented in the VEISIG.

Table 1. Some questions about the use of design patterns

Q1. Which is the pattern I need to solve my problem?
Q2. Will I create a new problem (solvable or not, more/less relevant for my purposes) if I apply this pattern?
Q3. Which is the cost of applying this pattern?
Q4. Am I missing other relevant issues not covered by the patterns?
Q5. How can I adapt the general solution suggested by the pattern to my problem?

Finding an answer to any of these questions using the interface provided by existing design patterns catalogues implies relatively deep knowledge about the patterns, a situation quite contradictory if we take into account that knowledge reuse should be addressed primarily to those who do not have that knowledge and sometimes cannot even express any kind of searching criteria. Our goal is to use visual representations of the patterns that could help novice designers by providing them enough and understandable information as to offer rational answers to questions Q1, Q2, Q3 and Q4 in Table 1. In order to deal with Q5 we rely on the formal representation of these patterns, which makes it possible to start a wizard that helps the user to convert a number of design patterns into design models of ADM [15].

Table 2. Elements of the VEISIG

ELEMENT	RELATION WITH THE PATTERN LANGUAGE
Goals	*Intent* part of the design pattern
AND/OR relationships	Composition relationships derived form the *solution* description and from the *related patterns* part
Contribution relationship	Positive contributions derived form the *solution* description and from the *related patterns* part
Hurt relationship	Trade-offs derived form the *solution* description and from the *related patterns* part
Break relationship	Insurmountable conflicts derived form the *solution* description and from the *related patterns* part
Operationalization	Whole pattern

Table 2 summarizes the main elements of a VEISIG. The basic idea is to represent the goals to be addressed in any web design as well as their relationships (composition, contribution, hurts) as a VEISIG. Goals are derived from the *intent* part of the pattern and they are organized in a hierarchical space respecting the structure of the patterns language, for which the composition relationship is used. However, hurt and contribution relationships can be established between goals belonging to different levels of the hierarchy, so that the final structure of the problem space is a graph. In figure 1 we can see two VESIGs corresponding to the first level of the hierarchy and the second level for a specific goal. In the figure we assume the classical notation for SIGs [9] so that the clouds are the design goals (that is, the *intent* part of our design patterns) and boxes with rounded edges are their operationalizations (that is, the whole design pattern description). AND relationships are lines; OR relationships are lines with two arcs; contributions are lines tagged with a plus sign; hurts and breaks are lines tagged with one or two minus respectively. In the example the second level corresponds to the "Guide the user through the information space" goal addressed by the design pattern [AN1] Multiple ways of navigate, an adaptation of the homonym pattern in [3].

Users can start with the first level of the hierarchy, which includes the root patterns in the language. In figure 1, the main goal "Designing a useful web site" is decomposed into six subgoals each of which corresponds to a design view: "Organize information according to the user needs" that help designers to structure the information space; "Guide the user through the information space", that helps designers to provide useful navigation tools; "Design an adequate user interface" that assist in defining the presentation features; "Improve the interaction with the system" used to specify interactive elements; "Provide for different kinds of users" that deals with personalization issues and "Ensure system security" where all patterns concerning security are grouped. Each of the six patterns associated with these subgoals are divided in turn into even more specific patterns which may also be further decomposed according to the structure of the pattern language.

From this general overview, users can select a goal to navigate to the next level of detail, thereby exploring visually the problem space underlying the selected goal. In this way they can begin to appreciate the complexity of the problem (question Q3) and realize which patterns can be applied (question Q1). Moreover as users browse the whole problem space, they can realize if there are issues not covered by the pattern language (question Q4).

When a goal is selected, a color code is used to show the effect of applying the corresponding operationalization into other goals in the VEISIG. Three colors are used based on the interdependencies amongst goals: green for contributions; orange for hurt relationships (trade-offs); and red for break relationships (conflicting goals). In the example, selecting "Users need direct access to some nodes" makes the goal "Ensure system security" which belongs to the upper level in the hierarchy, turn orange: when access to some nodes is restricted to specific users, the designer should consider that access rules should be respected regardless of the access tool used, including indexes or search engines. In this way, users can realize that if they are creating new problems with such a pattern (question Q2), and they see the pattern description selecting the corresponding operationalization, they can realize the cost of fixing the problem and determine which pattern to apply according to the priorities of their project.

4. CONCLUSIONS AND ONGOING WORK

The use of design patterns may reduce development effort if designers are able to reuse the design knowledge underlying the patterns. The main reason to apply design patterns is to quickly find solutions to recurrent problems, so that a designer can spend more time on thinking creative solutions to problems not yet covered by the patterns. In this poster we have described and ongoing work aimed at generating a visual representation of the problem space that is implied by a specific pattern language with a view to helping novice designers to make rationale choices.

Currently we are developing a software prototype that will support the pattern visualization and problem exploration approach we have introduced; in parallel we are developing an empirical evaluation procedure to assess the usability of this approach for end users and casual developers.

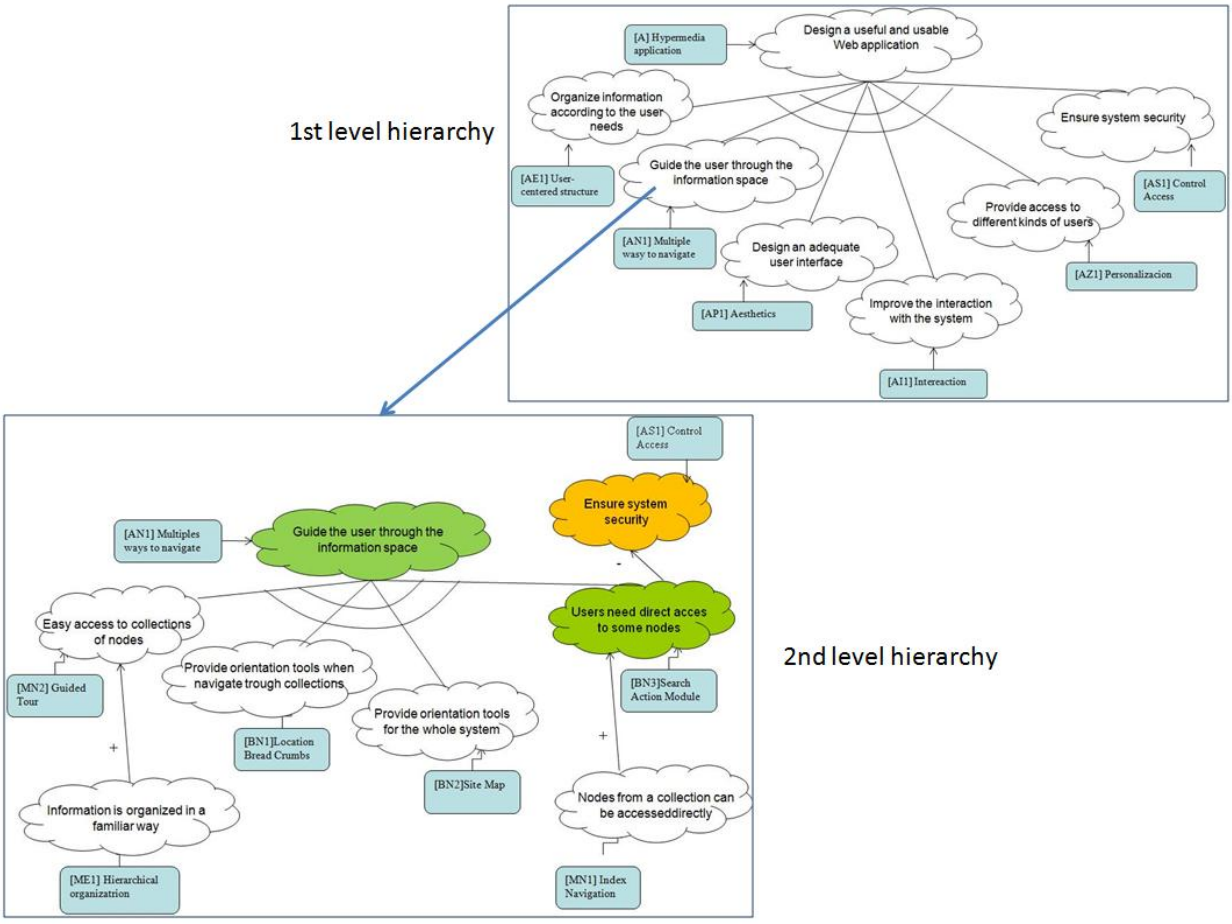


Figure 1. Excerpt from a VEISIG for web design

5. ACKNOWLEDGMENTS

This work is funded by the Spanish Ministry of Education through the grant (MEC PRY2007-0267) and the MODUWEB project (TIN2006-09678).

6. REFERENCES

[1] Gamma, E., Helm, R., Johnson, R. and Vlissides, J.M. 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.

[2] Alexander, C. 1979. The timeless way of building. Oxford University Press, New York.

[3] van Duyne, D.K, Landay, J.A. and Hong, J.I. 2002. The design of sites. Addison-Wesley.

[4] Blakley B. and Heath C. 2004. Security design patterns. Technical report.The Open Group,

[5] Graham, I. 2003. A pattern language for web usability. Addison-Wesley.Tavel, P.

[6] Bolchini, D. 2002. Hypermedia Design Patterns Repository. http://www.designpattern.lu.unisi.ch

[7] van Melie, M. 2006. Web design patterns. http://www.welie.com/patterns/

[8] Chung, E.S. Hong, J.I., Lin, J., Prabaker M. K., Landay, J. A. and Liu, A.L. 2004. Development and evaluation of emerging design patterns for ubiquitous computing. Proc. of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques. 233-242.

[9] Chung, L., Nixon, B. A., Yu, A. and Mylopoulos, J. 2000. Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers.

[10] Borchers, J.O. 2001. A Pattern Approach to Interaction Design. John Wiley & Sons

[11] Tidwell, J. 2006. UI Patterns and Techniques http://time-tripper.com/uipatterns/

[12] Kampffmeyer, H. and Zschaler, S. 2007. Finding the Pattern You Need: The Design Pattern Intent Ontology. In "Model Driven Engineering Languages and Systems ", LNCS 4735/2007, 211-225.

[13] Weiss, M. Pattern-Driven Design of Agent Systems: Approach and Case Study. Proc. of CAISE 2003

[14] dino2.dei.inf.uc3m.es/hyperpatterns

[15] Montero, S., Díaz, P. and Aedo I. (2007). From requirements to implementations: a model-driven approach for web development. European Journal of Information Systems. 16 (4), 407-419.

Memoria Mobile: Sharing Pictures of a Point of Interest

Rui Jesus[1,2], Ricardo Dias[2], Rute Frias[2], Arnaldo J. Abrantes[1], Nuno Correia[2]
[1]Multimedia and Machine Learning Group, Instituto Superior de Engenharia de Lisboa
[2]Interactive Multimedia Group, DI/FCT, New University of Lisbon
2829 - 516 Caparica, Portugal
rjesus@deetc.isel.ipl.pt, nmc@di.fct.unl.pt

ABSTRACT
This paper presents the Memoria mobile interface, an application to share and access personal memories when visiting historical sites, museums or other points of interest. With the proposed interface people can navigate the memory space of the place they are visiting and, using their camera-phones or Personal Digital Assistants (PDA), view what has interested them or other people in past occasions. The system consists of a retrieval engine and a mobile user interface that allows capture and automatic annotation of images. Experimental results are presented to show the performance of the retrieval mechanisms and the usability of the interface.

Categories and Subject Descriptors
H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *Evaluation/methodology*; H.3.3 [**Information Storage and Retrieval**]: Information Search.

General Terms
Design, Experimentation, Human Factors, Standardization.

Keywords
Mobile User Interfaces, Personal Memories, Multimedia Information Retrieval.

1 INTRODUCTION
Recent advances in mobile technology contribute to enhance the processes of capturing, sharing and storing personal pictures and videos. People can take photos or make small video clips of everything, everywhere and, through the World Wide Web share this information using, for example, Flickr or YouTube. The success of these two Web sites demonstrates that people like to share personal media not only with friends, but also with unknown people. Visits to historical sites, museums and other leisure activities are among the situations where most of these photos and videos are captured. Would people be willing to share personal pictures in these situations?

If individuals enjoy exchanging personal pictures with completely unrelated people on the Internet, it is very likely that they will also enjoy sharing images, when they are visiting a museum or an historical site, with the other visitors. Moreover, sharing this information is perceived as improving the quality of the photo collection that illustrates the visit and it becomes a fun ingredient of that experience.

This paper describes the Memoria mobile interface, a PDA application to capture, share and access personal memories composed by pictures or videos when visiting sites of interest. The paper presents the tests conducted in order to design the mobile user interface and to evaluate the multimedia retrieval system.

The paper is structured as follows. Next section presents the related work and the following gives an overview of the system. The subsequent sections describe the mobile application and the methodology used to design the interface. The paper ends with the evaluation of the interface and the conclusions and directions for future work.

2 RELATED WORK
The approach proposed for sharing and for navigation is supported by a mobile interface to manage personal memories composed by images. There are already available commercial applications to manage images in small displays e.g., HP Image Zone and ACDSee. Pocket PhotoMesa [1] is other interface to browse images in PDAs that employs Treemap layout to view hierarchies of image directories and provides zoomable interfaces for navigation. These interfaces allow browsing based on directory names or in manual annotations not in automatic annotation as we do. To search for personal pictures using automatic annotation, several systems were proposed using context information, including the MediAssist [2] which uses GPS information and time. To retrieve similar images using visual content and context information, in [3] was proposed a system that uses GPS information and visual features. Another system that also uses context and content information is proposed in [4]. This system automatically stores location, time, user data and the picture taken in a server. It also uses image features to recognize faces and to identify the picture location. Recently, it was proposed the Zurfer [5] system, an application to share images between nearby users. However, it does not use audiovisual information to search images as we do. Our work has similarities with some of these systems, but we also use audio information and different visual characteristics to retrieve images. The application interface proposes new features, namely an iconic

AVI'08, 28-30 May , 2008, Napoli, Italy

query system and the ability to share images when visiting sites of interest.

3 LOCAL SHARING SYSTEM

This paper presents a system to share pictures when visiting a point of interest. The local sharing system is based on a client-server architecture. Users can access personal pictures by means of a client program running on a mobile device where the tasks of capturing, annotating, visualizing and retrieving image files are done. Shared images are stored in a repository of memories placed in a server accessed via wireless connection (WLAN). Visitors can access these images by defining queries on the mobile application and submitting them to the server that includes an image retrieval system. Whenever the user takes a picture, GPS data and audio information related with some comments provided by the user are automatically annotated. Any time the user wishes, she can submit queries to the server. She can search for similar pictures, nearby pictures or retrieve images according to a user-defined context (e.g., images with people or with buildings). With this strategy, the database placed on the server (Memory of the place) is built by all the visitors that are willing to share their pictures. The retrieved images can be used to guide the visit and the personal collection is augmented by these images.

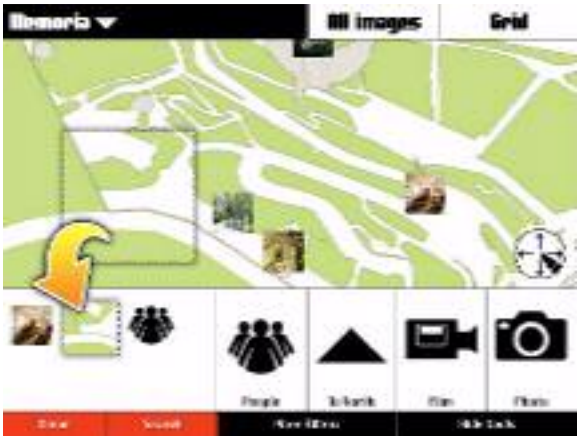


Figure 1. Query definition – Drag and drop of images, concepts, directions or map regions in the query box.

4 MOBILE APPLICATION

The proposed mobile interface is an application to manage personal images. This application has four functions: picture capture, automatic image annotation, picture visualization and retrieval of photos from a database. The interface contains (figure 1): a title bar, a status bar, a navigation bar, a toolbox, filters and a query box. The "More Filters" button shows all filters (e.g., concepts, directions) available to query the database. The "Query Box" is used to define queries by dragging and dropping filters into the container space (see figure 1). We chose this technique because it allows combining, as well as selecting different types of elements (e.g., images, text and buttons) to search for other images. Four types of elements can be dragged to define queries:

- Images, to retrieve similar pictures;
- Region maps, to search for local images;
- Directions, to look for pictures in a direction;
- Semantic Concepts, to retrieve images according to specific interests.– e.g., search for photos of human made objects, outdoor photos or photos that include people.

Other types of queries can be performed by combining these elements. For instance, it is possible to use an image to retrieve all similar images (using visual content) in the database but limit that query to a user defined region or to a particular direction. The following subsections describe the main functions of the application.

4.1 Annotation

The client application running on a mobile device allows it to be used as a regular camera, with the additional advantage that it captures GPS data (labeling the images with location information) and audio information when the picture is taken. Instants after the picture capture, the system opens the microphone to save some possible user comments. The idea is to record some comments that include extra information about the context or the nature of the picture. This audio information is converted to text using ASR (Automatic Speech Recognition) tools. Then, each image is annotated with the words that were recognized. Both types of annotation are used in the image retrieval system.

4.2 Visualization

Users can view the images results in three visualization modes: slide show, thumbnails grid (see figure 2) or spatial visualization in a map (see figure 1). The map visualization mode gives the visitor a greater sense of orientation since it displays the current position, the journey path as well as the images mapped to its locations (it exhibits spatial information and visual information simultaneously).

4.3 Retrieval

The mobile application relies on a multimedia retrieval system, which uses multimodal information to represent each picture in the database: visual features, GPS data and audio information annotated at capture time. Each image is represented by low-level visual features (color and texture) and a bag of features that expresses the number of times the visual "words" occur in an image. The visual words belong to a visual vocabulary obtained by applying the k-means method to a large set of visual descriptors (e.g., SIFT - Scale Invariant Features Transform) extracted from all images of the database. We use these representations to perform image queries. The low-level features are also used to train semantic concepts (e.g., indoor, outdoor, or manmade) in order to automatically annotate images.

Each image is also represented by a bag of words using the annotated words obtained from the audio information. Then, the Latent Semantic Analysis [6] and the cosine distance are used to rank the database. This information and the concepts trained using the visual features are used in the concept queries. The GPS data is used when a direction or a part of the map is dragged to the query box. The multimedia retrieval system used in this application is described in [7].

5 DESIGN

As initial user data we used facts and insights from previous mobile user interaction studies [8], ethnographic studies [9], and sociologic information [10] related with museum studies and tourism, as well as from the experience of designing the platform of a previous project [11] in a cultural heritage site. The data collected and the scenarios created were gathered in the form of textual descriptions and image boards. Then, several paper prototypes were created with which the interaction proposal was evaluated and optimized. With this proposal and subsequent documentation, a prototype in a PDA was implemented to test the most important features in the interface. The content and outcome of these evaluations are described below.

Figure 2. Grid visualization mode – To show a list of images.

5.1 Field Studies and Scenario Building

When designing the platform of a previous project [11] in a cultural heritage site, we gathered a significant understanding of the tourism domain along with the problems that characterize visits to an unknown leisure location. For instance, one problem that arises at this cultural heritage site, a luxury centennial estate, is that even though most visitors are provided with a detailed map, they are frequently unable to find directions and fail to go to significant spots that are not represented in the form of photographs.

5.2 Interface Design

The first step was to round up ideas and create an initial list of functionalities and related interactions, which were roughly put together on a high-fidelity static prototype interface in Adobe Photoshop (a visual composition with real content). This early visually detailed design was useful for several reasons: it reduced complexity in the visual interface, enabled early heuristic evaluation, improved the paper prototyping and allowed for an iterative aesthetic development.

With the high-fidelity static prototype as a basis we were able to create a paper prototype that emulated nearly all interface elements. The paper prototypes (see figure 3) were carried out in a series of sessions with potential PDA users with different profiles. During and following each evaluation, the user interface was further refined. After the PDA working prototype (described in section 4) had been implemented we performed additional tests with users.

Figure 3. Paper Prototype.

6 EVALUATION

This section presents and discusses the results of the tests performed to design the user interface. We also evaluate the multimedia retrieval system performance (see [7] for a complete evaluation).

6.1 User Interface Tests

We performed paper prototypes tests with four users aged 27, 26, 40 and 16 years old. None of the users was an actual PDA owner however they were mobile phone users, and in two cases were familiar with digital media applications. Each test consisted in three phases:

- To start, users were encouraged to explore the interface as they desired during 45 minutes. The goal of this phase is to analyse, if users understand the main purpose of the application.
- Next, they were asked to carry 3 tasks (retrieve images using the three types of query) using the paper prototypes and the PDA prototype. We include a PDA prototype at this stage because some of the features of the application are difficult to use in paper prototypes (e.g., the drag and drop technique).
- Finally, we made interviews - the users answered a set of questions about their experience using the interface.

We use a video camera to record all the tests. After each session we analyse the video data and some notes taken by the interviewer in the final phase of the session. Both prototypes were refined after each session. Therefore, the last test presents the best results. At the end of the four sessions all the data was analysed again to find common difficulties.

The four sessions showed that:

- Users understood clearly the main purpose of the interface - making queries to get more information (images) about a place;

- Novice users could easily utilize the common interactions in the interface (snapping photos, browsing images, handling menus, interacting with the map);
- All users considered that the functions in Memoria had interest in both dealing with personal memories and visiting leisure sites.
- Users failed the first attempts to make queries since they required a small period of time to discover how to use filters (the drag and drop feature).

At this stage the drag and drop is the feature that creates a significant usability problem in the interface. We do not have enough evidence to make a general statement but it seems that drag and drop components on small screens are hard to deal with and slow task performance, when compared to the simple click interaction. Nevertheless, after a few attempts, all users were able to "discover" the drag and drop interaction without assistance, with only one user "failing" to re-use it in the first queries afterwards (an occurrence exclusive of the paper prototype). Furthermore, during and following each evaluation the interface was refined and in the last session, the user (also the youngest) was able to perform better in all tasks, hence discovering the drag and drop feature much easier. Lastly, with the PDA working prototype, the users discovered the drag feature much faster than in the paper prototype, and never once forgot to use it afterwards. The ultimate decision to maintain the drag and drop interface was based on the effect it had on user's perception of the system model – having combined queries – which in our opinion, surpasses the mentioned disadvantages.

7 CONCLUSIONS AND FUTURE WORK

The paper presents a system for navigating and browsing in digital memories while at the physical locations (e.g., historical or cultural heritage sites) using mobile devices. The mobile interface is guided by a retrieval system that drives the user interaction using pictures shared by other visitors. The multimedia retrieval system uses visual content, audio information and GPS providing location information to retrieve related images with the ones the user is capturing.

The tests performed to evaluate the application shown the performance of the multimedia retrieval system and the interaction techniques used. The drag and drop of items to a query box was the interaction feature that created more difficulties to the user because it is not a common way to define queries. However, this technique seems to be a good choice to define queries in mobile devices (small displays) since it requires a minimal interaction from the user. We also described the relevance of the audio information in the performance of the proposed system. Nevertheless, we do not know if visitors are willing to make some comments about the picture taken.

The tests presented were performed using the Memoria Mobile application in the tourism context but it can also be used in all contexts (e.g., archeology or geology) where the knowledge shared by previous users can improve future experiences in a place.

This work will be extended to handle video which, in a way, can be done by using similar techniques to the ones that are being used for images, but can also benefit from the temporal properties of the video. Additional future work includes field usability tests, specifically with visitors at sites of interest.

8 REFERENCES

[1] Khella, and Bederson, B., "Pocket PhotoMesa: a Zoomable image browser for PDAs," *Proc. of the international conference on Mobile and ubiquitous multimedia* pp. 19-24, 2004.

[2] C. Gurrin, Jones, G. J., Lee, H., O'Hare, N., Smeaton, A. F., and Murphy, N. , "Mobile access to personal digital photograph archives," *Proc. of MobileHCI '05*, vol. 111, pp. 311-314, 2005.

[3] J. Lim, Chevallet, J. and Merah, S., "SnapToTell: Ubiquitous Information Access from Camera. A Picture-Driven Tourist Information Directory Service," *Mobile Human Computer Interaction with Mobile Devices and Services*, pp. 21-27, 2004.

[4] M. Davis, King, S., Good N., Sarvas, R., "From Context to Content: Leveraging Context to Infer Media Metadata," *Proc. of ACM International Conference on Multimedia* pp. 188-195, 2004.

[5] Hwang, A., Ahern, S., King, S., Naaman, M., Nair, R., Yang, J., Zurfer: Mobile Multimedia Access in Spatial, Social and Topical Context. in Proc. Fifteenth ACM International Conference on Multimedia (ACM MM 07), (2007).

[6] S. Deerwester, Dumais, S., Furnas, G., Landauer, T., and Harshman, R., "Indexing by Latent Semantic Analysis," Journal of the Society for Information Science vol. 41, pp. 391-407, 1990.

[7] R. Jesus, Dias, R., Frias, R., Abrantes, A., Correia, N., "Sharing Personal Experiences while Navigating in Physical Spaces," *5th Workshop on Multimedia Information Retrieval in 30th international ACM Information Retrieval Conf (SIGIR2007)*, 2007.

[8] A. Dix, Rodden, T., Davies, N., Trevor, J., Friday, A., and Palfreyman, K. , "Exploiting Space and Location as a Design Framework for Interactive Mobile Systems," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 7, pp. 285-321, 2000.

[9] D. Frohlich, Kuchinsky, A., Pering, C., Don, A., and Ariss, S. , "Requirements for Photoware," *Proc. of the ACM conference on Computer supported cooperative work* pp. 166-175, 2002.

[10] M. Levasseur, and Veron, E., "Ethographie de l'exposition," *Paris, Bibliotheque publique d'Information, Centre Georges Pompidou*, 1983.

[11] N. Correia, Alves, L., Correia, H., Morgado, C., Soares, L., Cunha, J., Romão, T., Dias, A. E., and Jorge, J., "InStory: A System for Mobile Information Access, Storytelling and Gaming Activities in Physical Spaces," *ACE2005 - ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, 2005.

An Eye Tracking Approach to Image Search Activities Using RSVP Display Techniques

Simone Corsato
Dip. di Informatica e Sistemistica
Università di Pavia
Via Ferrata, 1 - 27100 - Pavia - Italy
Phone +39 0382 985486
simocor@tele2.it

Mauro Mosconi
Dip. di Informatica e Sistemistica
Università di Pavia
Via Ferrata, 1 - 27100 - Pavia - Italy
Phone +39 0382 985486
mauro.mosconi@unipv.it

Marco Porta
Dip. di Informatica e Sistemistica
Università di Pavia
Via Ferrata, 1 - 27100 - Pavia - Italy
Phone +39 0382 985486
marco.porta@unipv.it

ABSTRACT
Rapid Serial Visual Presentation (RSVP) is now a well-established category of image display methods. In this paper we compare four RSVP techniques when applied to very large collections of images (thousands), in order to extract the highest quantity of items that match a textual description. We report on experiments with more than 30 testers, in which we exploit an eye tracking system to perform the selection of images, thus obtaining quantitative and qualitative data about the efficacy of each presentation mode with respect to this task. Our study aims at confirming the feasibility and convenience of an eye tracking approach for effective image selection in RSVP techniques, compared to the mouse-click "traditional" selection method, in view of a future where eye trackers might become nearly as common as LCD displays are now. We propose an interpretation of the experimental data and provide short considerations on technical issues.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications – *image databases*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *graphical user interfaces (GUI).*

General Terms
Performance, Experimentation, Human Factors.

Keywords
Image database, image presentation, image browsing, rapid serial visual presentation, eye tracking.

Conference'08, May 28-30, 2008, Napoli, Italy.

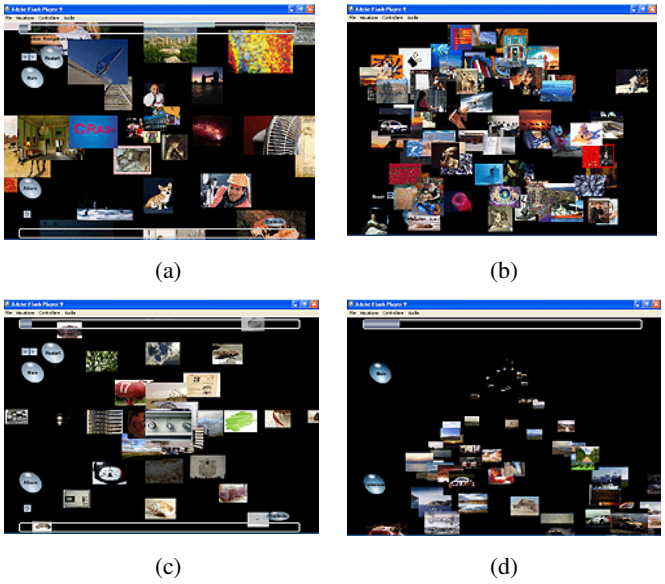
1. INTRODUCTION
Very often, we need to deal with large collections of images, and we want to select only some pictures according to certain criteria. For example, we may be interested in finding images with well-defined features, such as specific contents or technical properties; or, on the contrary, we may want to browse the database to search for something that only we can judge as suitable for our purposes. Quite common is also the case where the user simply desires to get some idea of the content of the picture database, like when rapidly riffling the pages of a book.

In the spatial domain, the most familiar visualization method is certainly the grid, in which pictures are arranged according to a matrix layout. In web pages and file folders, thumbnail images are usually displayed this way.

1.1 Rapid Serial Visual Presentation Modes
To achieve high search speeds, however, several dynamic visualization approaches have been proposed in the last years, among which those pertaining to the RSVP group deserve special attention. RSVP stems from Rapid Serial Visual Presentation and indicates a visualization mode where images are displayed in sequence, in the same location, for a short period of time (e.g. 100 milliseconds) [8]. A number of variants of RSVP have been proposed (also by our research group [4, 6]), which are more or less directly connected with it [1]. For instance, the *Floating* presentation mode is a time-dependent visualization technique in which small images appear about at the center of the screen and progressively enlarge, disappearing at the four sides (similarly to motorway signs which seem to move towards the driver). In our implementation (Figure 1a), to reduce image overlapping, pictures follow eight radial paths, and the angular distance between the directions of consecutive images is 135°. In the *Collage* display (Figure 1b), pictures appear very rapidly, in random positions, thus overlapping each other, like if being thrown onto a table. In the *Volcano* method (Figure 1c), images are "erupted" by the central "crater" of a virtual volcano, and slide down laterally along the virtual slopes, with a perspective effect; like in the Floating method, pictures follow eight radial paths. In the *Shot* display mode (Figure 1d), images, like "bullets", are "fired" by a virtual "gun" and progressively reach the lower part of the screen with a perspective effect.

While also other RSVP variants have been devised, in the experiments we present in this paper we focused on the above-described techniques for two major reasons. Firstly, our main

interest is in presentation modes able to display many images at a time (so that very fast visualization rates can be achieved with large image collections), and the four display methods considered satisfy such requirement more than others. Secondly, unlike other techniques, such approaches have as a common trait the fact of being characterized by image spatial distributions that occupy most of the screen area, thus potentially requiring eye-intensive screen exploration from the user.



(a) (b)

(c) (d)

Figure 1. Floating (a), Collage (b), Volcano (c), and Shot (d) display modes

1.2 Eye Tracking for the Evaluation of RSVP modes

Eye tracking can undoubtedly be a valuable source of information in the study of image presentation modes. The observation of eye scanpaths can in fact provide hints about the design of display methods, as well as suggestions about new interaction modalities. This is the reason why in our experiments we have considered both common efficiency indicators for search tasks (e.g. the number of correct pictures found within a set in a defined time period) and eye tracking data, obtained through the use of an unobtrusive eye tracker.

In the last decade, the potential of eye tracking for assessing RSVP methods has begun to be exploited. One of the first investigations of this kind was [3], where four visualization techniques were considered, namely Carousel, Collage, Floating and Shelf. Experiments, conducted with two testers who were asked to search a pre-viewed target image using the above-quoted approaches, had the main purpose to find correlations between eye gaze data and the trajectories of pictures. The tests showed that while the four display modes do not present specific perceptual problems, there may be differences among them for what concerns the effort required from users: methods in which images move may, in fact, cause some strain of the visual system.

Another interesting work, described in [9], investigated the relation between the space and time domains in display methods. The study considered the task of detecting the presence or absence of a previously viewed picture within a collection, using three modes: Slide Show (where 64 images were displayed in sequence, in the same position, at regular time intervals), Static (which was a static grid of pictures) and Mixed (a combination of the two previous modes, where 2x2 grids of images were displayed in sequence, in the same position, at regular time intervals). Since the testers expressed a strong preference for the Mixed presentation approach, which seemed also to be the one less prone to errors, eye tracking was used to try to better understand such an outcome. In particular, the hypothesis was tested that users tend to fix their gazes at the center of the four images of the Mixed mode, thus reducing the eye exploration extent while getting a "quick-glance" understanding of the images being displayed.

A more recent study [1] exploited the Slide Show, Mixed and Static presentation modes along with three other RSVP display techniques, namely Diagonal (images move diagonally from the upper left to the lower right corner of the screen), Ring (pictures appear at the center of the screen, rotate around it, and then disappear through the upper edge) and Stream (images flow along a hyperbolic trajectory, starting from the lower right corner of the screen and disappearing in the opposite corner with a perspective effect). Three tasks were considered: searching a pre-viewed target image, searching an image described in detail and searching an image described in general terms. Three different presentation rates were used. The study took into account such parameters to obtain, for each display mode, data about recognition rate and accuracy, as well as other indirect measures.

Our study focuses on the identification of all the images matching a textual description. Such images have to be selected as soon as they are identified, without interrupting the normal presentation flow. We want the identification times to be decoupled from the selection times: all the images should be selectable with the same (minimal) motor effort. This can be achieved by means of an eye tracking approach.

1.3 A Scenario for an Eye Tracking Approach to RSVP

In view of a future where eye trackers might become as common as LCD displays are now, we desire to find clear evidences that an eye-driven approach, besides being more natural [5], could really speed up search activities within very large collections of images.

We imagine a scenario where a graphic designer (the "user") deals with a collection of some thousands of images. According to certain criteria, he wants to rapidly reduce the number of images to a subset that can be reasonably managed later with more attention: for instance, he may want to pre-select all the pictures representing a cat. Quickness is here major concern: it is not important if some wrong images will be selected or if some appropriate ones will be missed, because a second, more accurate, selection will be later performed, based on other convenient criteria (for instance, to select a few images of lazy cats, suitable for a graphic project). This kind of research may be performed effectively also on small resolution images (as it happens on the web with stock photos).

Within our scenario, the user, after having spent few seconds for calibrating the eye tracking system, starts examining the rapid sequence of pictures displayed on the computer screen according to a RSVP mode. As soon as he identifies a proper picture, to select it, he just presses a key on the keyboard (or possibly acti-

vates a special sensor): the system marks the picture which corresponds best to the current user's gaze screen coordinates. After a session in which thousands of pictures have been displayed, the user can now concentrate in a small subset of few dozens pictures.

2. EXPERIMENTS

In the context of image search activities where the user is required to find pictures pertaining to well-defined categories within large databases, our hypothesis was that the described eye tracking approach for image selection in RSVP techniques could be feasible and convenient with respect to the mouse-click "traditional" selection method. Moreover, we expected to find significant differences in the efficacy of the chosen presentation modes, as suggested by researches mentioned in section 1.2, concerning similar tasks.

We tested our hypothesis by comparing the performances of a group of 31 students, all aged between 20 and 27. Each tester tried both the four considered dynamic RSVP methods (*Volcano, Floating, Collage, Shot*) with the eye tracking approach and a simple grid interface, with a point-and-click approach, which may be considered the present standard solution for this task. We stress again that the aim of this research was not to directly compare the four methods with the grid: a dynamic grid will be tested in future experiments.

Platform.

As an eye tracker, we used the Tobii 1750 [10], which integrates all its components (camera, infrared lighting, etc.) into a 17'' monitor. With an accuracy of 0.5 degrees and a relatively high freedom of movements, the system is ideal for simulating real-use settings, where it would be intolerable to constrain users too much in their activities. The device returns the *x* and *y* user's gaze screen coordinates, recorded by dedicated software 50 times a second. User interfaces for the tests were coded using Adobe Flash technology.

In our tests, as soon as a potential target picture was recognized in the four RSVP techniques, the user had to press any key on the keyboard. We didn't implement pictures selection entirely in real-time. Rather, we relied on data registered by the eye tracker, which include video files showing real-time scanpaths during the session and Microsoft Excel files reporting, among other information, fixation times and duration, as well as timestamps relative to 'key press' keyboard events. This way, it was possible to correlate 'a posteriori' images which were being looked at by the testers at a particular moment and their "conscious" selection action. In total, 155 clips were analyzed, each one lasting about 210 seconds.

Variables.

As it can be easily guessed, presentation speed in RSVP methods does influence the accuracy of search, as well as, of course, the total exploration time. However, in this preliminary phase of our activity we decided to limit the number of experiment variables, in favor of test sessions characterized by reasonable durations and well-defined comparable data. Before the actual tests, we carried out several pilot trials aimed at identifying the "optimal" presentation rates for each method, in terms of number of correct images found and subjective judgments about the chosen speeds ("too fast", "acceptable", etc.). We selected a presentation rate of a new picture every 105 milliseconds. This way, for each RSVP display mode, the presentation time for 2000 images was fixed to 3 minutes and 30 seconds.

Within the considered visualization techniques, pictures are displayed for different amounts of time, due to their different paths. They occupy different portions of the screen; they may overlap and they may progressively shrink or enlarge in different ways. Instead of devising sophisticated parameters to make the methods more directly comparable (such as, for instance, the integral of pictures area on the display time interval), we decided to separately "optimize" the different methods (in terms of picture sizes, paths and display time) during the preliminary phase in order to set the parameters of the tests. Figure 2 shows the average size (length of the diagonal, in pixels) and life-time (in seconds) of pictures as used in the experiments for the Floating, Volcano and Shot methods; for the Collage and the Grid methods the size is constant and the life-time is variable.

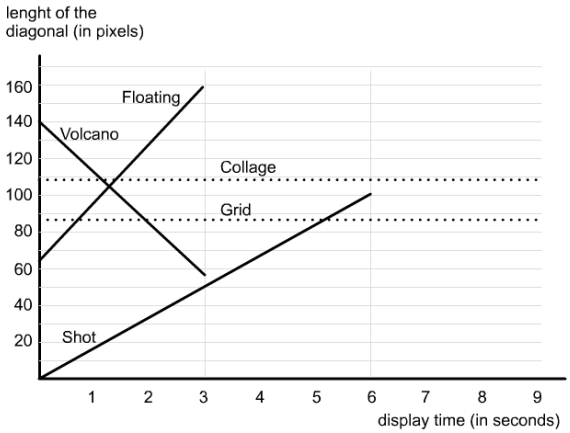


Figure 2. Average size (length of the diagonal, in pixels) and life-time (in seconds) of pictures used in the experiments for the different display modes

Experimental design.

After a short calibration procedure, necessary for the eye tracker to correctly understand where the specific tester is looking at, each user searched for target images using the four RSVP display techniques, plus the grid, in five different sessions.

Five different sets of 2000 images were employed. Each set contained 40 pictures pertaining to a specific target theme (namely cats, dogs, ships, planes and cars) and 1960 images with other content. Each presentation mode had of course a different set of pictures.

The testing order of display methods and their associated image sets and target themes varied among testers, so as to prevent results from being biased by learning effects, kind of target and possible user's mental fatigue. The total time necessary to introduce each participant to the experiment, explain the procedure, show examples and perform the five tests was about 50 minutes.

The purpose of the experiments was to find, in 3 minutes and 30 seconds, as many images as possible pertaining to a theme (40 out of 2000 in total). For the grid display, images were subdivided into 32 screens, arranged in 8x8 grids, and the user could move among them through 'next'/'previous' buttons.

At the end of each test session, users were also asked to express a subjective judgment about each method in terms of efficacy and fatigue, with values from 1 (lowest efficacy/fatigue) to 5 (highest

efficacy/fatigue). After the tests, data produced by the eye tracker were analyzed in detail, to extract both quantitative and qualitative data about the effectiveness of the display modes.

3. RESULTS AND DISCUSSION

Thanks to the devised approach, we have been able to compare the RSVP display methods according to "how easily" they allow image recognition (and selection), rather than on the basis of mouse clicks (which strongly depends on individual reaction times and hand/mouse spatial coordination ability). Main results are summarized in the following tables while the graphics in Figure 3 provide a clue about the distribution of the data.

Table 1. Average performances of testers by presentation method

	Floating	Volcano	Shot	Collage	Grid
num. of right pictures selected	29.51	28.00	19.29	19.35	15.58
num. of wrong pictures selected	1,87	2,06	2,35	1,64	0,74
wrong / right pictures (perc.)	7.12%	7.94%	16.72%	9.95%	6.25%

Table 2. Average eye movements for testers by presentation method (monitor resolution: 800x600)

	Floating	Volcano	Shot	Collage	Grid
scan path length per minute (in pixels)	13,462	14,810	25,783	36,349	31,868
aver. duration of fixations (msec)	274	307	182	187	181
aver. duration of saccades (msec)	46.5	50.1	60.0	56.2	59.3

Table 3. Average users feedback by presentation method. Subjective judgment in terms of efficacy and fatigue, with values from 1 (lowest efficacy/fatigue) to 5 (highest efficacy/fatigue)

	Floating	Volcano	Shot	Collage	Grid
efficacy (subjective)	3.97	3.70	3.26	2.42	2.16
fatigue (subjective)	2.52	2.71	3.13	3.90	1.74
efficacy / fatigue ratio	1,58	1,37	1,04	0,62	1,24

As can be noted from the previous tables, the results obtained do confirm our hypothesis according to which a selection method based on eye tracking is practicable and convenient compared to the state-of-the-art mouse-click approach. During the 3 minutes and 30 seconds allotted to each session, in the test with the grid users were able to freely control the presentation rate (screen change), but most of the time they did not succeed in examining all the images: on average, in fact, only 51,5% of the 2000 pictures was inspected, against 100% of RSVP methods. Of course, using the grid only few images were erroneously selected, since they were still. The ratio between wrong and correct images was very close to that of the Floating and Volcano techniques, which, however, allowed almost twice the number of images to be inspected. Also, users judged the "traditional" interface less effective than the others (although less tiring).

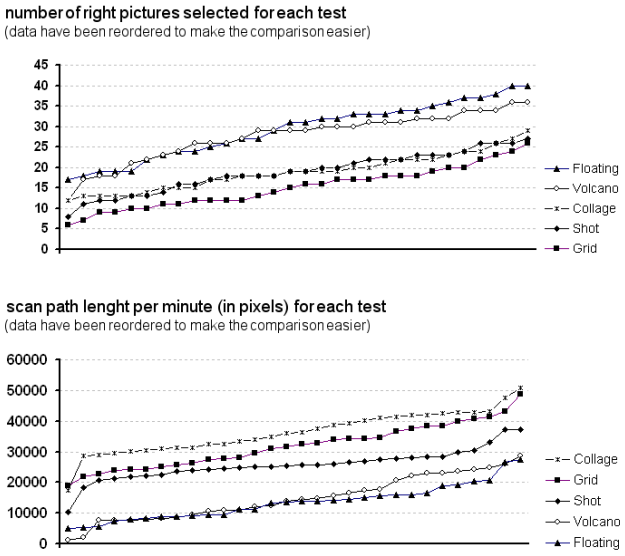


Figure 3. Number of right pictures selected and scan path length per minute for each test

To date, the study has demonstrated the viability of the eye tracking approach. We expect that also a display mode based on a dynamic grid (the *Tile* method described in [1]) could allow good performance using this new approach, due to the fact that pictures remain still. Our future experiments will just compare the Tile technique with the best among the four RSVP methods considered in this study.

According to our measured results and to the opinion of the testers, the Floating method has emerged as the most promising one, that is the most effective, efficient and satisfactory. Using the devised "visual selection", the performance of the Volcano mode is similar to that of the Floating technique, while in an interface based on mouse-click the selection of rapid-moving targets which get smaller and smaller would be rather demanding (Fitts' Law).

Data about eye-gaze behaviour confirm that the effort required from the observer is different in the various methods, and that such effort is related to the effectiveness of the methods themselves. The Floating and Volcano techniques, in fact, besides allowing better performances, are characterized by shorter average saccade times (even if this is only a rough figure) and shorter scan paths. Gazeplot and hotspot graphs generated by the eye tracking system prove that the Floating and Volcano modes are visually less "disorganized".

Currently, we are still examining the huge amount of data obtained to identify possible correlations among image size, motion speed and recognition time.

4. IMPLEMENTATION ISSUES

The implementation of an image search system based on eye tracking like the one proposed in this paper surely poses some technical issues.

The main problem is due to the difficulty of automatically identifying the image to be selected when another picture is very close to it, or even partially overlapped. The observation of key press

times within the recorded video clips has allowed us to correlate gaze positions to the actual "will" to select an image. A method which simply selects the image that is closer to the gaze center would be rather error-prone, as shown in Figure 4.

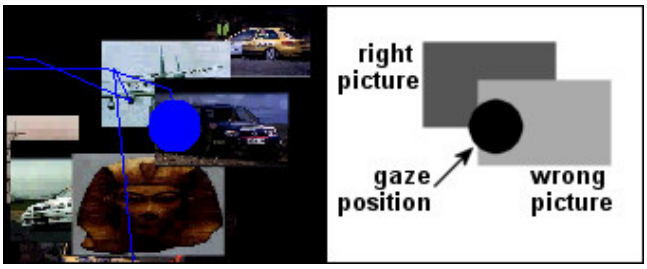


Figure 4. An example of image overlapping in the Shot display mode

However, the error probabilities could be reduced considering the user's eye-gaze behaviour. For instance, as shown in Figure 5, we have noted a tendency, more marked in some subjects, to anticipate the key press before the gaze is fully centered on a target picture.

Each column in Figure 5 shows the behaviour of a different tester while performing 5 selections (within each column, the different segments have been reordered by type). A black segment (type A) means that the corresponding keystroke occurred (more than 66 msec) before the gaze position was over the selected picture, which happens in about the 30% of the observed cases. Gray segments (type B) represent keystrokes which happen slightly prematurely (less than 66 msec before the "right" time, 9.7 %). Light grey segments (type C) represent "punctual" keystrokes (about 60%) . Less than 1% keystrokes were delayed (type D).

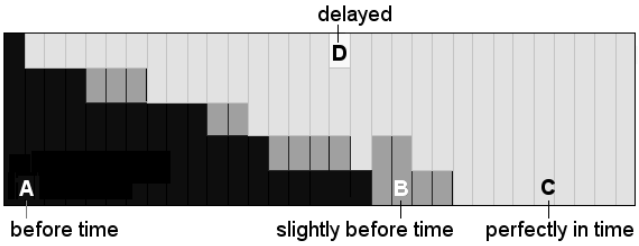


Figure 5. User's behavior: keystrokes occur before the eye gaze is centered on the target image in 1/3 of cases

Thus, while solutions without overlaps seem more promising, there are optimizations which may help to reduce ambiguities in display techniques that imply potential image conflicts.

5. CONCLUSIONS

In this paper we have considered the problem of searching well-defined target images within very large image databases. Eye tracking has been used to compare four RSVP display methods each other and with the "traditional" grid layout. Through tests involving 31 users, we have collected a huge amount of data, which we have now started to analyze. For example, we have discovered that the Floating and Volcano techniques are better than the Collage and Shot modes, in terms of number of correct images found, length of the eye path and user judgment. Indeed, the inspiring motivation of our study was the conviction that, for image selection, an eye tracking approach can be more efficient than the usual point-and-click mouse-based solution. The results of our investigations do confirm our hypothesis.

6. ACKNOWLEDGMENTS

This work has been supported by funds from the Italian FIRB project "Software and Communication Platforms for High-Performance Collaborative Grid" (grant RBIN043TKY).

7. REFERENCES

[1] Cooper, K., De Bruijn, O., Spence, R., and Witkowski, M. 2006. A Comparison of Static and Moving Presentation Modes for Image Collections. In *Proc. of AVI 2006*, Venice, Italy, May 23-26.

[2] De Bruijn, O., and Spence, R. 2000. Rapid Serial Visual Presentation: A space-time trade-off in information presentation. In *Proc. of AVI 2000*, Palermo, Italy, May 23-26, 189-192.

[3] De Bruijn, O., and Spence, R. 2002. Patterns of Eye Gaze during Rapid Serial Visual Presentation. In *Proc. of AVI 2002*, Trento, Italy, May 22-24, 209-214.

[4] Demontis, G., Mosconi, M., and Porta, M. 2003. Experimental Interfaces for Visual Browsing of Large Collections of Images. In *Proc. of HCI International 2003 (HCI '03)*, Crete, Greece, June 22-27.

[5] Oyekoya, O. K., and Stentiford, F. W. M. 2004. Eye Tracking as a New Interface for Image Retrieval. *BT Technology Journal* (Springer), Vol. 22, N. 3 / July, 161-169.

[6] Porta, M. 2006. Browsing Large Collections of Images through Unconventional Visualization Techniques. In *Proc. of AVI 2006*, Venice, Italy, 23-26 May.

[7] Simonin, J., Kieffer, S., and Carbonell, N. 2005. Effects of Display Layout on Gaze Activity During Visual Search. In *Proc. of INTERACT 2005 (13th International Conference on Human-Computer Interaction)*, Rome, Italy, September 12-16, 1054-1057.

[8] Spence, R. 2002. Rapid, Serial and Visual: a presentation technique with potential. *Information Visualization*, 1, 1, 13-19.

[9] Spence, R., Witkowski, M., Fawcett, C., Craft, B., and De Bruijn, O. 2004. Image Presentation in Space and Time: Errors, Preferences and Eye-gaze Activity. In *Proc. of AVI 2004*, Gallipoli (LE), Italy, May 25-28, 141-148.

[10] Tobii Technology AB 2003. Tobii 1750 Eye-tracker (Release B), November '03.

[11] Wittenburg, K., Chiyoda, C., Heinrichs, M., and Lanning, T. 2000. Browsing Through Rapid-Fire Imaging: Requirements and Industry Initiatives. In *Proc. of Electronic Imaging 2000*, San Jose, CA, USA, January 23-28, 48-56.

[12] Wittenburg, K., Forlines, C., Lanning, T., Esenther, A., Harada, S., and Miyachi, T. 2003. Rapid Serial Visual Presentation Techniques for Consumer Digital Video Devices. In *Proc. of 16th ACM Symposium on User Interface Software and Technology (UIST '03)*, Vancouver, Canada, November 2-5, 115-124.

Advanced Interfaces for Music Enjoyment

Adriano Baratè
Laboratorio di Informatica Musicale
Università degli Studi di Milano
Via Comelico, 39 – 20151 Milano
+39 02 50316382
barate@dico.unimi.it

Luca A. Ludovico
Laboratorio di Informatica Musicale
Università degli Studi di Milano
Via Comelico, 39 – 20151 Milano
+39 02 50316382
ludovico@dico.unimi.it

ABSTRACT
Music enjoyment in a digital format is more than listening to a binary file. An overall music description is made of many interdependent aspects, that should be taken into account in an integrated and synchronized way. In this article, a proposal for an advanced interface to enjoy music in all its aspects will be described. The encoding language that allows the design and implementation of such interface is the IEEE P1599 standard, an XML-based format known as MX.

Categories and Subject Descriptors
H.5.1 [**User Interfaces**]: Graphical user interfaces (GUI)
H.5.5 [**Sound and Music Computing**]: Modeling

General Terms
Algorithms, Standardization, Languages.

Keywords
Music, XML, MX, multimedia, synchronization.

1. INTRODUCTION
Describing music in all its aspects can be a challenging matter. For example, pop songs are usually distributed in form of tracks (audio content), nevertheless those music pieces are based on a score (symbolic content), have their own lyrics (text content), can be associated to a video clip (video content), etc.; besides, for each of those multimedia categories, many descriptions are allowed: for instance, the radio, the "unplugged" and the live version of audio tracks, the official video clip and video recordings of a live concert, etc. Of course, even more complex examples could be cited: for an opera, also sketches, fashion plates, on-stage photos, playbills constitute descriptions of music from a particular point of view.

In other words, in order to describe a music piece, many complementary multimedia objects can be used. Heterogeneity is involved from two different standpoints: i) the number of different multimedia descriptions (metadata, music symbols, text, still graphics, audio, and video), and ii) the number of different objects belonging to each category. As a matter of fact, institutions oriented to music performance - such as *Teatro alla Scala* - or to audio archives - such as *Discoteca di Stato* (the Italian national discotheque) - preserve not only the material belonging to their mission (scores and audio supports, respectively), but also a number of related objects, in order to provide the most comprehensive description of music pieces.

If heterogeneous music-related documents are available, the definition of a multimedia and multimodal environment is particularly interesting for a user who wants to investigate a music work from a number of perspectives, ranging from score analysis to performance and interpretation comparisons.

The advanced interface proposed in this work has two goals: first, it is aimed at demonstrating the integrated and synchronized description of music within a single XML-based file; besides, it provides an example of future ways to enjoy music at different degrees of comprehension and abstraction.

Avi'08, May 28-30, 2008, Napoli, Italy

2. MX (IEEE P1599)
The design and the implementation of the interface is based on an XML format named MX, an acronym which stands for *Musical application using XML*. MX international standardization is in progress, and its development follows the guidelines of IEEE P1599, *Recommended Practice Dealing With Applications and Representations of Symbolic Music Information Using the XML Language* [1]. This project proposes to represent music symbolically in a comprehensive way, opening up new ways to make both music and music-related information available to musicologists and performers on one hand, and to non-practitioners on the other. Accordingly, its ultimate goal is to provide a highly integrated representation of music, where score, audio, video, and graphical contents can be appreciated together. For further details please refer to [2] or to the on-line documentation at http://www.mx.dico.unimi.it.

3. REPRESENTATION OF MUSIC IN MX
MX provides a framework for a comprehensive description of music. It is based on two key issues: i) richness in the kinds of description related to the same music piece, ranging from symbolical and logical to analytical and media descriptions, and ii) the possibility to link a number of instances for each media type, i.e. a number of media objects of the same type.

These requirements can be satisfied thanks to the MX's multi-layer structure, made of 6 levels: *General*, *Logic*, *Structural*, *Notational*, *Performance*, and *Audio*. This structure has been proposed and studied in [3]. Each layer aims at describing a music piece from a different and complementary perspective. The

General layer contains catalog metadata about the piece and its XML encoding. The *Logic* layer describes the music piece in terms of symbols, such as notes, rests and other music signs. The *Structural* layer provides an environment to define music objects and to describe their relationships, such as in harmony or formal analysis. The *Notational* layer describes the digital objects containing graphical representations of scores, typically graphic files. The *Performance* layer allows to link and synchronize computer-driven performance formats, such as MIDI or SASL/SAOL. Finally, the *Audio* layer contains audio and video recordings.

It is worth to stress the presence of synchronizable and non-synchronizable objects within a single MX file. Audio, video and still graphics usually belong to the former family, whereas catalog metadata fall into the latter. Where a number of homogeneous or heterogeneous synchronizable objects are available for a given music piece, MX implements mechanisms to provide full synchronization. In this way, it is possible to enjoy music in a highly integrated environment where a cursor highlights the current chord in the score and simultaneously the corresponding point in an audio track is playing. Similarly, it is possible to switch from a score version to another, or from an audio performance to another in real-time, while the music is being played. Of course, this feature is not available for non-synchronizable objects such as metadata (track title, author, …) or music-related material (on-stage photos, sketches, playbills, …).

4. DESIGN OF THE INTERFACE

The design of the interface follows guidelines which strictly depend on the definitions and the key issues provided in the previous sections.

First, heterogeneity in music contents should find a counterpart in the layout of controls and views. Players, panels, floating windows or other devices should be used to present multimedia contents in a unique framework. A simple way to view and navigate complex contents consists in keeping different multimedia types separated by using different controls, and grouping a number of objects of the same type within the same control. As a matter of fact, homogeneous media types require similar controls and imply similar behavior, so this approach proves to be both user-friendly and effective. For instance, the part of the interface dedicated to audio/video contents should contain the playlist of such media objects (dynamically loaded from the MX file) and the usual controls of a media player. On the contrary, the panel dedicated to score images should contain the list of scores and pages of each score (dynamically loaded from the MX file) and image-oriented navigation tools. Please note that the simultaneous presence of all the six layers is not required for a generic music piece: a jazz piece could present no traditional score, as well as a never performed music work could be described from a symbolic point of view only, without any media attached. As a consequence, also the corresponding controls of the interface should be dynamically shown or hidden according to the characteristics of the encoding.

Besides, our comprehensive approach assigns the same dignity to all the forms of music description, thus the interface should present no "privileged" media type. Nevertheless, from a user-oriented standpoint, it is preferable to have a main window where a given media is shown with greater evidence. A solution to this dichotomy could be allowing any media to be played on the main window, as well as to be resized to a secondary view panel.

Full synchronization among synchronizable objects should be provided. As a consequence, the interface should allow the simultaneous enjoyment of all the views involved in the representation of media objects. A problem could occur with objects belonging to the same media category: for instance, following music simultaneously on pages belonging to different score versions could be difficult, but not impossible for a human user; on the contrary, listening to many performances, each with its own absolute temporization of music events, is confusing and difficult to implement. Finally, also non-synchronizable descriptions should be accessible, but in this case layout requirements are less problematic.

As stated before, a comprehensive music description can be performed by using six layers. In the following, we will discuss the design of an interface through a layer-by-layer approach.

4.1 The General layer

In this articulate approach to music description, the *General* layer contains catalogue metadata and additional information about the music work. Music entities are not directly involved here, neither as graphical nor as audio objects; moreover, the *General* layer does not convey symbolical or structural information. This section mainly contains metadata such as the title of the piece, the name of its authors and their role, the catalogue number (if any) and so on. The type of information presented here is typically text-based and non-synchronizable, thus *ad hoc* players are not required. Usually, titles and authors are shown in the title bar of the main window, and – if other information is required – a popup control could provide a more detailed view. In addition to catalogue metadata and other text contents, the *General* layer could host the description of digital objects which do not describe music itself but related aspects: sketches and fashion plates for an opera, on-stage photos and playbills for a rock concert, etc. In this case, an *ad hoc* player is required, above all if related contents should be somehow synchronized to music events. For example, during a duet between opera soloists, the photos of the performers or the pictures of their characters could appear as soon as they start singing and disappear when they stop, providing a graphical representation of the duet. A comprehensive interface for music enjoyment should support even this sort of "music-driven slideshows".

4.2 The Logic layer

In the *Logic* layer, music contents are represented in symbolic format. As regards traditional pieces, this section contains the description of notes, rests and other music signs. The encoding could be text-based (e.g. DARMS, GUIDO, Plaine and Easie Code), binary (Sibelius 5, MakeMusic Finale 2008, NIFF), or XML-based (e.g. MusicXML, MusiXML, MX). Please note that no absolute information about timing is provided here, as this aspect depends on specific performances and it is treated in the *Audio* layer. The *Logic* layer can be displayed by parsing the symbolic description and providing the corresponding layout. This latter aspect could imply a traditional representation – namely a score – or a "revised" version aimed at stressing inner properties of the piece. Examples of non-traditional representations are the typical view of note pitch and duration

provided by MIDI sequencers, and experimental layouts coming from musicological studies [4].

4.3 The Structural layer

The *Structural* layer investigates the musicological aspects of the work. In this section, music objects can be identified and put in relationship in order to justify the architecture of the piece. The locution *music object* is intentionally vague, so that any aggregation of music entities with distinctive features and a common meaning can fall under this definition. For instance, chords can be considered music objects built as vertical sets of notes, and their structural relationships produce harmonic grids and harmonic paths. Other examples of music objects are represented by melodic sequences, which can constitute themes and music subjects; in this case, discovering their relationships brings to considerations about the form and the architecture of the work, at various degrees of detail. A set of interfaces could be designed to underline structural relationships and to make them evident to the untrained user. An easy-to-implement proposal consists in using colors and geometrical shapes over traditional scores to characterize recurrent music objects or peculiar behaviors of parts and voices.

4.4 The Notational layer

The *Notational* layer contains information about graphical scores in form of digital objects. In a standard environment aimed at music enjoyment, this section is fundamental: in fact, it provides graphical contents to be synchronized. When traditional scores are involved, their scans can be described here, so that the user can follow symbolic contents within the interface. A more interesting representation can take place when no traditional score is available. In this case, a more general approach to music description and representation, based on graphical objects instead of symbolic representations, can solve the score-following problem. The *Notational* layer typically describes synchronizable music objects: thanks to its absolute timing, any audio performance can drive a cursor that points the corresponding music symbols over the score. In notation software, the cursor is usually represented as a vertical line that embraces all the staves of a system and moves along the horizontal axis. However, a more detailed view can be provided by drawing a number of bounding boxes around the current events, so that: i) also vertical movements of melodic lines can be appreciated, and ii) score following can be filtered by parts/voices.

4.5 The Performance layer

The *Performance* layer contains symbolic codes aiming at computer-driven music performances. Once again, the materials that fall under this category belong to the family of synchronizable objects. Examples of in-use digital formats for performance information are MIDI and SASL/SAOL. Their nature intrinsically allows a number of different representations. If we consider the case of a MIDI file, usually it is performed in a media player, so that the user can enjoy its contents as a waveform; as a matter of fact, the original file actually does not contain audio information, but numeric instructions to allow a synthesizer (or a more complex audio chain) to produce audio information. From another standpoint, many MIDI-related software applications try to provide a symbolic view of MIDI contents, by interpreting pitches (namely frequency-based classes) and durations (expressed in MIDI ticks) in order to recreate traditional scores. Finally, software such as sequencers show MIDI information by some kind of graphical representation, like colored rectangles and circles disposed on a grid. All these forms represent effective visualizations for computer-driven performance information, and our comprehensive approach should take them into account. In the interface we propose, the notational approach (even if the score is not expressed, but somehow recreated) or the audio performance (even if the waveform is generated by some MIDI instrument) can rely on the dedicated players provided for the corresponding layers, namely the *Notational* and the *Audio* layers. The innovative contribution of this layer could be providing an alternative representation of music data, such as depicting chords as geometrical shapes, assigning given colors to pitch families.

4.6 The Audio layer

In general terms, the *Audio* layer addresses music tracks encoded in some digital format. First, music events in an audio file are synchronizable objects; moreover, from the implementation point of view, such contents drive the synchronization of the other layers, as they have the most strict temporization requirements. A basic interface to listen to music should provide the standard controls to play, pause and stop the current track. Other features, such as the possibility to adjust the volume, should be provided too. The same interface, with *ad hoc* extensions, can be adopted to include also video contents: e.g. videoclips, live concert recordings, movie scenes with a soundtrack, 3D-animations, etc.

5. A PROPOSAL FOR AN INTERFACE

After discussing the levels of abstraction and the different perspectives in music description, and after introducing a number of features for a comprehensive music-oriented visualization, now a generic implementation of the interface will be proposed and shown in Figure 1. The format for music files is MX, which has been described in Section 3.

In our approach, the layout is made of a number of floating panels which can be either enabled or disabled depending on the features of the piece, and can be either opened or closed depending on the user's needs. Each panel is dedicated to a specific kind of visualization. All the homogeneous objects, namely the material belonging to the same layer, are selectable in each panel designed to reproduce them. In a simple case, we could assume a 1:1 relationship between music layers and the corresponding panels. However, this would limit the representational capabilities of the interface for the following reasons. First, even a single layer can host various kinds of information. For example, the *Logic* layer describes music symbols (such as notes and rests) as well as lyrics; and the *Structural* layer can contain harmony grids as well as musicological analyses of music forms. The coexistence of such heterogeneous aspects within a single viewer would be difficult to design and implement. Besides, the same information can be described and represented in different ways. For instance, the performance data contained in a MIDI file can originate both a traditional score and an audio rendition; and an audio track itself can be played but also visualized through some graphic algorithm. As a consequence, the interface we propose includes more than the six players related to the corresponding layers. The goal is to fulfill all the user's needs and desires about a comprehensive

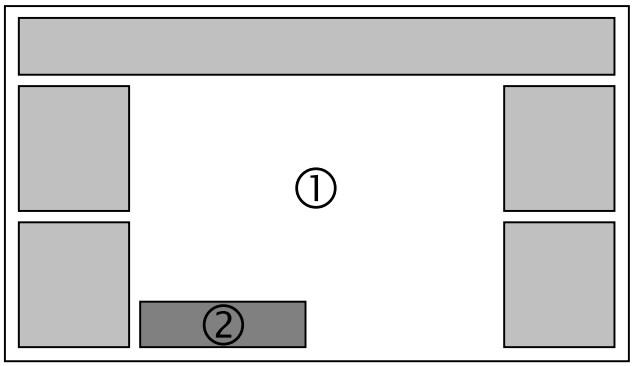


Figure 1. A proposal for the interface.

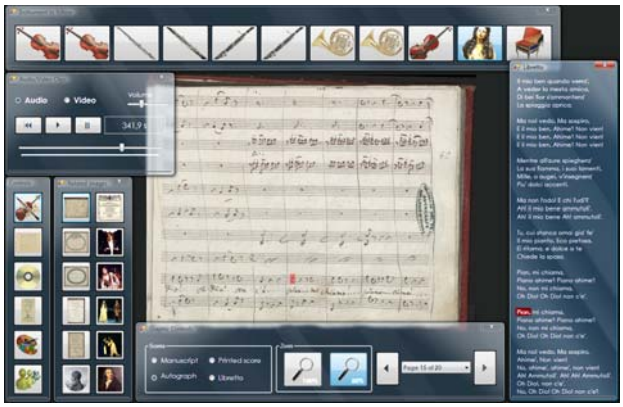


Figure 2. An implementation of the proposed interface.

representation of music and music-related information, both for a professional use or for untrained people.

Two panels are fundamental for the interface: the *main monitor*, denoted by ① in Figure 1, and the *meta-panel*, identified by ②. The purpose of the main monitor is to play graphic or audio/video materials in full-screen mode, thus allowing a better enjoyment of the contents. Among the standard uses of the main monitor we can cite showing the score with a running cursor, playing a related videoclip or providing other graphical representations of music contents, including slideshows. The meta-panel contains the controls to enable and show the other panels, when available. When the application is launched, the default behavior is the following: the MX file is parsed, and all the panels related to available materials are opened. Thus, the presence of each layer, and of the corresponding instruments, depends on the characteristics of the piece. Besides, the user should be allowed to close any panel, but the main monitor and the meta-panel, at any time.

Finally, let us recall the discussion about synchronizable and non-synchronizable objects. On the one side, the latter objects never conflict, which means that opening *n* panels with this kind of information do not generate errors or abnormal behaviors. On the other side, synchronizable objects should be accurately studied, in order to understand which situations bring to implementation or fruition problems. The possibility to view and play simultaneously contents from the same layer or from different layers derive from such considerations. For example, it is possible to follow two or more score versions simultaneously, even if the score scans all belong to the same layer. On the contrary, two simultaneous audio tracks can not be managed together, unless the timing characteristics of one of them is adjusted. Please note that even heterogeneous descriptions, coming from different layers, could conflict: this is the case of playing a MIDI file, described in the *Performance* layer and having its own event timing, together with an MP3 file, coming from the *Audio* layer.

6. CONCLUSIONS

The generic interface defined in this article can provide guidelines towards an integrated enjoyment of music. The proposed application represents the evolution of a number of earlier software demos and working applications developed by the LIM staff, as documented in [5] and [6]. The most recent implementation is N.I.N.A., standing for *Navigating and Interacting with Notation and Audio*. The interface is shown in Figure 2. The music piece chosen for this demonstration is the operatic aria "Il mio ben quando verrà", from Giovanni Paisiello's *Nina, o sia la pazza per amore*. This software was designed and implemented for the exhibition "Napoli, nel nobil core della musica" held in May 2007 at ResidenzGalerie in Salzburg, Austria. One of the purposes of the exhibition was that of making music tangible and visible, bringing together all five senses, beyond hearing.

7. ACKNOWLEDGMENTS

Our thanks to the LIM staff, who has contributed to the development of the MX standard and to the design of a number of XML-based music-oriented applications. The authors want to acknowledge researchers and graduate students at LIM, and the members of the IEEE Standards Association Working Group on Music Application of XML (P1599) for their cooperation and efforts. Special acknowledgments are due to Denis Baggi and Goffredo Haus for their invaluable work as working group chair and co-chair of the IEEE Standard Association WG on MX.

8. REFERENCES

[1] Baggi, D. 1995. Technical Committee on Computer-Generated Music. In: Computer, vol. 28, no. 11, 91-92.

[2] Baratè, A., Haus, G., and Ludovico, L.A. 2007. Music representation of score, sound, MIDI, structure and metadata all integrated in a single multilayer environment based on XML. Intelligent Music Information Systems: Tools and Methodologies, Idea Group Reference.

[3] Haus, G., and Longari, M. 2005. A Multi-Layered, Time-Based Music Description Approach Based on XML. Computer Music Journal, vol. 29, no. 1, 2005, 70-85.

[4] Hsu, K.J., and Hsu, A.J. 1990. Fractal Geometry of Music. Proceedings of the National Academy of Sciences of the United States of America, Vol. 87, No. 3, 938-941.

[5] Baratè, A., Haus, G., and Ludovico, L.A. 2006. An XML-Based Format for Advanced Music Fruition. Proceedings of the Sound and Music Computing Conference 2006 (SMC06), Marseille, France.

[6] Baggi, D., Baratè, A., Haus, G., and Ludovico, L.A. 2007. NINA - Navigating and Interacting with Notation and Audio. Proceedings of the 2nd International Workshop on Semantic Media Adaptation and Personalization (SMAP 2007), London, Great Britain.

Funky Wall: Presenting Mood Boards Using Gesture, Speech and Visuals

Andrés Lucero
Department of Industrial Design
Eindhoven University of Technology
5600MB Eindhoven, the Netherlands
a.a.lucero@tue.nl

Dzmitry Aliakseyeu
Media Interaction Group
Philips Research Eindhoven
5656AA Eindhoven, the Netherlands
dzmitry.aliakseyeu@philips.com

Jean-Bernard Martens
Department of Industrial Design
Eindhoven University of Technology
5600MB Eindhoven, the Netherlands
j.b.o.s.martens@tue.nl

Figure 1. Presenting the story behind a mood board using gestures, speech and visuals.

ABSTRACT
In our studies aimed at understanding design practice we have identified the creation of mood boards as a relevant task for designers. In this paper we introduce an interactive wall-mounted display system that supports the presentation of mood boards. The system allows designers to easily record their mood board presentations while capturing the richness of their individual presentation skills and style. Designers and clients can play back, explore and comment on different aspects of the presentation using an intuitive and flexible interaction based on hand gestures thus supporting two-way communication. The system records the presentation and organizes it into three information layers (i.e. gesture, sound and visuals), which are first used to segment the presentation into meaningful parts, and later for playback. Exploratory evaluations show that designers are able to use the system with no prior training, and see a practical use of the proposed system in their design studios.

Categories and Subject Descriptors
H.5.m [**Information Interfaces & Presentation**]: Miscellaneous.

General Terms
Design, Human Factors, Performance.

Keywords
Gesture-based interaction, wall projection displays.

AVI'08, May 28-30, 2008, Napoli, Italy

1. INTRODUCTION
Designers commonly use mood boards in the early stages of the design process [4], to explore, communicate, and discuss ideas together with their clients. These boards can be created with different types of media although designers usually use images to say something about the target audience, product, and/or company they are designing for. What may be easily overlooked, however, is that there is a story behind every mood board.

Once the mood board is completed, designers must communicate the story (and the ideas) behind the mood board. Usually designers will meet their clients to directly present, discuss and receive feedback on their mood boards. However, in large companies mood boards are uploaded to the company's Intranet so they can be experienced by and inspire different departments (e.g. design, marketing, sales, etc.). It is also common that clients and the design team itself are distributed over the globe, working in different time zones. Mood boards are then embedded in PowerPoint presentations and attached to an extra A4 text document that explains the mood board. In these cases, Intranet or PowerPoint presentation, the main question is, how can designers make sure that the right message is conveyed? Why was a given image chosen? What is the path through the mood board that the designer intended in order to tell the story? And equally important, how can clients reply and give feedback on what they are thinking? More generally speaking, how can we support presenting and receiving feedback for a mood board?

We propose an interactive system called 'Funky Wall' (Figure 1) that supports the presentation of mood boards by recording and keeping essential aspects of the presentation at three main information layers: gesture, sound (speech) and visuals. These information layers are first analyzed in order to segment the presentation in a meaningful way. Each segment is then associated with a specific time, interval and area on the mood board allowing designers and clients to experience the mood board, taking into account both time and space.

2. BACKGROUND

The field of human-computer interaction (HCI) has been investigating how people interact with computer systems at work (and more recently at home), trying to help them achieve their goals. Within HCI, researchers have already identified the potential behind interactive vertical surfaces as a more natural and familiar setting to address design (collaborative) interactions. The ID-MIX project [9] tries to assess the relevance and impact of augmented reality systems in design practices. The question the project addresses is if professional users (i.e. industrial designers) are willing to change their current work practices when confronted with alternative 'augmented reality' approaches. To gain a better understanding of design practice, we have conducted several user studies using diverse methods. By using probes in a professional context [10] we were able to identify a relevant task for designers: making mood boards. Subsequently, we have conducted contextual inquiries with Dutch industrial designers and interviews with Finnish fashion and textile designers to get a better understanding of why designers use mood boards and how they create them. In these studies, we have identified five stages of the mood-board making process: 1) 'collecting', 2) 'browsing', 3) 'piling', 4) 'building', and 5) 'presenting'.

3. RELATED WORK

Clark and Brenan [2] have extensively studied the relation between gestures and speech, and the role of gestures in human communication. Clark and Brenan argue that gestures together with communicative statements help establish common understanding, and that an appropriate gesture that is easily interpretable is preferable over complex sentence constructions. Gestures have also been widely explored as a natural way of interaction for a range of systems such as tabletop, vertical displays, multi-device environments, and 3D virtual environments. An example of a public display system that is controlled by gestures was presented in [13]. The authors aimed at studying shared, interactive public displays that support transition from implicit to explicit interaction. They used hand gestures and touch for explicit interaction, while body orientation and location played part in implicit interaction. A few systems employed gesture-based interaction in addition to speech, for either enriching the presentation process or to improve the communication with remote parties. The Charade system [1] allows presenters to use free-hand gestures to control a remote computer display, while also using gestures for communicating with the audience. Kirk et al. [7] studied different ways to represent gesture shadows (hands, hands and sketch, sketch only). They concluded that unmediated video representations of hands speed up performance without affecting accuracy. There is also a large area of research that looks at optimal meeting content capturing and browsing [5]. Many of these systems are based on the idea of Activity-based Information Retrieval, which proposes to use user activity (such as note-taking, annotating, writing on whiteboards) to index multimedia data and make data retrieval easier [8]. However only a few examples can be found where a speech plus gesture approach is used to enrich the capturing and (re)viewing of presentations. Ju et al. [6] use a motion estimation technique to detect key frames and segment the video (recorded presentation). Another example is the Active Multimodal Presentations [3] concept. The main difference with the Funky wall is that it addresses offline communication and attempts to create a structure using only implicit information (speech and gestures) for segmentation.

4. DESIGNING THE 'FUNKY WALL'

From the five stages of the mood-board making process, we have conducted exploratory studies in relation to 'browsing' [9]. We now focus our work on supporting the final stage, 'presenting', by designing a 'Funky Wall' that: 1) allows designers to easily record their mood board presentations while capturing the richness of their individual presentation skills and style, 2) allows both designers and clients to play back and explore different aspects of the presentation using an intuitive and flexible interaction involving hand gestures, and 3) supports two-way communication needed for successful mood-board design, by allowing clients to reply and share their thoughts on the mood board contents.

4.1 Proximity-Based Interaction

The 'Funky Wall' employs four different ranges of interaction (Figure 2) depending on the designer's proximity to the mood board: 'presenting', 'contemplating', 'replaying', and 'exploring'. Different functionalities are made available for each range. Gesturing close to the screen is used to record a presentation (<0.5m). When the presentation has been created, designers or clients can then 'contemplate' the mood board from a distance (>2m, no gestures), they can 'replay' the entire presentation (gesturing 1.5-2.m), or they can also 'explore' specific parts of the recorded presentation (gesturing 0.5-1.5m). Our four ranges of interaction resemble the ranges proposed in [13], and [11].

4.2 Intuitive & Flexible: Gestures & Speech

From our studies we have learned that for activities involving creation designers prefer working with their hands and with tools that allow flexibility and intuitive interaction (e.g. pencil and paper). To keep the interaction simple, designers can record their presentation by gesturing and explaining the mood board in front of the screen, using their hands to point or outline specific areas of the mood board. Preliminary observations show that location and speed of the gesture can be used to create meaningful indices, i.e., to associate the speech layer with a particular area of interest.

4.3 Two-Way Communication

A mood board is an idea development tool. During the mood-board making process, designers and clients have several rounds of discussions to reach agreement on the ideas being presented in the mood board. Therefore, for a successful mood-board design the tool should support two-way communication between designer and client. The 'Funky Wall' supports this iterative process by allowing designers and clients to provide input by creating a presentation and share their thoughts by providing feedback. For this type of communication to happen, two 'Funky Walls' are needed, one for the designer and another for the client.

5. INTERACTION TECHNIQUES

5.1 Presenting

To begin recording their presentation, designers simply need to gesture and speak next to the screen (<0.5m) (Figure 2a). The system displays white traces of the gestures made, as if designers were putting down a continuous flow of paint with their hands. To allow good visibility of the mood board the opacity of the white trace is set to 30%. The system captures and segments the speech and the natural hand movements made by the designer, creating associations between audio segments and gestures.

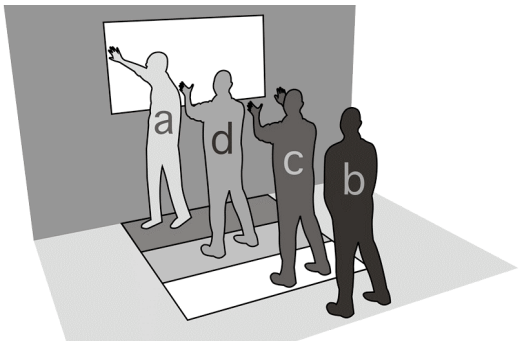


Figure 2. Interaction modalities revealed based on proximity. (a) 'Presenting' by gesturing next to the screen (<0.5m), (b) 'contemplating' the mood board (no gestures >2m), (c) 'replaying' the entire presentation (gesturing 1.5-2m), and (d) 'exploring' parts of the presentation (gesturing 0.5-1.5m).

5.2 Contemplating

Once a presentation has been completed, designers or clients can contemplate the mood board from a distance (>2m) (Figure 2b) for a comfortable overview. No gesturing is possible at this range.

5.3 Replaying

Spectators can replay the entire presentation by approaching the screen (between 1.5-2m) (Figure 2c). Raising the dominant hand results in displaying all gestures made during the presentation semitransparent on top of the mood board. Raising the non-dominant hand will trigger the complete recorded speech or audio explanation. By putting both hands together, both the recorded speech and the dynamic gestures unfold as the presentation progresses. Having an overview of all gestures allows spectators to quickly see areas of high interest where gestures concentrate.

5.4 Exploring

Taking one step closer towards the screen allows exploring specific parts of the presentation (between 0.5-1.5m) (Figure 2d). By pointing with the dominant hand to a given area in the mood board, users can view a static representation of the traces made in that area. These overlaid traces of gestures serve as guides for retrieval. Putting both hands together will display the dynamic gestures together with its corresponding spoken explanation. The mood board remains visible throughout the exploration process. To provide visual contextual feedback within the presentation, the tool highlights the explanations made by the designer just before and immediately after the current gesture. The previous gesture is shown in a lighter shade of white as if faded. The next gesture is displayed in black, as something that still needs to be discovered. Mood-board presentations last somewhere between 5 and 8 minutes. Therefore if the designer is unsatisfied with the results of the presentation, we propose that they present once again instead of providing a tool that allows editing specific parts.

5.5 Supporting Two-Way Communication

To truly support two-way communication, clients must be able to give designers feedback based on their perception and interpretation of the mood board. By having a similar 'Funky Wall' in their office, clients can explore the entire presentation (or parts of it), and later reply by adding their own comments to the mood board using the same interaction modalities described in 'presenting', 'contemplating', 'replaying', and 'exploring'.

6. EVALUATION

We conducted an exploratory user study of the 'Funky Wall' to test its usefulness and usability. First, we wanted to see if practicing designers would see the prototype as a relevant tool to present their mood boards. Second, we wanted to test the interaction techniques in terms of naturalness, ease of learning and use. We recruited five practicing designers with at least 5 years of experience. The participants varied in gender (1 female, 4 male), age (between 30 and 40), and preferred hand (4 right-handed, 1 left-handed). The evaluations were conducted individually.

6.1 Tasks

In the first part of the study participants created their own story for a mood board we gave them (approx. 5 minutes). Each participant was told that they would be using a system that tracked and displayed traces of their hand movements. In the second part participants explored an existing presentation using the system. Following a brief description of the interaction we allowed them to freely explore the functionality and get acquainted with the system (approx. 10 minutes). In the third part we asked them to walk us through their experience using the system (approx. 30 minutes per participant). All sessions were recorded on video.

6.2 Implementation

The system was set up using a desktop PC connected to a back-projection screen of size 2.0x1.5m (1024x768 pixels), as well as an ultrasonic tracking system – InterSense IS-600 used to track hands. During the sessions participants wore custom-designed interaction gloves that contained the sensors. The application was written in C# and used OpenGL for visualization purposes. The presentation and replay parts were fully functional. The analysis phase, where the presentation is segmented, was done manually.

6.3 Findings

6.3.1 Participants Agreed on the Principles

Designers were positive about the general underlying principles of the system to support the presentation of mood boards:

"This system helps you get the explanation the designer intended. I can experience the thoughts behind the images. Clients will have their own associations and thoughts behind these images." [P1]

"In case of long and complex presentations this allows you to have reminders of where certain parts were, like chapters. You see the entire mood board and you can zoom into parts." [P3]

6.3.2 Hand Gestures

In the first part of the study, designers were able to interact with the system with no prior training, especially liking the naturalness and simplicity of interaction. However, in the second part designers began to experience some difficulties when exploring:

"Bringing both hands together to trigger sounds is very uncomfortable. Maybe a quick movement in the air to press." [P4]

"I found it a bit difficult to navigate to the next and the previous thing. Maybe a flick of the wrist in a given direction to the sides should allow you to go forward and backwards." [P5]

6.3.3 Visual Feedback

Regarding the visual feedback provided by the system (i.e. traces of gestures on top of the mood board), designers first reflected on the amount of visual clutter, and had different opinions:

"At a certain point it is getting increasingly cluttered." [P4]

"The way the visual feedback is presented is done in a subtle way; it does not ruin the impression of the mood board." [P5]

Participants also commented on the helpfulness of playing back gestures dynamically as they heard the explanation:

"It helps to better explain the picture. It gives a touch of sensibility. It makes it easier to connect. Although you are not present, it seems that you are there. It is like a ghost of you." [P1]

"It really (makes it) much more alive. I can feel that the designer was there doing those gestures. It makes it more human." [P2]

Finally, designers reflected on the usefulness of having contextual feedback for the current, previous, and next speech segment:

"In traditional presentations you have no cues about what is happening. This is much more intuitive than just having a timeline or something similar because now you can actually see how things unfold temporally alongside the thematic unfolding." [P5]

7. DISCUSSION

7.1 Feasibility of the System

In our prototype the analysis phase, where the presentation is segmented, was done manually. The main reason for doing this was that the goal of the study was to first assess the potential usefulness and usability of such a system. However, based on the results reported in the literature and the analysis of gesture-speech synchronization automation, our system seems feasible [12].

For segmentation our system does not need to recognize speech, we only need to detect phrase boundaries. One way of detecting phrase boundaries is by using pauses (intervals of non-speech audio between speech segments) [14]. Stifelman [12] found that phrases could be robustly identified using a threshold of 155 ms; pauses shorter than the threshold are most likely pauses within a phrase while longer ones are pauses between phrases. The speed and location of gestures can also be used to make the segmentation more robust. In our study we observed that speed could be used to separate between explanations of specific parts (slow movements), connections between different parts (fast long movements), and the general discussion of the mood board (often fast short movements).

7.2 Using Other Media to Record and Replay

We believe the use of gestures enriches the presentation of mood boards by allowing designers to clearly express the feelings and the ideas contained. The same is applicable to the replaying and annotating of the presentation. However the latter part can also be done on any desktop system using a standard pointing device such as a mouse. In principle, the presentation could also be done on a desktop but we fear that the added richness will be lost.

8. CONCLUSION

We built and evaluated a system that supports designers in conveying the story behind mood boards in situations when face-to-face communication is not possible. The 'Funky Wall' allows designers to easily record mood board presentations while capturing the richness of their individual presentation skills and style. It also allows both designers and clients to play back, explore and comment on different aspects of the presentation using an intuitive and flexible gesture-based interaction. We have evaluated the system with professional designers in order to test its usefulness and usability. The results of the study showed that designers saw a practical use of the system in their design studios. Participants felt that the system gave them control over the presentation, so they could, with little effort, explore different aspects of the mood board. Moreover they felt that the combination of speech and traces of hand movements gives a touch of sensibility and makes it easier to connect with the message. Participants also liked the naturalness and simplicity of the interaction. Future work includes implementing a fully automated system for segmentation.

9. REFERENCES

[1] Baudel, T. and Beaudouin-Lafon, M. 1993. Charade: remote control of objects using free-hand gestures. *Communications 36*, 7 (1993), ACM Press, 28-35.

[2] Clark, H. H., & Brennan, S.E. Grounding in Communication. In L.B. Resnick, R.M. Levine, & S.D. Teasley (Eds.). Perspectives on socially shared cognition, (1991), 127-149.

[3] Elsayed, A. Machine-Mediated Communication: The Techology. In *Proc. ICALT'06*, IEEE.

[4] Garner, S., and McDonagh-Philp, D. Problem Interpretation and Resolution via Visual Stimuli: The Use of 'Mood Boards' in Design Education. *International Journal of Art & Design Education 20*, 1 (2001), 57-64.

[5] Geyer, W., Richter, H., Abowd, G. 2005. Toward a Smarter Meeting Record – Capture and Access of Meetings Revisited. *Multimedia Tools and Applications 27* (2005), Springer Science, 393-410.

[6] Ju, S. X., Black, M. J., Minneman, S., and Kimber, D. 1997. Analysis of Gesture and Action in Technical Talks for Video Indexing. In *Proc. CVPR '97*, IEEE (1997).

[7] Kirk, D. and Stanton Fraser, D. 2006. Comparing remote gesture technologies for supporting collaborative physical tasks. In *Proc. CHI '06*, ACM Press (2006), 1191-1200.

[8] Lamming, M. G. *Towards a Human Memory Prosthesis.* Technical Report #EPC-91-116 EPC-91-116. Rank Xerox EuroPARC, 1991.

[9] Lucero, A., Aliakseyeu, D., and Martens, J.B. Augmenting Mood Boards: Flexible and Intuitive Interaction in the Context of the Design Studio. In *Proc. TableTop 2007*, IEEE (2007), 147-154.

[10] Lucero, A. and Mattelmäki, T. 2007. Professional Probes: A Pleasurable Little Extra for the Participant's Work. In *Proc. IASTED HCI 2007*, Acta Press, 170-176.

[11] Prante, T., Röcker, C., Streitz, N.A., Stenzel, R., Magerkurth, C., van Alphen, D., and Plewe, D.A. Hello.Wall - Beyond Ambient Displays. In *Adj. Proc. UBICOMP 2003*, 277-278.

[12] Stifelman, L. *The Audio Notebook: Paper and Pen Interaction with Structured Speech.* Ph.D. dissertation, MIT Media Laboratory, 1997.

[13] Vogel, D. and Balakrishnan, R. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proc. UIST 2004*, ACM Press (2004), 137-146.

[14] Wang, M. Q. and Hirschberg, J. Automatic Classification of Intonational Phrase Boundaries. Computer, Speech, and Language, 6:175–196, 1992.

Toward a Natural Interface to Virtual Medical Imaging Environments

Luigi Gallo[1, 2], Giuseppe De Pietro[1], Antonio Coronato[1], Ivana Marra[1]

[1] ICAR-CNR, Via Pietro Castellino 111, 80131 Napoli, Italy

{gallo.l, depietro.g, coronato.a, marra.i}@na.icar.cnr.it

[2] Università degli Studi di Napoli "Parthenope", Via Amm. F. Acton 38, 80133 Napoli, Italy

gallo.l@uniparthenope.it

ABSTRACT

Immersive Virtual Reality environments are suitable to support activities related to medicine and medical practice. The immersive visualization of information-rich 3D objects, coming from patient scanned data, provides clinicians with a clear perception of depth and shapes. However, to benefit from immersive visualization in medical imaging, where inspection and manipulation of volumetric data are fundamental tasks, medical experts have to be able to act in the virtual environment by exploiting their real life abilities. In order to reach this goal, it is necessary to take into account user skills and needs so as to design and implement usable and accessible human-computer interaction interfaces. In this paper we present a natural interface for a semi-immersive virtual environment. Such interface is based on an off-the-shelf handheld wireless device and a speech recognition component, and provides clinicians with intuitive interaction modes for inspecting volumetric medical data.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques – *User Interfaces*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Input devices and strategies*; I.3.6 [**Computer Graphics**]: Methodology and Techniques – *Interaction techniques*; I.3.7 [**Computer Graphics**]: Three Dimensional Graphics and Realism – *Virtual reality*.

General Terms

Design, Human Factors.

Keywords

3D user interface, 3D interaction, Virtual Reality, Wireless, Medical Imaging, VTK.

AVI'08, May 28-30, 2008, Napoli, Italy.□□

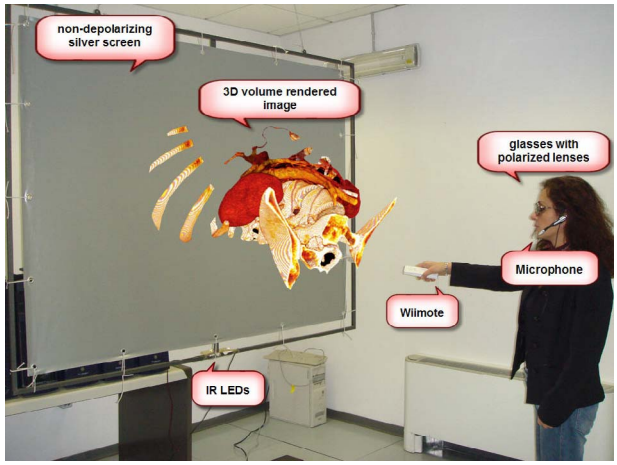


Figure 1. The semi-immersive 3D interaction.

1. INTRODUCTION AND BACKGROUND

In the daily clinical practice different post-processing techniques can be used to represent all the information related to anatomical scan data, acquired by CT, PET or TAC instrumentation [1]. Most Medical Imaging software applications, which support medical experts in the 3D reconstruction process of anatomical structures coming from DICOM images, are capable to offer a three-dimensional visualization of all volumetric data with a high degree of accuracy [2, 3, 4].

Several studies have demonstrated that the use of Immersive Virtual Reality (IVR) technologies can enhance the performance of clinician tasks [5]. However, even if some medical applications present immersive VR facilities for 3D models visualization, they usually do not provide the natural 3D interaction methods necessary to completely benefit from IVR environments.

In order to enhance usability, traditional input interfaces should be replaced with more user-friendly human-computer interaction interfaces. In particular, a 3D User Interface (UI) should allow the user to act in a way similar to real life, making it possible to exploit human abilities in a virtual environment [6]. Thus, to reach this goal, user has to interact through non-obstructive devices. Moreover, more than one mode of input should be used

to make the human-computer interaction more natural and intuitive.

In this paper we present a 3D interface for visualizing, manipulating and investigating volumetric medical data of real patients in a semi-immersive context. Users can interact with three-dimensional data through the use of the Nintendo Wii controller [7] combined with a speech recognition component. The developed interface has been implemented by using the Visualization Toolkit (VTK) library, and has been integrated into an open-source and cross-platform Medical Imaging ToolKit (MITO) [8].

2. THE WIIMOTE DEVICE

During the 3D interface design, we have interviewed several clinicians and medical students to better understand their needs in the inspection of medical data. According to their considerations, a suitable 3D UI should be:

- Wireless;
- Ergonomic;
- Suitable for a near-real-time interactivity;
- Suitable to implement a pointing feature and to rotate objects with 3 DOF.

After a brief research among all the off-the-shelf 3D user interfaces, we have chosen a wireless and economical input device: the Wiimote.

The Wiimote, alias Wii Controller or Wii Remote, is the primary controller for the Nintendo's Wii console. It weighs about 148 grams, its height is 14.8 cm where as the width is 3.62 cm with thickness of 3.08 cm. Thanks to its motion sensing capability, the Wiimote allows users to interact with and manipulate items on the screen by simply moving it in the space.

The easiness of use of this controller has made it very popular on the web. Many unofficial websites provide accurate technical information obtained by a reverse engineering process [9, 10].

In this section we briefly introduce the most important Wiimote features.

2.1 Communication

The Wiimote communicates via a Bluetooth wireless link. It follows the Bluetooth *Human Interface Device* (HID) standard, which is directly based upon the USB HID standard. It is able to send reports to the host with a maximum frequency of 100 reports per second. The Wiimote does not require authentication: once put in the discoverable mode, a Bluetooth HID driver on the host can query it and establish the connection.

2.2 Inputs

The controller movements can be sensed, over a range of +/- 3g with 10% sensivity, thanks to a *3-axis linear accelerometer*. So it is possible to detect the controller orientation in the space. Calibration data are stored in the Wiimote flash memory, and can be modified via software.

Another feature is the *optical sensor* placed in front of it, able to track up to four infrared (IR) hotspots. By tracking the position of these points in the 2D camera field of view, accurate pointing information can be derived.

There are 12 buttons on the Wiimote. Four of them are arranged into a directional pad, and the other ones spread over the controller.

2.3 Outputs

The Wiimote can send outputs by using three different modalities: switching on/off up to four blue LEDs; vibrating itself and emitting sounds.

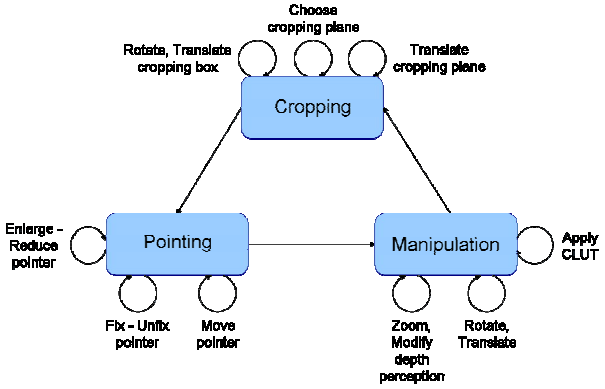
3. THE PROPOSED ITs

Usually, the interaction techniques needed to grant a natural interaction in a virtual medical imaging environment are not the same for a generic virtual reality application. In fact, interaction should take into account the "typical" user, the application, the task, the domain and the input device. So clinicians' requirements, alone, are not sufficient to reach a "natural" interaction.

Regarding applications requirements there are two different methods of operating in VR [11]: one for the industrial, architectural and art related applications, the other for clinical medicine or biomedical research. In the first method, the *Virtual Reality Space Method* (VRSM), the user is placed in a space to be navigated; since there is a large virtual space to explore, the navigation task is essential. In the second one, the *Virtual Reality Object Method* (VROM), there is only a data object to examine and manipulate and it is already within touching distance. For interacting in a virtual medical imaging environment, clearly a VROM method should be applied.

Even if there is only one object in the scene the selection task is still necessary. For a generic VR user there is only a single object in the scene, but for a clinician that object comprises of several different anatomic regions. So the selection task is not necessary to grab the object, that is always selected, but rather to "point to" a part of it. Obviously the manipulation task is essential. Once the reconstructed 3D object is visualized, clinicians want to accurately inspect it.

Following these considerations, we have identified the following features as necessary for interacting with volumetric medical data:



- **Object rotation / translation** – to visualize the 3D object

Figure 2. 3D interaction FSM.

from all possible points of view;

- **Pointing** – to point out a precise point of the visualized data;
- **Zoom in / out** – to better focus regions of data;
- **Control at run-time of the depth perception** – to move inward / outward the object in the virtual space;
- **Object cropping** – to cut off and view inside the data;
- **Apply CLUT** – to colour the object by applying various Color Look-Up Tables (CLUT).

The 3D interaction in a virtual medical imaging environment can be modeled by using a Finite State Machine (in Figure 2).

The system consists of three macro-states: the *Pointing* state, in which the input device is used like a laser pointer-style; the *Manipulation* state, where the object can be rotated simply by wheeling the input device in the real space; the *Cropping* state, in which the object could be cropped by arbitrarily moving six clipping planes in the 3D space. The user can switch among the states by pushing a button on the input device.

4. IMPLEMENTATION DETAILS

The proposed interaction techniques and the driver, implemented to use the Wiimote as a 3D UI, have been integrated into the MITO system. MITO has been written entirely in C++ and built on open-source and cross-platform libraries, which have been combined and extended to support medical imaging processing, semi-immersive 3D model visualization and interaction functionalities. The used libraries are: OpenGL – for fast 2D and 3D display; VTK (Visualization ToolKit) – for 3D rendering; ITK (Insight segmentation and registration Toolkit) – for image segmentation and registration; WxWidgets (Windows and X widgets) – for graphical user interface.

4.1 Integration of the Wiimote into MITO

The Wiimote uses the standard Bluetooth Human Interface Device (HID) to communicate with the host. Any Bluetooth host is able to detect the Wii controller as a standard input device. But, the Wiimote does not use the HID standard data type, only defines the length of its reports. This is the reason why standard HID drivers cannot be used.

We have developed several C++ classes to integrate the Wii Controller into MITO. The lowest level class is *HIDdevice*, whose methods allow fulfilling all the operations required to connect and to communicate with a generic HID device. The *wxWiimoteDriver* class, built upon this, makes it possible to create detailed wxWidgets custom events by exploiting its Wiimote state knowledge. All the generated events are handled by an event catcher, which is defined by the *wxEventCatcher* class. Finally the *appWxVtkInteractor* class converts wxWidgets events in the corresponding VTK events to modify the 3D scene.

More technical details on the Wiimote integration into MITO can be found in [12].

4.2 Interaction features

During the 3D interaction with the Wiimote, the system can be into three states: pointing, manipulation and cropping (see Figure 2). According to the interaction state, user actions have different effects (see Table 1).

The Wiimote motion sensing capability is differently interpreted in each state. In the cropping and manipulation states, when the user presses the B button, he can rotate the volume by whirling the input device. In these states, we use both the optical sensor and the accelerometer in order to determine the Wiimote position. In the pointing state, instead, only the optical sensor is used.

VTK handles the interaction with a 3D scene by using *interactor styles*. Mouse and keyboard events are handled in different ways depending on the interactor style that is active in the scene. In order to allow the Wiimote integration, we have implemented two new interactor styles, later included in the VTK package. The added classes are:

- *vtkInteractorStyleWiiJoystickStyle* – for a joystick style interaction (i.e. continuous rotation of the volume according to the Wiimote orientation);
- *vtkInteractorStyleWiiTrackballStyle* – for a trackball style interaction (i.e. rotation of the volume proportional to the Wiimote rotation).

The pointing functionality, instead, does not require a new interactor style. The pointer, in fact, is only a new 3D object added to the scene. The 3D cursor, when the volume is rotated, rotates together with it. This allows users to point a particular region, fix there the cursor, and then rotate the volume in order to visualize that region from other viewpoints.

Table 1. Wiimote buttons mapping table

	Pointing state	**Manipulation state**	**Cropping state**
A	Fix / Unfix the pointer	Apply CLUT	-
B	Show / Hide the pointer	Rotate the object	Rotate the cropping box
-	Move outward the pointer	Move outward the object	Change the cropping box face
+	Move inward the pointer	Move inward the object	Change the cropping box face
1	Enlarge the pointer	Zoom out	-
2	Reduce the pointer	Zoom in	-
PAD	Move inward / outward the pointer	Translate the object	Translate the box / selected face
HOME	Switch to manipulation state	Switch to cropping state	Switch to pointing state

4.2.1 *Vocal commands*

The speech recognition component has been written in C++ by using the Sphinx-III speech recognition system from Carnegie Mellon University [13]. It has been integrated into the MITO system as for the Wiimote driver. This speech recognition system includes both a decoder and an acoustic trainer.

The Sphinx-III decoder is based on the conventional *Viterbi search* algorithm. It needs a lexical model, an acoustic model and a language model in order to perform recognition. Since we need to recognize only 10 commands (the available CLUTs) we have developed and trained an "ad-hoc" acoustic model. The output of the decoding operation is the best recognition hypothesis.

In order to activate the decoder, a user has to press and keep pressed the Apply CLUT button on the Wiimote (available in the manipulation state). An event is sent to the speech recognition component that activates the decoder thread. A feedback is visualized on the display (so the user knows that he can speak). Once the user releases the Apply CLUT button, another event is sent to the speech component that asks the decoder to stop the vocal command acquisition and to provide its best recognition hypothesis. A different visual feedback informs the user if the command has been recognized or if the recognition has failed.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an intuitive non-obstructive interface for semi-immersive virtual medical environments. It combines the Nintendo Wii controller and speech recognition technology, and has been specifically designed and developed by taking into account skills and needs of the clinicians we have interviewed.

The aim has been to support and enhance the performance of daily clinical tasks, where manipulation and examination of three-dimensional organs, reconstructed from scan data, are scheduled. It is our conviction that medical experts can obtain great advantages by using virtual reality technologies in their work but only if they are provided with a natural and intuitive manner to interact in the 3D space with the structures of interest.

Currently we are carrying out a formal usability evaluation of the proposed ITs involving clinicians and students of medicine. The collected data analysis will be important to validate the proposed interaction model and to better understand the usability issues not yet recognized. We are also planning to enhance both the speech component, by integrating additional voice commands, and the pointing feature, by adding a filter for hand tremors. Finally, in order to provide a very suitable and usable tool for computer-assisted education and training of clinicians, a semi-immersive multi-user interaction approach is under development.

6. REFERENCES

[1] Salgado, R., Mulkens, T., Bellinck, P. and Termote, J. L., "Volume rendering in clinical practice. A pictorial review", JBR-BTR, 86(4): 215-20 (2003).

[2] Robb, R. A. "The Biomedical Imaging Resource at Mayo Clinic", IEEE Trans. Med. Imaging, 20(9): 854-867 (2001).

[3] Wu, Y. and Yencharis, L. "Commercial 3-D Imaging Software Migrates to PC Medical Diagnostics", Advanced Imaging Magazine, 16-21 (1998).

[4] Rosset, A., Spadola, L., Ratib, O. 2004. OsiriX: a new generation of multidimensional DICOM viewer based on new imaging standards. In Proceedings of the 18th International Congress and Exhibition of Computer Assisted Radiology and Surgery (Chicago, USA, June 23 – 26, 2004). CARS 2004.

[5] Narayan, M., Waugh, L., Zhang, X., Bafna, P. and Bowman, D. 2005. Quantifying the benefits of immersion for collaboration in virtual environments. In Proceedings of the ACM symposium on Virtual reality software and technology (Monterey, California, USA, November 07 – 09, 2005). VRST 2005.

[6] Bowman, D. A., Chen, J., Wrave, C. A, Lucas, J., Ray, A., Polys, N. F., Li, Q., Haciahmetoglu, Y., Kim, J. S., Kim, S., Boehringer, R. and Ni, T., "New Directions in 3D User Interfaces", The International Journal of Virtual Reality, 5(2): 3-14 (2006).

[7] Nintendo Wii, http://it.wii.com/

[8] Coronato, A., De Pietro, G., and Marra, I. 2006. An Open-Source Software Architecture for Immersive Medical Imaging. In Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (La Coruna, Spain, July 10 – 12, 2006). VECIMS 2006.

[9] WiiLi, a GNU/Linux port for the Nintendo Wii, http://www.wiili.org

[10] WiiBrew Wiki, http://wiibrew.org/index.php?title=Wiimote

[11] Dech, F. and Silverstein, J. C. 2002. Rigorous Exploration of Medical Data in Collaborative Virtual Reality Applications. In Proceedings of the International Conference on Information Visualisation (London, UK, July 10 – 12, 2002). IV 2002.

[12] Gallo, L., De Pietro, G. and Marra, I. 2008. 3D Interaction with Volumetric Medical Data: experiencing the Wiimote. To appear in Proceedings of the First International Conference on Ambient Media and Systems (Quebec City, Canada, February 4 – 7, 2008). Ambi-sys 2008.

[13] Sphinx-III speech recognition system, http://cmusphinx.sourceforge.net/html/cmusphinx.php

Music Selection using the PartyVote Democratic Jukebox

David Sprague
University of Victoria
3800 Finnerty Road
Victoria, BC, Canada
dsprague@cs.uvic.ca

Fuqu Wu
University of Victoria
3800 Finnerty Road
Victoria, BC, Canada
fuquwu@cs.uvic.ca

Melanie Tory
University of Victoria
3800 Finnerty Road
Victoria, BC, Canada
mtory@cs.uvic.ca

ABSTRACT

PartyVote is a democratic music jukebox designed to give all participants an equal influence on the music played at social gatherings or parties. PartyVote is designed to provide appropriate music in established social groups with minimal user interventions and no pre-existing user profiles. The visualization uses dimensionality reduction to show song similarity and overlays information about how votes affect the music played. Visualizing voting decisions allows users to link music selections with individuals, providing social awareness. Traditional group norms can subsequently be leveraged to maintain fair system use and empower users.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Screen design; H.5.3 [**Group & Organization Interfaces**]: computer-supported cooperative work; H.5.5 [**Sound & Music Computing**]: Systems

General Terms

Performance, Design, Human Factors.

Keywords

CSCW, information visualization, music systems, music map, entertainment, voting, social interaction, group dynamics

1. INTRODUCTION

Conflicts in informal social environments such as house parties often arise from differences in individual preferences. Unlike music sharing, choosing music for a group involves making compromises between each user's individual tastes. Music is frequently chosen by a party's host to avoid conflict. This leaves music selection in the hands of an individual. Although some hosts allow anyone to help determine the music played, the time required to repeatedly select songs or albums is often unappealing. Small groups of self-designated disk jockeys (DJs) can take over music selection responsibilities, also limiting decisions to a select few users. Playing music that everyone knows can frustrate more 'musically adventurous' party-goers. Conversely, playing less popular music can annoy people who want to hear familiar songs.

We present PartyVote, a system that provides established groups with a simple democratic mechanism for selecting and playing music at social events (see Figure 1). Implicit rules or *norms* are more likely to restrict behaviors in small established groups since undesirable actions are evident and peer pressure is effective[4, 8]. For example, actively preventing friends from choosing music is probably contrary to group norms. A visualization system for a casual setting should be intuitive, informative and would ideally support unwritten social rules to ensure fair system use while enabling individuals to express their preferences. Our system accomplishes this by improving *user visibility* - information presented to everyone about an individual's actions. Our design goals for PartyVote include:

1. *Support individuals & appease the group:* The system should act as a 'discount DJ', taking requests and playing music that will appease the most people.

2. *Leverage existing social dynamics:* We believe that system use will be constrained by group norms. The system should allow users to see each other's votes and their influence on the music selection. This allows social pressures to be applied.

3. *Minimize & simplify necessary interactions:* The time commitment for choosing music should be minimal to maximize general appeal. For example, users should not need to choose more than one song. The system should not need to be the center of attention. The visualization should be transparent, fun, and intuitive.

4. *Use common hardware & personal music:* Small casual social gatherings are more likely to have a PC, speakers, and a local digital music library than to have a tabletop display or other specialized equipment.

PartyVote allows each participant at a party to choose a song, album, artist, or genre from a local digital music collection. Each voter is guaranteed that at least one song from their choice will be played. Each song is given a weight dictating the probability that it will be played. User votes increase the weight of similar songs and define the boundaries of the potentially playable song region in the collection's music information space (the black region in Figure 1). Songs

AVI '08, May 28-30, Napoli, Italy

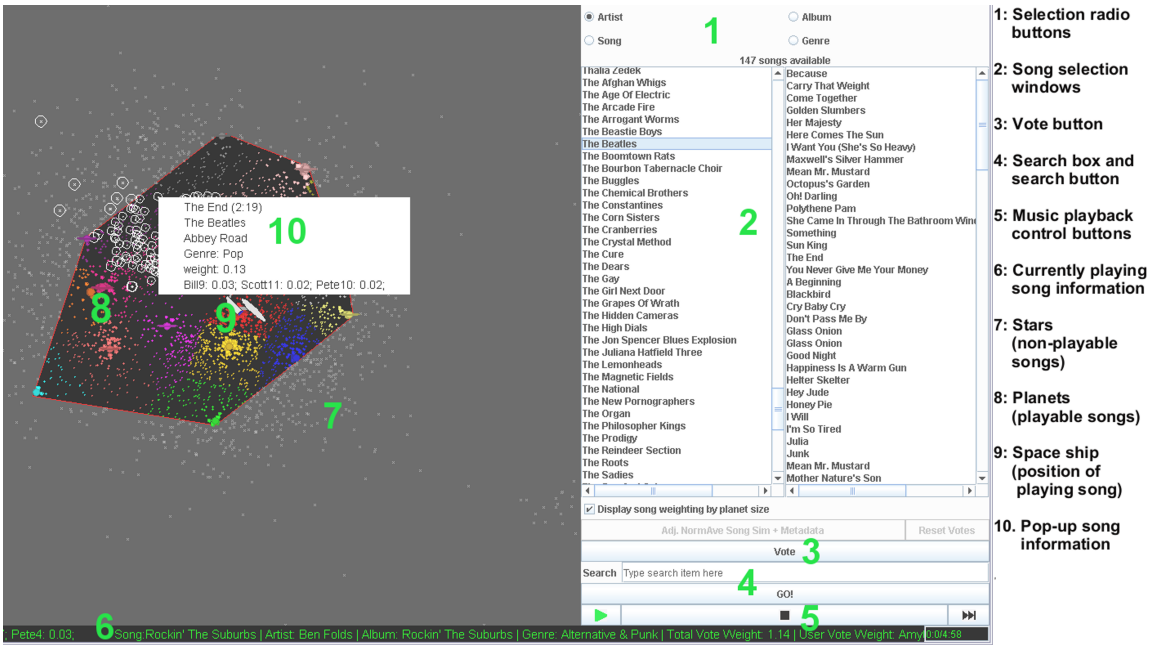


Figure 1: A screen shot of the PartyVote system demonstrating the space-themed visualization window (points 6-10) and the selection window (points 1-5). The visualization matches twenty-two votes cast during a system evaluation with the Beatles selected. System components are identified on the right.

outside of this area are not played. Each voter determines at least one song that is played while the remaining songs are chosen to appease the greatest number of people.

Our research offers two main contributions. First, mediating social conflicts and decision making using a minimal commitment voting mechanism is a novel approach to group music selection with substantial benefits. Second, PartyVote uses visualized social awareness cues, enabling peer pressure to enforce system fairness. No previous group music jukeboxes use either approach.

2. PREVIOUS WORK

PartyVote builds on previous research involving synchronous co-located (co-present) computer assisted collaborations, digital music system design and music recommender systems. Musical preferences can be highly individual, yet party attendees share a common audio signal. Consequently, our review focuses on co-present music sharing.

Previous CSCW research for small casual groups, such as the Notification Collage [6], and work by Morris et al.[11], identify a need for awareness information. These systems rely heavily on group norms to ensure the systems are not abused. Computer supported social mediation systems such as Meme Tags [1] and Ticket2Talk [10] were designed for large parties. Our research similarly looks at social mediation, but we focus on smaller more-integrated groups.

There are three common approaches to computer mediated music selection for synchronous co-located groups: playlist generation systems, recommender systems, and collaborative music jukeboxes. Playlist generation systems create song sequences based on either a user's tastes or musical flow. Systems like Pandora.com [14] use user feedback, similar to votes, to refine and guide this playlist generation. These systems, however, provide no feedback as to how music is selected and are designed for single users.

Recommender systems such as musicFX [9], Adaptive Radio [2] and Flytrap [3], rely on user profiles to find the best compromises for the group. Profiles consist of a person's opinion (or vote) about each item in a set. Acquiring a reasonable number of votes demands a large time cost but does not account for a user's current mood if done in advance. McCarthy and Anagnost [9] reported that some users discovered how their algorithm worked and constrained their interests to force other users to listen to their top preferences. Jameson [7] suggests using transparent decision mechanisms and providing user awareness information to avoid such manipulations. PartyVote addresses both of these suggestions.

Collaborative jukebox systems enable users to jointly select music. Existing systems require frequent user interactions. Jukola [12] is a democratic MP3 jukebox designed for use in public places, such as a coffee shop. Jukola relies on users to nominate and vote for songs. A staff member was reported unfairly using the system by repeatedly skipping songs. We believe this was due in part to the absence of a peer pressure mechanism to mediate conflicts. By contrast, PartyVote's visualization allows users to see each other's influence on the music selection and thereby influence their peer's choices. MUSICtable [15], represents songs using a collaborative music library visualization system, using a static two-dimensional (2D) geographic map metaphor, similar to PartyVote. Unlike PartyVote, MUSICtable requires regular user interactions, and it is designed to be the attentional focus at a party.

3. PARTYVOTE SYSTEM OVERVIEW

PartyVote is designed for use with a regular keyboard, mouse, and monitor. Our system discussion focuses on our three research priorities: the visualization, the voting mech-

anism, and how we expect the design to affect decisions and group dynamics. Thus, system interaction techniques are only discussed as needed.

3.1 System Interface

The PartyVote interface consists of two main sections: a visualization window and a text based selection window (see Figure 1). Clicking an entry in the selection window results in corresponding items in the visualization window being selected and vice versa (i.e. brushing and linking).

3.1.1 Selection Window

The selection window permits 'iTunes-like' text-based music browsing by artist, album, song title, or genre. Selecting an artist, album or genre in the left selection window results in all songs in that sub-list being displayed in the right selection window. Selected items can be voted for using the 'vote' button, which requires the user to input a user name at that time [8]. Conventional pause/play, stop and skip buttons provide music control.

3.1.2 Visualization Window

PartyVote uses people's votes to weight and cull the set of potentially playable songs. This is done by determining a similarity between each pair of songs, and positioning the songs in 2D space, a priori, so that similar songs are close together. A space travel analogy provides a playful, intuitive and informal atmosphere for the system, while still conveying similarity-based point distances. Initially, all songs are represented as *stars*: songs that have no probability of being played. Each star is drawn as two intersecting light gray line segments so it fades toward the gray background. Each star has a fixed 2D position.

Votes define the playable music region, which is shown as a convex polygon. Each song in this region, or *planet*, is provided a weight. Votes increase the weight of nearby songs. Planets are either guaranteed to be played (*guaranteed songs*) or potentially playable (*potential songs*). Guaranteed song planets are distinguished from other planets by an orbiting moon or Saturn-like ring. Originally planet ornamentation was random, but early users believed this conveyed information. A planet's size represents its weight, and its colour matches the vote that influenced its weighting the most. If vote v1 represented by the colour red contributes 0.3 to a song's total weight of 0.4, that planet is coloured red. Thus, users can quickly determine the main influence a vote had and who's vote caused a song to be played. Played songs have a darker planet colour and cannot be replayed.

The currently playing song is identified by a space ship, which flies to the planet when the song begins playing. Song and voting information about the current song is displayed at the bottom of the application. Pop-up song metadata, weight and voting data appear when the mouse rolls over a planet or star. Left clicking a planet or star selects it. Zooming is controlled by a right mouse click.

PartyVote's visualization aims to provide system transparency. Figure 2 demonstrates how the playable songs change following a vote. By visualizing how songs are chosen, users can make informed choices. Indecisive voters can choose to vote for popular music to appease the group. A user who dislikes a band may avoid voting for music in that band's region. Some people may vote for a song simply because they like it, while others may choose a song to increase

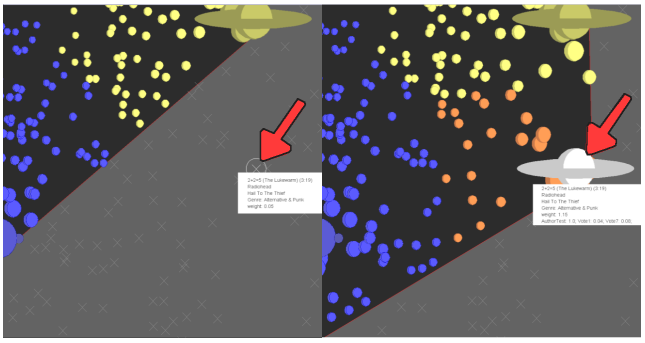


Figure 2: Alterations to the potentially playable song region based on a vote for the song indicated by a red arrow. Songs whose weights are primarily determined by the new vote are orange. The selected song is white.

the probability of a region of songs being played. A planet's weight and the three primary contributing vote weights are provided as song text information. This can help users find people with similar tastes, identify people breaking group norms, and permit strategic voting.

3.2 Algorithms

Multidimensional Scaling: Principal component analysis, self-organizing maps, and multidimensional scaling (MDS) are frequently used to map multidimensional spaces to 2D [13]. We chose to use MDS because it is efficient, tends to provide a globally correct solution, and permits the use of any distance metric. Sound similarity and music metadata were used to calculate song distances. Layout implementation details are discussed later.

Voting Algorithm: Each user vote affects music selection in two ways: at least one song per vote is guaranteed to be played and music similar to the selection is more likely to be played. A vote identifies a collection of one or more songs. The guaranteed song is randomly chosen from this collection. Each vote has a weight of 1.0 and this weight is distributed evenly across all songs in the collection.

The Playable Music Area: A convex hull of the guaranteed song positions is calculated using a 2D Graham's Scan, and defines the potentially playable area of music [5]. A convex hull defines the minimal convex polygon bounding all user preferences. Points within the hull define a compromise between the votes. Numerous music regions could be used, however, we feel the convex hull results in more musical compromises between users and leads to music discoveries. Similarly, a higher dimensional Graham's Scan could be used but calculations would be extremely slow.

For each song voted for in the playable area, similar songs are weighted according to the formula $W_j = \sum_{i=1}^{N}(W_i \div D_{ij})$. D_{ij} is the Euclidean distance between songs i and j, W_i is the weight given to song i and N is the set of songs in the library. Thus, songs frequently voted for and songs near popular music are more likely to be played. We believe this approach will satisfy most listeners while boundary voters get to listen to at least one song of their choice.

Music Playlists: Playlists are generated every time a vote is cast, with guaranteed songs played within two hours of their vote. Thanks to user feedback, future releases will

play guaranteed songs within a half hour. Potential songs fill in the remainder of the playlist based on song weight.

3.3 Implementation

PartyVote was designed for typical hardware and personal digital music libraries. The PartyVote system was written using Java Swing. MP3 files were played using JLayer. Initial user testing was conducted on a 1.66 GHz Intel Centrino Duo Core laptop with Windows XP and 1GB of RAM. The mean refresh rate was 34 frames per second. The primary author's 3364 song personal music collection was used.

3.3.1 Song Layout

Music layouts were generated a priori using custom tools. Song metadata was collected using iTunes and saved in a text file. Metadata was weighted with artist, album, genre, song title and user rating weighted heavily. Each dimension was normalized between 0.0 (a perfect match) and 1.0 (no match). String pair differences were either Boolean or 1−(# of common characters/text length). Numerical fields used absolute difference. Metadata errors were not corrected.

Sound similarity was calculated using bextract and the Marsyas sound analysis library [16]. The first 120 seconds of each song were analyzed using Mel Frequency Cepstral Coefficients and Short Time Fourier Transforms to provide 68 attributes. Dimensions were normalized and Euclidean distances between pairs of song vectors were calculated.

Normalized metadata and sound similarity distances were combined in a variety of ways and visually tested to ensure a good layout. The squared normalized average was ultimately chosen to remove variability differences between sound similarity and metadata distances and to emphasize tightly clustered song groups. Songs were laid out in 2D using MDSteer [17] without steering. Layout files were precomputed and read at PartyVote's startup.

4. USER EVALUATIONS

We have used PartyVote at two parties. Space constraints prevent us from discussing early user testing in detail; however, we noted that PartyVote use varied radically. Some users voted strategically while others simply voted for their favorite songs. Some users chose to vote multiple times because they wanted to use the system more or clarify their preferences. Overall the system was used as expected.

5. CONCLUSIONS AND FUTURE WORK

This study is a first step in investigating how informal collaborations can be facilitated and mediated via user voting. In the near future, we plan to conduct several larger evaluations and examine how other visualizations in the same PartyVote framework effect user behaviors. We also plan to look at how voting can constrain options and searches in other informal applications such as selecting movies to rent.

PartyVote provides a lightweight mechanism for established social groups to choose music at a party, relying on group norms and participant visibility to ensure fair system use. Guaranteed songs and song weightings allow the majority to determine the general style of music played while safeguarding individual choice. Unlike previously reported recommender systems and music jukeboxes, no user profiles are needed and constant user vigilance is not required. Thorough user testing is now required.

6. REFERENCES

[1] R. Borovoy, F. Martin, S. Vemuri, M. Resnick, B. Silverman, and C. Hancock. Meme tags and community mirrors: moving from conferences to collaboration. In *Proc. CSCW '98*, pages 159–168, 1998.

[2] D. L. Chao, J. Balthrop, and S. Forrest. Adaptive radio: achieving consensus using negative preferences. In *Proc. GROUP '05*, pages 120–123, 2005.

[3] A. Crossen, J. Budzik, and K. J. Hammond. Flytrap: intelligent group music recommendation. In *IUI '02 Conference Proceedings*, pages 184–185, 2002.

[4] C. Danis and A. Lee. Evolution of norms in a newly forming group. In *Proc. INTERACT '05*, 2005.

[5] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, Germany, 1997.

[6] S. Greenberg and M. Rounding. The notification collage: posting information to public and personal displays. In *Proc. CHI '01*, pages 514–521, 2001.

[7] A. Jameson and B. Smyth. Recommendation to groups. In *The Adaptive Web*, pages 596–627, 2007.

[8] B. Kules, H. Kang, C. Plaisant, A. Rose, and B. Shneiderman. Immediate usability: a case study of public access design for a community photo library. *Interacting with Computers*, 16(6):1171–1193, 2004.

[9] J. E. McCarthy and T. D. Anagnost. MUSICFX: an arbiter of group preferences for computer supported collaborative workouts. In *Proc. CSCW '98*, pages 363–372, 1998.

[10] J. F. McCarthy, D. W. McDonald, S. Soroczak, D. H. Nguyen, and A. M. Rashid. Augmenting the social space of an academic conference. In *Proc. CSCW '04*, pages 39–48, 2004.

[11] M. R. Morris, A. Cassanego, A. Paepcke, T. Winograd, A. M. Piper, and A. Huang. Mediating group dynamics through tabletop interface design. *IEEE Comput. Graph. Appl.*, 26(5):65–73, 2006.

[12] K. O'Hara, M. Lipson, M. Jansen, A. Unger, H. Jeffries, and P. Macer. Jukola: democratic music choice in a public space. In *Proc. DIS '04*, pages 145–154, 2004.

[13] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proc. MULTIMEDIA '02*, pages 570–579, 2002.

[14] Pandora.com. Radio from the music genome project. `http://www.pandora.com/`.

[15] I. Stavness, J. Gluck, L. Vilhan, and S. Fels. The MUSICtable: a map-based ubiquitous system for social interaction with a digital music collection. In *Proc. ICEC '05*, pages 291–302, 2005.

[16] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. Speech and Audio Processing*, 10(5):293–302, 2002.

[17] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *Proc. INFOVIS '04*, pages 57–64, 2004.

A Haptic Rendering Engine of Web Pages for Blind Users

Nikolaos Kaklanis[1], Juan González Calleros[2], Jean Vanderdonckt[2], Dimitrios Tzovaras[1]
[1]Informatics and Telematics Institute, Centre for Research and Technology Hellas, 1st Km Thermi-Panorama Road, 57001 (PO Box 361), Thermi-Thessaloniki (Greece) - +30 2310 464160 (internal 141, 177)
{nkak,Dimitrios.Tzovaras}@iti.gr
[2]Belgian Laboratory of Computer-Human Interaction (BCHI), Louvain School of Management (IAG), Université catholique de Louvain (UCL). Place des Doyens, 1, B-1348 Louvain-la-Neuve (Belgium) - +32.10/47{8349,8525}
juan.gonzalez@student.uclouvain.be, jean.vanderdonckt@uclouvain.be

ABSTRACT
To overcome the shortcomings posed by audio rendering of web pages for blind users, this paper implements an interaction technique where web pages are parsed so as to automatically generate a virtual reality scene that is augmented with a haptic feedback. All elements of a web page are transformed into a corresponding "hapget" (haptically-enhanced widget), a three dimensional widget exhibiting a behavior that is consistent with their web counterpart and having haptic extension governed by usability guidelines for haptic interaction. A set of implemented hapgets is described and used in some examples. All hapgets introduced an extension to UsiXML, a XML-compliant User Interface Description Language that fosters model-driven engineering of user interfaces. In this way, it possible to render any UsiXML-compatible user interface thanks to the interaction technique described, and not only web pages.

Categories and Subject Descriptors
D.2.2 [**Software Engineering**]: Design Tools and Techniques – *User interfaces*. H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Graphical user interfaces*. I.3.6 [**Computer Graphics**]: Methodology and Techniques – *Interaction techniques*

General Terms
Design, Human Factors, Languages.

Keywords
Haptically enhanced widget, haptic interaction, user interface extensible markup language, virtual reality.

1. INTRODUCTION
Although the visual channel probably is the most predominant modality for human-computer interaction in today's computer-based systems, studying when and where an alternative modality may be used instead is still an open and interesting issue, whether this is for a normal user or a person with disabilities. Even more challenging is when the virtual channel could or should be part while offering a supplementary modality for non-visual interaction.

AVI'08, May 28-30, 2008, Napoli, Italy.

There has been much work to offer an audio rendering of web pages to blind users [1,5,8,9,15]. Even the best audio rendering still suffer from some intrinsic limitations such as: sequential navigation, long processing time, difficult navigation within a long page or across web pages, audio rendering is independent of any widget and only works when HTML is well-formed.

In contrast, haptic interaction displays the abilities to overcome some of these limitations: the user may, in principle, freely navigate within a scene provided that it has been designed to appropriately support haptic interaction (the haptic pointer may asynchronously move from an object to another) no sequence is imposed. Consequently, the time required to switch from one screen object to another object may be reduced at the price of a haptic exploration of the scene. Additionally, via haptic channel the blind users can have a perception of the structure of the virtual environment [12], in our case the 3D corresponding of a web page that is very close to the real one (it cannot be exactly the same because 3D rendering puts some limitations in positioning). This is a very important issue because it is essential not only to give blind people raw information but give them the opportunity to navigate through the internet in a way that makes navigation really interesting.

Haptic rendering must be easy-customizable as the specific needs of each user may differ from one another. The characteristics (shape, effects, surface properties, etc) of each component in a haptic environment have to be changeable. This customization is can be achieved with the use of UsiXML.

2. HAPTIC EXTENSION TO USIXML CUI MODEL
The semantics of UsiXML are defined in a UML class diagram. Each class, attribute or relation of this class diagram is transformed into an XML Schema defining the concrete syntax of the UsiXML language in general. A concrete user interface (CUI) is assumed to be expressed without any reference to any particular computing platform or toolkit of that platform.

The CUI model is composed of concrete interaction objects (CIO) and the relationships mapping them (cuiRelationship). The CIO model corresponds to an entity of the UI that is perceivable by the user (e.g., text, image, animation, sound, vocal output) and/or that allows users to interact graphically or vocally with the UI (e.g., a push button, a list box, a check box, a vocal input, vocal menu). Currently, the cio model considers the vocal and 2D graphical (2DGraphicalCio) modalities of interaction.

A complete description of this model could be found in the documentation of UsiXML (www.usixml.org).

The haptic CUI extension of UsiXML, Figure 1, corresponds to the description of haptic graphical concrete interaction objects (HapticGraphicalCio). The new extension adds not just a new interaction type, the haptic, but also the 3-dimensional (3D) graphical representation.

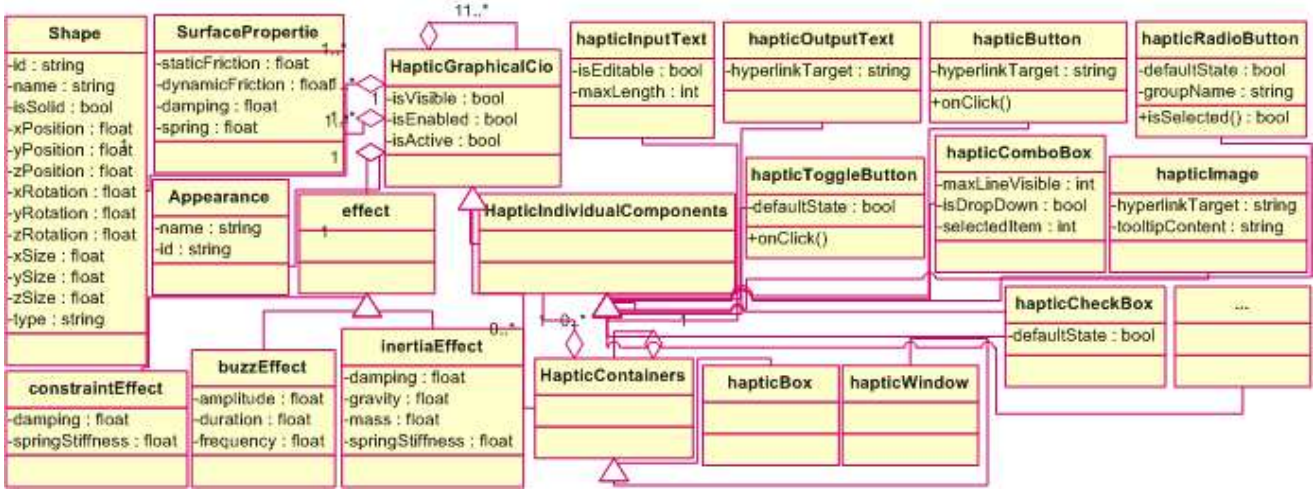


Figure 1 Haptic extension to UsiXML concrete user interface model

The haptic CUI model includes a set of effects:

- buzzEffect. The effect that vibrates the haptic machine.
- constraintEffect. This effect constraint the haptic machine to the point, line or plane using spring damping system.
- inertialEffect. This effect simulates inertial at the end point of the haptic machine as if a mass was a task there, using a spring/damping model.

The second component relevant to the haptic interaction is the *surface properties*. This model corresponds to the properties of the surface of the 3D haptic components. Its attributes are: static friction, dynamic friction, damping, spring. Finally, the shape model associated with the *HapticGraphicalCio* corresponds to the shape (*type*) of a hapget. Each shape is associated with an appearance, the surface, the sensors, for the behavior, all attributes compatible with the abstract definition of X3D language, proposed by web 3D consortium (www.web3d.org).

The HapticGraphical components are divided into HapticContainers and HapticIndividualComponents. For containers, currently, there are two implementations, the hapticWindow and hapticBox. The HapticIndividualComponents are haptic CIOs contained in a haptic container. These CIOs include: hapticTable, haticTree, hapticImage, hapticMenu, hapticMenuItem, hapticSlider, hapticoutputText (a component specialized for output text), hapticInputText, hapticButton, hapticToggleButton, hapticCheckBox, hapticRadioButton, hapticComboBox and hapticItem.

3. IMPLEMENTATION OF THE HAPTIC RENDERING ENGINE

The complexity and the skills required to develop Graphical/Haptic (Multimodal) User Interfaces (GHUI) stress for a toolkit where native GHUI are provided to deploy a GHUI application.

In this section the procedure that was followed for the implementation of the haptic rendering engine is described. The main concept is that user gives a URL as input to the application, then some necessary transformations are executed and finally a 3D scene corresponding to the web page is being created.

First of all, the HTML file is transformed to an XHTML file so as it can be parsed as an XML file. For this transformation an open source tool which is called "Tidy" (http://tidy.sourceforge.net/) is used. After the XHTML file parsing, the corresponding 3D components are presented in the 3D scene. Additionally, a UsiXML file that describes the specific web page is generated. User can save this UsiXML file for further use. When the user loads a previously saved UsiXML file, the 3D scene is updated immediately. This procedure is shown in Figure 2.

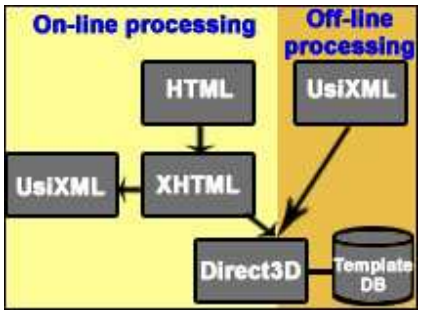


Figure 2 From HTML to a 3D scene

For the creation of the template DB, first off all, "Blender" (open source software for 3D modelling) was used to design a 3D shape for each component. Using "OGRE Meshes Exporter for Blender (exporter to take animated meshes from Blender to Ogre XML), a .mesh file (binary file that represent a 3D model in a form that OGRE rendering engine understands) for each component was exported. Then, the .mesh files were imported in a C++ project using OGRE (Object-Oriented Graphics Rendering Engine) and

finally the 3D UsiXML components were presented in the 3D scene.

The target group of users includes people with normal vision as well as visual-impaired people. Due to this, the representations of the components had to be meaningful for both categories. For instance, an image has no meaning for a blind person but the description of the image (alternate text) has. For this purpose, a speech synthesis engine was integrated to the haptic rendering engine so as to give blind people the opportunity to hear what they cannot read. A speech-recognition engine which offers the opportunity of inserting text without typing was also integrated. Additionally, earcons were used so as each widget can be identified by the unique short sound which is heard when the cursor touches one of the widget's surfaces.

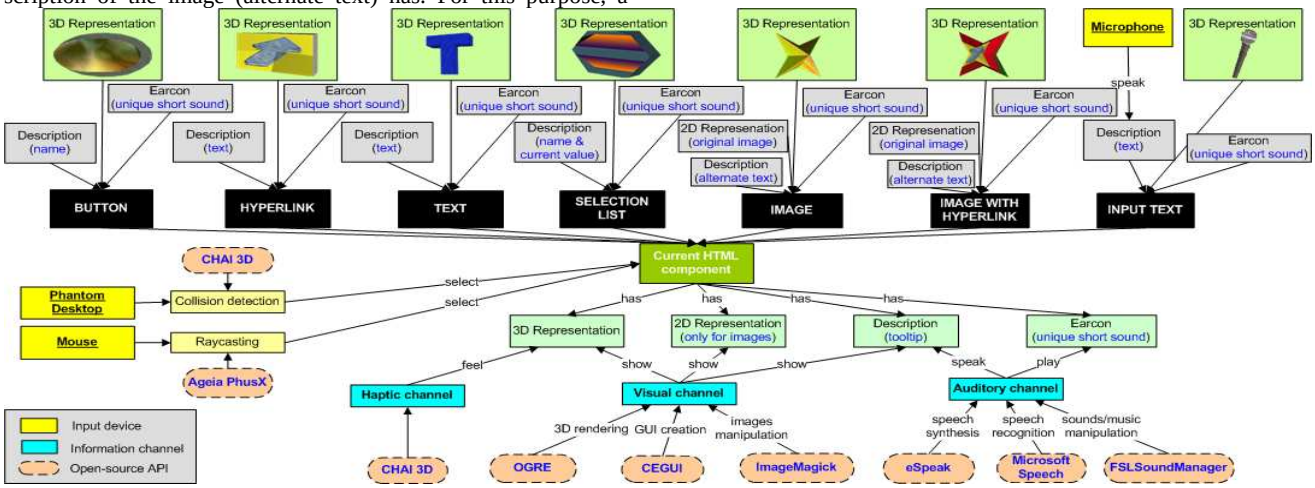


Figure 3 Haptic rendering engine's architecture

The application has mouse support for sighted people and Phantom support for blind users. When the user navigates through the 3D scene using the mouse, the raycasting technique is used for the component's selection while when he/she uses the Phantom the collision detection technique is used (Figure 3). In the 3D scene every HTML component has a 3D representation, a description and an earcon.

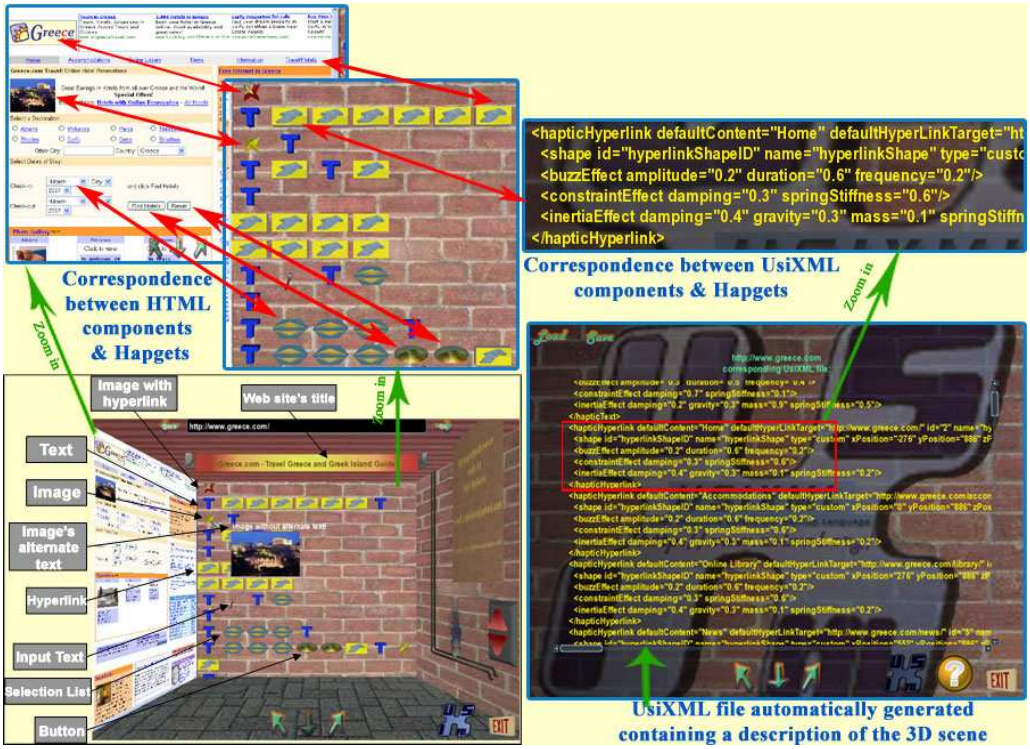


Figure 4 Test case: www.greece.com

For images, with or without hyperlink, there is also a 2D representation, which contains the original image.

When Phantom "touches" an object, the user immediately hears the earcon that corresponds to the objects of this type. If user

presses the LCTRL button of the keyboard while Phantom is in contact with an object, the object's description is being heard via the speech synthesis engine.

Figure 4 presents how the haptic rendering engine works. User starts the speech recognition engine (by pressing the SPACE button of the keyboard) and then gives a URL ("www.greece.com" in this test case) using the microphone. The corresponding to this URL 3D scene is being created immediately.

At the left of the scene, user can see the web page as it is presented in a normal web browser. This side of the scene also interacts as a common web browser. For instance, user can click on a hyperlink and go to another URL with simultaneous update of the 3D objects presented in the scene.

There are some buttons that give user the opportunity to move in the 3D scene and focus on whatever he/she wants into the scene and many visual effects that make navigation through the internet much more impressive than it is via the typical web browsers. For instance, when the cursor goes over a 3D image, the original image shows up and the alternate text of the image is following the cursor as a 3D component's tooltip. However, all these features have to do with the users that have normal vision. The functionality that concerns the visual-impaired users is limited to the haptic and the auditory channel. A blind user can only interact with the 3D components via the haptic device (Phantom Desktop), hear all the necessary information via a speech synthesis engine and pass to the application all the necessary input via a speech recognition engine (using a microphone).

4. CONCLUSION

In this paper we describe a rendering engine to support haptic interaction. We also introduce "hapgets", which are three dimensional haptically-enhanced widgets. The goal of this paper is to describe the rendering engine, so, future work will be dedicated to analyze the graphical representation so as the interaction.

5. ACKNOWLEDGMENTS

We gratefully acknowledge the support of the SIMILAR network of excellence (http://www.similar.cc), the European research task force creating human-machine interfaces similar to human-human communication of the European Sixth Framework Programme (FP6-2002-IST1-507609), the Alban program supported by European Commission and the CONACYT program supported by the Mexican government.

6. REFERENCES

[1] Avanzini, F., Crosato, P. 2006. Haptic-auditory rendering and perception of contact stiffness. In Proc. of 1st Workshop on Haptic and Audio Interaction Design HAID'2006 (Glasgow, August 31st - September 1st, 2006). Lecture Notes in Computer Science, Springer-Verlag, Berlin, 24-31.

[2] Gonzalez Calleros, J.M., Vanderdonckt, J., and Muñoz Arteaga, J. (2006) A Method for Developing 3D User Interfaces of Information Systems. In Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006 (Bucharest, June 6-8, 2006). Springer, Berlin, 85-100.

[3] Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., and Lopez-Jaquero, V. 2004. UsiXML: a Language Supporting Multi-Path Development of User Interfaces. In Proc. of Int. Conf. on Design, Specification, and Verification of Interactive Systems EHCI-DSVIS'2004 (Hamburg, July 11-13, 2004). Lecture Notes in Computer Science, Vol. 3425, Springer-Verlag, Berlin, 207-228.

[4] Mao, X., Hatanaka, Y., Imamiya, A., Kato, Y., and Go, K. Visualizing Computational Wear with Physical Wear. In Proc. of 6th ERCIM Workshop on "User Interfaces for All" (Florence, October 25-26, 2000).

[5] Magnusson, C., Danielsson, H., and Rassmus-Gröhn, K. 2006. In Proc. of 1st Workshop on Haptic and Audio Interaction Design HAID'2006 (Glasgow, August 31st - September 1st, 2006). Lecture Notes in Computer Science, Springer-Verlag, Berlin, 111-120.

[6] Magnusson, C., Tan, Ch., and Yu, W. Haptic access to 3D objects on the web. In Proc. of EuroHaptics'06.

[7] Molina, J.P., Vanderdonckt, J., Montero, F., and Gonzalez, P., Towards Virtualization of User Interfaces based on UsiXML. In Proc. of 10th ACM Int. Conf. on 3D Web Technology Web3D'2005 (Bangor, March 29-April 1, 2005). ACM Press, New York, 169-178.

[8] Ramstein, C. and Century, M. Navigation on the Web using Haptic Feedback. In Proc. of the Int. Symposium on Electronic Art ISEA'96.

[9] C. Ramstein O. Martia, A. Dufresne, M. Carignan, P. Chassé and P. Mabilleau, "Touching and hearing GUIs - Design Issues", *in PC-Access systems. Proceedings of the International conference on assistive technologies ACM/SIGCAPH ASSETS'96*, ACM, 1996, pp. 2-9.

[10] E. Sallnäs, K. Bjerstedt-Blom, F. Winberg and K. Severinson-Eklundh, "Navigation and Control in Haptic Applications Shared by Blind and Sighted Users", *in proc. of 1st workshop on haptic and audio interaction design*, David McGookin & Stephen Brewster (eds.), LNCS, Springer-Verlag, Glasgow, 31st August - 1st September, 2006, pp. 68-80.

[11] V. Tikka, P. Laitinen, "Designing Haptic Feedback for Touch Display: Experimental Study of Perceived Intensity and Integration of Haptic and Audio", *in proc. of 1st workshop on haptic and audio interaction design*, David McGookin & Stephen Brewster (eds.), LNCS, Springer-Verlag, Glasgow, Scotland, 31st August - 1st September 2006, pp. 36-44

[12] D.Tzovaras, G.Nikolakis, G.Fergadis, S.Malasiotis and M.Stavrakis: "Design and Implementation of Haptic Virtual Environments for the Training of Visually Impaired", *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, Vol. 12, No. 2, pp.266-278, June 2004.

[13] J. Vanderdonckt, "A MDA-Compliant Environment for Developing User Interfaces of Information Systems", *in Proc. of 17th Conf. on Advanced Information Systems Engineering CAiSE'05*, O. Pastor & J. Falcão e Cunha (eds.), Lecture Notes in Computer Science 3520, Springer-Verlag, Porto, Portugal, 13-17 June, 2005, pp. 16-31

[14] S. A. Wall and S. A. Brewster, "Assessing Haptic Properties for Data Representation", *to appear in Proceedings of ACM CHI 2003*, ACM, Fort Lauderdale, USA.

[15] W. Yu, D. Reid and S. A. Brewster, "Web-Based Multimodal Graphs for Visually Impaired People", *in Proceedings of the 1st Cambridge Workshop on Universal Access and Assistive Technology (CWUAAT)*, Cambridge, UK, pp.97-108.

Realizing the Hidden – Interactive Visualization and Analysis of Large Volumes of Structured Data

Olaf Noppens
Institute of Artificial Intelligence
Ulm University, Germany
olaf.noppens@uni-ulm.de

Thorsten Liebig
Institute of Artificial Intelligence
Ulm University, Germany
thorsten.liebig@uni-ulm.de

ABSTRACT

An emerging trend in Web computing aims at collecting and integrating distributed data. For instance, various communities recently have build large repositories of structured and interlinked data sets from different Web sources. However, up to date there is virtually no support in navigating, visualising or even analysing structured date sets of this size appropriately. This paper describes novel rendering techniques enabling a new level of visual analytics combined with interactive exploration principles. The underlying visualisation rationale is driven by the principle of providing detail information with respect to qualitative as well as quantitative aspects on user demand while offering an overview at any time. By means of our prototypical implementation and two real-world data sets we show how to answer several data specific tasks by interactive visual exploration.

1. MOTIVATION

The trend of collecting and integrating distributed data into one large repository is gaining more and more momentum. As an example, community efforts such as DBpedia [2] or ReSIST [1] have recently extracted large volumes of structured data from the Web (Wikipedia, US Census Data, DBLP, Citeseer, ACM, etc.). Those repositories are extreme in the sense that they are extraordinary in size and dominated by data sets incorporating only a small and typically lightweight schema.To the best of our knowledge there is no support in navigating, visualising or even analysing large volumes of data in an interactive way appropriately.

As an example, consider a social-network describing members of research communities defined by concepts such as `Project`, `Person`, `Publication`, `Institution` as well as relationships such as `has-Author`, `has-Project-Member`, `has-Research-Interest` etc. A researcher, for instance, can be a working person, an affiliated person, a student or a PhD student (or even an arbitrary combination of these characteristics). An employee can be characterised by his title, his affiliation(s), his degree(s), his project membership(s) in the past and present, his research interests etc. Keeping this in mind, we have to deal with a broad network of people and different kinds of relationships. In this paper we present our approach of combining techniques from visual analytics and interactive exploration of large volumes of heavily interrelated data sets in order to answer data specific tasks. The following describes various selection, exploration and analysis techniques which have been implemented and integrated into our OntoTrack [7] framework. This is done on the basis of two data sets introduced in the next section.

AVI '08, 28-30 May , 2008, Napoli, Italy.

2. DATA SETS

For the rest of the paper, we have chosen two real-world data sets from different domains in order to show how data can easily understood with help of our approach. The first one has been extracted from the MONDIAL Database and the second from the ReSIST Network of Excellence. Both data sets consist of more than hundred thousands entities.

The MONDIAL Database

The Mondial Database[1] (MONDIAL) is a collection of geographic information compiled from different Web data sources such as the World Factbook, Global Statistics and the Terra database [8]. The core of a MONDIAL record consists of data about countries, cities as well as deserts, rivers, or ethnic groups mainly collected from the World Factbook. In addition the collection includes statistical data about populations, area, or length. Entities are typed in a lightweight manner with respect to common geographical concepts such as countries, rivers etc. In addition, various relationships relate entities among each other: for instance, the relationship `has-City` relates countries to cities, `flows-through-country` tells us which countries a river flows through.

ReSIST Network – Resilience Knowledge Base

The Resilience Knowledge Base (RKB) has been created during the first year of the European Network of Excellence in Resilience Computing[2]. The RKB aims at supporting researchers in accessing knowledge on resilience concepts, methods, tools, and the community itself. For that purpose, resilience data has been captured from each partner's information resources such as research interest, details and courseware. This data has been complemented by external sources captured from research information services CORDIS and NSF. Moreover, meta-data about publications

[1] http://www.dbis.informatik.uni-goettingen.de/Mondial/
[2] http://www.resist-noe.org/

and the RISKS index of "Computer-related Risks to the Public" has been gathered from the Citeseer and ACM repositories forming a social-network of researchers and publications. The data is held in a RDFS triple store and accessible via a SPARQL interface.[3] The system incorporates a consistent reference service which maps different URIs from various sources into one reference [5].

3. VISUAL ANALYSIS THROUGH INTERACTIVE EXPLORATION

One lesson learnt from the visual analysis of large data sets in general is that it is not advisable to arbitrarily visualise both all dependencies and all particulars at any time [6]. Therefore, our approach follows the *Visual Information-Seeking Mantra* of "Overview first, zoom and filter, then details-on-demand" [9] by providing detail information only on user demand while offering an overview at the same time.

3.1 Abstraction and Clustering

Following the Information-Seeking Mantra and similar studies, all the connections and relationships between entities can not be visualized and understood at once. We believe that, from the user's point of view, entities with similar characteristics should build obvious or "natural" clusters. For instance, in the MONDIAL domain all European capitals and all countries to which these capitals belong to should automatically be pooled within a cluster as shown in Figure 1. Here, entities are visualised as small filled circles within clusters. Relationships are represented by clubs originating from the set of entities which are considered as the relationship's subject to its objects. However, not only the union of all entities in a cluster can form the origin of a club but also single entities as one can see in Figure 2.

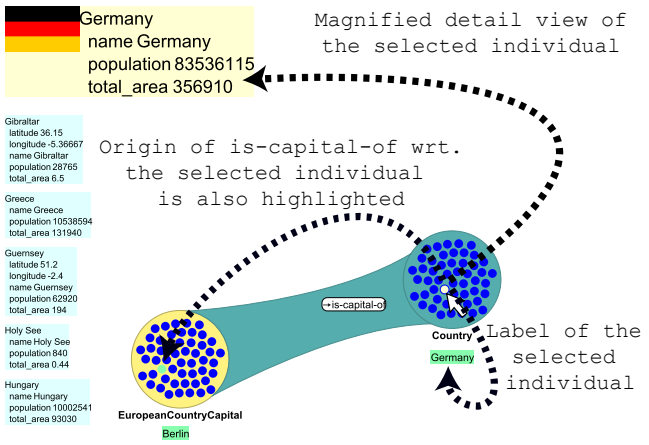


Figure 1: Clustering and club visualisation.

Abstraction also means that entities in a cluster are only drawn if their number is below a user-definable limit. Moreover, the diameter of clusters showing no entities explicitly approximates the number of entities and allows to easily compare the number of entities by the cluster's rendering size. In addition, the number of entities are drawn within a cluster. Additional detail information for each entity such as an image is provided in an optional list as shown on the left hand side of Figure 1. When hovering over an entity with the mouse pointer the list of detail view entries will be scrolled to the corresponding entry and it will be magnified.

To easily grasp all related entities to a focused source entity they are highlighted in all visible clubs when hovering over the source circle. In addition, the labels of these entities are rendered at the bottom of the cluster. For instance, in Figure 1 the mouse pointer is hovering over "Germany". As a result its label is rendered and because "Berlin" is the only origin of the `is-capital-of` relationship the corresponding graphical representation is also highlighted and its label also rendered at the bottom of the `EuropeanCountryCapital` cluster (left hand side of the club).

3.2 Interactive Exploration

When exploring heavily interconnected data sets it is not advisable to show all entities and all their relationships at once. In order to prevent the user in being overwhelmed with currently non-relevant information pieces our user-directed interactive exploration strategy allows for focusing on relevant parts of a data set, or fractions thereof which promise to unveil deeper insights. Initially, one can either start with an user selected entity (e. g. as the result of a query) or with entities showing the same characteristics such as all European capitals in case of the MONDIAL domain. This will result either in showing the graphical representation of that entity or in the case of a set of entities, a slice containing all these entities. As the visualisation and analysis component is integrated into our OntoTrack framework the latter task is carried out by dragging a concept from the schema representation pane on the data analysis pane.

After clicking on the graphical representation of an entity or a cluster a graphical radial preview menu of related entities grouped by their connecting relationships is displayed in an overlay manner. Standard interaction techniques such as mouse-over highlighting and mouse-over zooming as well as intermediate displayed detail information support the user in easily selecting the next club to expand: one or more related clusters are expandable by single mouse-click interaction. For instance, the club shown in Figure 2 represents all project members of "Resilience for Survivability in IST" in the cluster of the right hand side. Here, the preview displays 5 relationships such as `has-author` or `has-affiliation` and one can easily grasp that the entity "Thorsten" for which the menu has been activated is assigned to two affiliations. In order to allow a more flexible exploration of the data set, the exploration direction is not limited to the defined direction of the relationship but also allows to be inversely expanded. The displayed club of Figure 2 represents the relationship `has-project-member` and the preview menu of the selected person also contains the same relationship but in inverse direction allowing a bidirectional exploration of the data set. The direction is denoted by an arrow next to the relationship's label.

Each cluster as well as each (visible) entity can serve as a follow-up point for further expansions. This also allows to branch the expansion by selecting other relationships or de-expand clusters. For instance, in Figure 3 all publications as well as all project memberships (via the inverse expansion of `has-author` resp. `has-project-member` of the members of the Institute of Artificial Intelligence at Ulm University are visible.

To understand how single entities are related to entities in

[3] http://www.rkbexplorer.com/

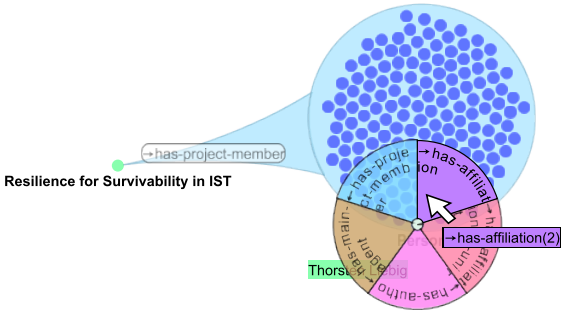


Figure 2: Previews for connected entities grouped by relationships.

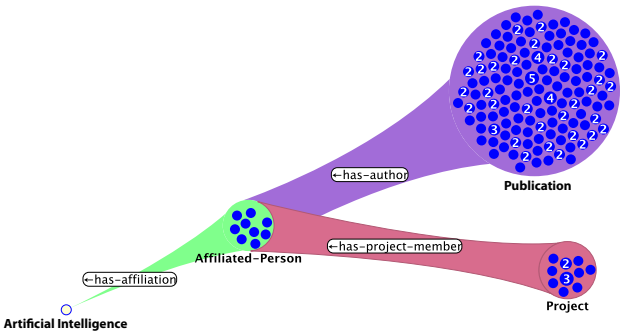


Figure 3: Multiple expansion paths.

preceding or succeeding clusters along the same expansion-path these entities are highlighted as sketched in the following: when hovering over an entity with the mouse pointer all entities in the preceding cluster which are related to that entity will be instantly highlighted. Then, for these entities the procedure repeats recursively. For instance, the leftmost club in Figure 4 shows deserts and the middle one all countries which they are located in. Here, the mouse pointer is hovering over "Algeria" in the second cluster and as a result all deserts which are located there are highlighted in the previous cluster. In addition, their names are displayed on the bottom of the surrounding cluster.

3.3 Analysing Quantities

Besides qualities, the representation of quantities is another important dimension of visual analytics: we found out that quite a number of queries inherently require to take quantities into account. Even if one can easily grasp how many entities are related to a given one with respect to a specific source by manually counting them, it is more complex to visually answer how man entities in a cluster are related to a specific one within its successor cluster. For instance, to answer the question within the MONDIAL domain which is the country in which the highest number of deserts can be found, one would start with the cluster representing all deserts. After expanding the club connected via the `located-in-country` relationship one gets a cluster consisting of all corresponding countries as one can see in Figure 4. Derived from well-known methods from visual analytics we have implemented a simple but powerful solution: the diameters of each country circle scales proportionally with the number of related deserts in the predecessor club. In

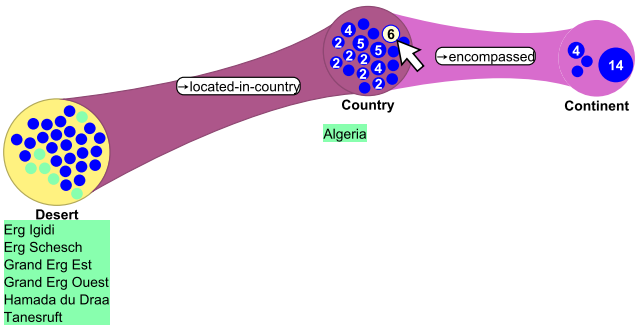


Figure 4: Utilizing quantities to answer questions such as "In which countries can be found the highest number of deserts?".

case that there are more than one single source entity their number is also drawn within the entity circle as shown in Figure 4. Furthermore, one can also see that most deserts can be found on the African continent (when hovering with the mouse pointer over the circle labeled "14" in the `encompassed` cluster the entity's name is shown and the interlinked entities in the preceding cluster are instantly highlighted).

Even if the original question does not refer to quantities, an additional rendering about quantities is a good benefit. Consider a follow-up expansion of Figure 2 (b) to get all co-authors of all publication of "Thorsten Liebig". At a glance one gets the information which is the co-author of most of the publications as shown in Figure 5 (a). The circle labeled "47" is Thorsten himself. Note that the same approach could answer from which affiliation tend to come most of them (Figure 5 (b)).

4. IMPLEMENTATION

Real-world data sets typically consist of thousands of thousand of entities and relationships between them. This makes great demands on the scalability and performance of the implementation of our visualisation approach. We address this with our decision to implement the visualisation and analysing component as a plug-in for our OntoTrack ontology framework: the visualisation is based on the Piccolo framework which can manage huge numbers of graphical objects [3]. The data management is based on a combination of a relational database storage and the wide-spread Java OWL 1.1 API [4] which provides high-level access mechanisms to concepts and relationships.

5. CONCLUSION

In this paper we presented a gainful combination of established methods from visual exploration and visual analytics introducing our new "club visualisation" metaphor. It enables to discover hidden connections between entities while not disturbing the user when exploring large structured data sets. The exploration examples throughout this paper should give an idea how this will help to gain deeper insights into large and heavily interlinked data sets from different domains. The exploration direction as well as the level of detail are determined by the user. In addition to qualities a visual feedback about quantities and the outlining of connected entities enables the user to easily grasp the overall structure as well as on the same time interrelations between

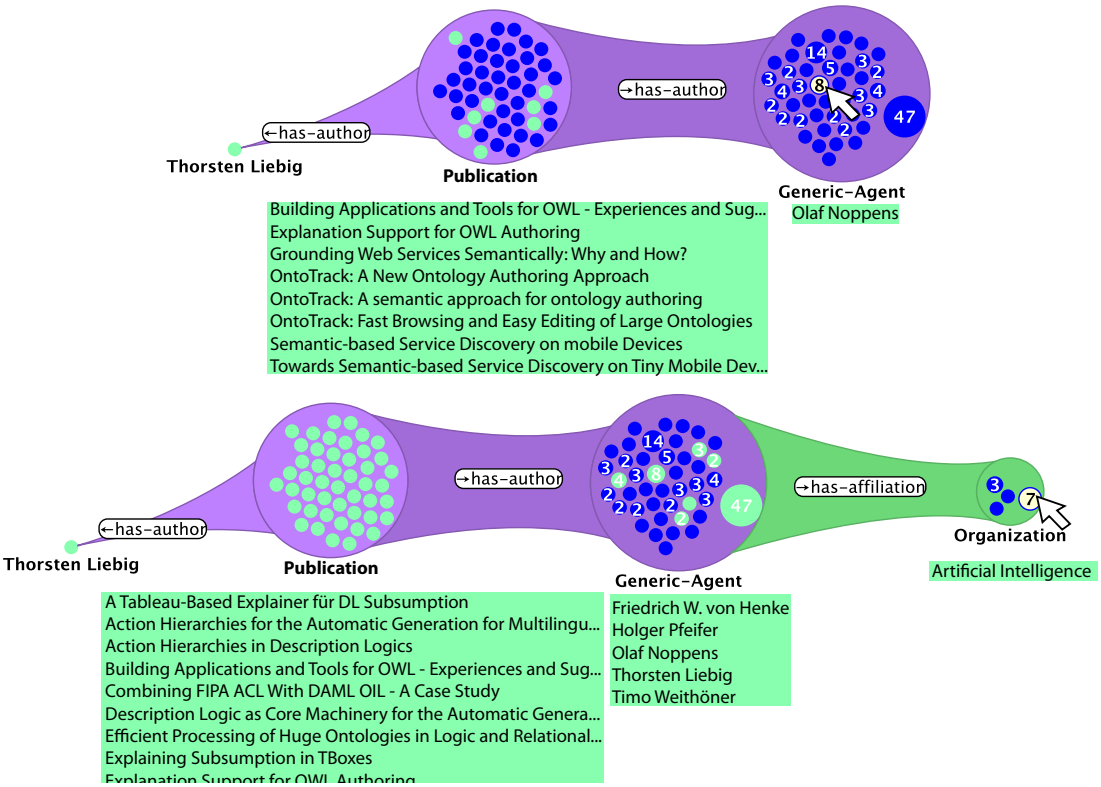


Figure 5: (a) Who is the most co-author of papers involving Thorsten. (b) From which affiliation the most co-authors come from?

specific entities. The interlocking of these techniques adds new exploration and understanding possibilities not found in current tools. All these features have been implemented and integrated into our OntoTrack framework.

Acknowledgments.

This work has been partly supported under the ReSIST Network of Excellence, which is sponsored by the Information Society Technology (IST) priority in the EU Sixth Framework Programme (FP6) under contract number IST 4 026764 NOE.

6. REFERENCES

[1] T. Anderson, Z. Andrews, J. Fitzgerald, B. Randell, H. Glaser, and I. Millard. The ReSIST Resilience Knowledge Base. In *Proc. of the 37th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2007)*, June 2007.

[2] S. Auer, C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proc. of the 6th International Semantic Web Conference (ISWC 2007)*, volume 4805 of *LNCS*, pages 722–735. Springer, 2007.

[3] Ben Bederson, Jesse Grosjean, and Jon Meyer. Toolkit Design for Interactive Structured Graphics. Technical Report CS-TR-4432, University of Maryland, January 2002.

[4] Matthew Horridge, Sean Bechhofer, and Olaf Noppens. Igniting the OWL 1.1 Touch Paper: The OWL API. In *Proc. of the 3rd OWL Experiences and Directions Workshop (OWLED'07) at the ESWC'07*, Innsbruck, Austria, 2007.

[5] Afraz Jaffri, Hugh Glaser, and Ian Millard. URI Identity Management for Semantic Web Data Integration and Linkage. In *Proc. of the 3rd International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2007)*, Vilamoura, Algarve, Portugal, 2007. Springer.

[6] Daniel A. Keim. Visual exploration of large data sets. *Communications of the ACM*, 44(8):38–44, 2001.

[7] Thorsten Liebig and Olaf Noppens. OntoTrack: A semantic approach for ontology authoring. *Journal of Web Semantics*, 3(2):116 – 131, 2005.

[8] Wolfgang May. Information extraction and integration with Florid: The Mondial case study. Technical Report 131, Universität Freiburg, Institut für Informatik, 1999.

[9] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc. of the IEEE Symposium on Visual Languages*, pages 336–343, Washington, USA, 1996.

A Wearable Malossi Alphabet Interface for Deafblind People

Nicholas Caporusso
IMT Alti Studi Lucca
Institutions, Markets, Technology
P.zza S. Ponziano, 6 - 55100 Lucca
Tel: +39 0583 4326561
n.caporusso@imtlucca.it

ABSTRACT
Deafblind people have a severe degree of combined visual and auditory impairment resulting in problems with communication, (access to) information and mobility. Moreover, in order to interact with other people, most of them need the constant presence of a caregiver who plays the role of an interpreter with an external world organized for hearing and sighted people. As a result, they usually live behind an invisible wall of silence, in a unique and inexplicable condition of isolation.

In this paper, we describe DB-HAND, an assistive hardware/software system that supports users to autonomously interact with the environment, to establish social relationships and to gain access to information sources without an assistant. DB-HAND consists of an input/output wearable peripheral (a glove equipped with sensors and actuators) that acts as a natural interface since it enables communication using a language that is easily learned by a deafblind: Malossi method. Interaction with DB-HAND is managed by a software environment, whose purpose is to translate text into sequences of tactile stimuli (and vice-versa), to execute commands and to deliver messages to other users. It also provides multi-modal feedback on several standard output devices to support interaction with the hearing and the sighted people.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces And Presentation**]: User interfaces - *Input devices and strategies*, H.5.2 [**Information Interfaces And Presentation**]: User interfaces - *Haptic I/O*, K.4.2 [**Computers And Society**]: Social Issues – *Assistive technologies for persons with disabilities*.

General Terms
Design, Human Factors.

Keywords
Ubiquitous Computing, multimodal feedback, deafblindness, tactile alphabet.

AVI'08, May 28-30, 2008, Napoli, Italy.

1. INTRODUCTION
According to the Australian Deafblind Council (ADBC), "deafblindness is described as a unique and isolating sensory disability resulting from the combination of both a hearing and vision loss or impairment which significantly affects communication, socialization, mobility and daily living". The deafblind population is small (just over 20,000 in the UK and about 150,000 in the European Community [2]), widely dispersed and also very heterogeneous because it consists of individuals of different ages, whose sensory conditions vary as well as the causes of their disability and the period of life when the phenomenon occurred. Thus, no statistical data about the actual population is available. Although the majority of deafblind people has some residual vision or hearing, their sensory impairment often comes in association with other handicaps such as physical problems, mental retardation or developmental and behavioral disorders.
Managing to acquire a complete independence in communication is the most difficult task for the deafblind community, but it is also the most important: especially congenitally deafblind people have to rely on the presence of a caregiver who advocates for them. They can use various non-verbal methods to relate to others and to overcome the problems of isolation such as behavioral, pictographic and object communication. However, the most effective way for a deafblind person to relate to the others is by using the hands to recognize gestures (e.g. sign language and dactylology, block letters) or facial expressions resulting by the speech act (e.g. Tadoma); tactile alphabets, such as Braille and Malossi, are also very effective [3]. Not surprisingly, each of the above individualized communication systems need an appropriate knowledge of the language. Nonetheless, the shortage of interpreters is a serious barrier to gain access to all levels of communication and interaction.
In conclusion, to break through the isolation of deafblindness, children and adults need communication systems to allow them to have autonomous access to information, to express themselves without the need of an assistant and to gain independence in social participation.

2. STATE OF THE ART
During the past few years deafblind people have benefited from advances in information technology: several new communication solutions have come on the market. Nonetheless, recent research projects also focused on ubiquitous assistive technologies. Most of these devices consist of a sign language sensor and a display.

Unfortunately, sign language and finger-spelling are visual by definition and vision loss can greatly affect the ability of the deafblind to access these forms of communication. The same can be said about speech-recognition systems and text-to-speech engines with respect to the auditory channel. Since little or no attention has been focused on this problem, it is typically not addressed in evaluations: deafblindness is still considered and treated as a single-sense handicap. As a result, many deafblind subjects are incapable of learning the above language systems. Other types of systems, such as the patent CA2179559, [4] rely on gloves equipped with tactile transducers that acquire characters written in sign language. Moreover, a robotic arm [5] aimed to convert text messages into a manual alphabet. Such devices seem to be more expensive than efficient; in fact, their projects are actually discontinued.

As a result, the most effective assistive technology actually available for deafblind people is still based on Braille alphabet: users learn very fast how to type it on dedicated PC keyboards and they read on ad-hoc displays. However, once they leave the site in which the Braille device is installed, they are lost and alone again, since is difficult to realize portable versions of these expensive peripherals.

3. MALOSSI ALPHABET

Deafblind people can communicate using Malossi method, in which the hand (usually the left one) becomes a typewriter for the receiver of the message: words are composed letter by letter by the sender by touching and pinching in sequence different parts of the interlocutor's palm which correspond to the characters. It is very impressive to see the speed with which two deafblind people can communicate using Malossi alphabet. This method is often used by those who had learned to read and write before becoming deafblind, it is also taught to children and it is, in general, an excellent way to relate with people who see and hear normally [1]. Figure 1 shows the distribution of the symbols in Malossi alphabet: the first 15 symbols (from "A" to "O") have to be pressed. The letters are disposed on the phalanxes and on the upper metacarpus (right under the knuckles) from the first to the fifth finger, as a 5x3 matrix. The other letters (15, from "P" to "Z", excluding "W") have to be pinched. The characters are located on two phalanxes (the middle one is left empty), from the first to the fifth finger, as a 5x2 matrix; the remaining character (the letter "W", which is also pinched) is located between the second and the third finger, in the middle of the letters "L" and "M". Thus, 26 symbols can be written. Numbers are expressed as in the Braille alphabet (using the first 10 characters, preceded by the letter "N").

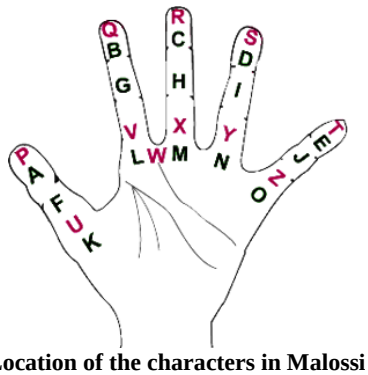


Figure 1. Location of the characters in Malossi alphabet

4. DB-HAND

The main idea at the basis of the interaction design of the system is that it is easier to write a dot on a tiny block-notes held in a hand and to show it to all of our friends than to draw a shape on a big album on a desk and to ask to the members of our family to come and see it. The metaphor fits the problem if we consider that sign languages consist of shapes while Braille displays and keyboards have approximately the size of a big photo book. We have had the opportunity to observe Malossi communication between deafblind people: once they had known the palm of their interlocutor, they could type as fast as standard users do on a keyboard; surprisingly, they could also write on their caregiver's hand and read his answers while walking.

DB-HAND combines both the advantages of Malossi technique as an alphabet-based language and as a ubiquitous tactile communication system and implements them into a natural interface. It is a wearable-hardware/portable-software system that includes a glove equipped with transducers that convert signals from tactile impulses to text messages and vice-versa. Messages are sent and received by the user in Malossi alphabet exactly as it is taught to deafblind people, without any other variation. However, although the output modality remains the same (words are delivered as sequences of tactile impulses to the palm of the left hand of the user) we introduced a modification to the original communication method in order to realize the input interface as a wearable device: when he wants to interact, instead of taking the receiver's hand, the user simply types messages on the same glove he wears on his left hand (he writes on his own hand as it was that of his interlocutor).

Communication in Malossi is turn based: it is bidirectional but it occurs in half-duplex mode; thus, input signals and output stimuli never interfere, because they are never concurrent. Once they are sent by the user, tactile impulses are converted to digital format and they are then interpreted by DB-HAND software application, which distinguishes them as commands or simple text, depending on the content of the message. In the first case, the user's input is recognized as a control signal for an application (e.g. "close the window" or "open a file"); otherwise, it is entered as text in the program having the focus. Feedback is provided by converting the response of an application in Malossi alphabet: messages are delivered to the user's hand as sequences of vibrations in the areas where the letters are located. So, deafblind people are enabled to autonomously operate a personal computer, read and write text documents, surf on Internet pages, chat, send e-mails, participate to forums and, extensively gain access to information and establish social relationships.

We approached to the interaction design of DB-HAND with the intention to realize an interpreter between the deafblind and the hearing and sighted people. So, we implemented multimodal input and output capabilities: hence, deafblind users read and write using tactile impulses while interlocutors view messages using a visual display (or hear them through audio speakers) and reply by typing on a standard keyboard. As a result, deafblind people are enabled to interact with the external world as well as their friends and their family are not required to learn any dedicated language if they want to talk to their beloved (and they do not need the presence of an interpreter). In addition, we hypothesized that DB-HAND might also be a support to learn other communication systems: we applied a Braille label on each to assist the transition between these communication methods.

4.1 Hardware design

DB-HAND interactive glove incorporates couples of transducers (pressure sensors and tactile actuators), which are located on the 16 points defined by Malossi alphabet. Phalanxes that can be pinched as well as pressed contain two symbols and, consequently, two couples of transducers. Once worn, the DB-HAND glove does not prevent the grasping of light objects and tools (e.g. a stick), also without impairing the tactile feeling of their use; furthermore, the user has a free hand (usually the right one) that can perform specific tasks that require a better grip (e.g. holding a Hearing Dog) or a more accurate sense of feedback (i.e. distinguishing different coins).

The peripheral was designed to be modular: it consists of three independent functional units (Physical, Control and Connection layers); in addition, input and output can be joined or split into two separate devices. It is also extensible: additional electronic boards can be plugged directly on the control layer to provide new functionalities to the device (e.g. an LCD display). Also, it is detachable: it can be converted to a stand-alone device (so, it is enabled to work without a PC) by adding a board equipped with an additional control unit, a text-to-speech module and a battery. By doing so, the DB-HAND glove is enabled to convert written messages to speech; this may be useful to interact in contexts where communication is one-directional only (e.g. a centre for deafblind people where an assistant can be invoked in case of need).

The device operates at low-voltage (5 V) and can be powered by its host PC or with a (rechargeable) battery. The actuators ensure low power consumption and high battery duration thanks to their limited current absorption (75-90mA).

Figure 2. An early prototype of DB-HAND

4.1.1 Physical Layer

This layer contains the circuitry and the electronic components (respectively sensors and actuators) that are required to acquire the input (press and pinch actions on the surface of the palm of the hand) and provide the output (vibrations in different areas of the inner part where the fabric is in contact with the skin). We employed low-profile (0,5 x 0,5 x 0,3 cm) tactile switches to acquire impulses. Miniaturized (1 x 1 x 0,3 cm) button-style (shaftless) pager motors were used as transducers for the conversion of electrical signals into tactile stimuli to provide vibrations. The Physical Layer also consists of all the cables that connect the input sensors and output actuators circuitries, which are assembled as two different subsystems, to the Control Layer.

Figure 3. A switch (on the left) and a motor (on the right)

4.1.2 Control Layer

This layer mainly consists of the control unit that manages the device operation. Its main purpose is to decode (respectively encode) input (respectively output) messages in Malossi alphabet: when the user types on the glove, the microcontroller receives sequences of electrical inputs from the sensors located in the physical layer, converts them and sends them as characters to the Connection Layer; when it receives data from the CL, it realizes a letter-to-letter conversion of messages from text to tactile stimuli and fires the actuators in sequence.

4.1.3 Connection Layer

This module consists of the electronic components that allow the device to transfer data and to interact with the computer. DB-HAND is designed to support several types of cable or wireless connection protocols, depending on the connection module: Serial (DB9) and USB options allow the device to be powered by the computer. Wireless solutions, such as Bluetooth and X-Bee (under development) require an additional battery. However, the other layers work regardless of what connection is established, so they are not affected by any change within this module. There is no CL in a detached setup of the DB-HAND hardware.

4.2 Software design

Interaction with the physical interface is managed by the software component, which purpose is to setup the device and to allow the user to control the Operating System and to gain access to applications such as Internet browsers, word processors, instant messaging tools and many other programs. In addition, it parsers the content of the messages which are sent with the device: they may contain a command for of the operating system (i.e. “open a file”) or an input for an application (i.e. for a registration form or a chat) or a sentence that the user wants to communicate (i.e. “I need some water”). Nonetheless, since standard output is not tactile, WIMP interfaces of the Operating Systems and its programs have to be converted into simple text before they can be “visualized” with DB-HAND glove. So, the software architecture was designed to be also an extensible framework which contains the main elements to realize multimodal input and output. In fact, one of the most important elements we had to take into account is that even if DB-HAND allows deafblind people to be autonomous in the interaction, there is usually an assistant with them, especially if they have other disabilities. Therefore, we developed a set of tools dedicated to the interaction with co-located people, who may sit in front of the same computer where the DB-HAND device is connected. The software was coded using Java and C#.

4.2.1 Device Layer

The low-level subsystem of the software component allows the Control Application and other computer programs to directly interact with the glove using a higher-level instruction set. The driver exposes several commands. There are four main directives: writeString, readString, getParameter and setParameter, which are used respectively to fire an actuator, to wait for a sensor to be pressed (or pinched), to get the current status of a parameter or to configure the device by modifying a parameter value. The DB-HAND Device Driver was developed as a portable library and as a stand-alone application for many of the Operating Systems or desktop and mobile computers.

4.2.2 Control Application

The Control Application manages the DB-HAND hardware and software configuration and contains programs developed ad-hoc, such as utilities to support deafblind people in their most common tasks (e.g. an application that converts the messages they type to speech). Not surprisingly, this category of impaired users need a highly customizable interface (more than normal users do): tactile sensitivity varies from one subject to another and it may differ from the upper phalanx of the first finger to the middle phalanx of the fifth; so, the strength and the duration of vibrations have to be calibrated, as well as the speed of each message (or the interval between letters); All these setup operations are realized in the control application and various configurations can be saved for different users.

4.2.3 Communication Framework

The Communication Framework allows several applications to exchange input and output with DB-HAND glove. Once acquired from the device, the input is redirect to the program that has the focus using a Virtual Keyboard: whenever a tactile impulse is detected, the VK emulates the corresponding keystroke event and the character is written. To overcome the lack of support for tactile output in WIMP applications, the Communication Framework also contains a module that interprets Windows menus and controls into a tactile interface. The set of application that can interact with the Communication Framework can be extended with dedicated plug-ins.

5. CONCLUSIONS AND FUTURE WORK

The use of a set of discrete symbols instead of continuous gestures allows a less complex design because sensors and actuators do not require to be read or fired in clusters to acquire an impulse or to produce a stimulus (there is a one-to-one correspondence between characters and sensor-actuator couples). Thus, the device is cheaper. Compared to an average portable Braille display (output only), which price is about 1400$, DB-HAND has a manufacturing cost of 150$ and implements both input and output functionalities.

Tactile switches and coin-style vibrating actuators are really low profile and compact size transducers, which can be embedded within a thin (less than 1 cm) layer of wired fabric. As a result, the device does not have the bouffant aspect of a pugilism glove: it is flexible, easy to wear and also comfortable, it. An advantage with respect to text-to-speech systems is that DB-HAND is silent: stimuli are provided as small vibrations so, even if actuators emit a soft drowning noise when they utter, their sound form has a fast decay, thus it is perceivable only within a very short range (about 0,5 mt). Furthermore, the peripheral grants a high-level of privacy to the user: unlike visual or auditory systems, whenever a message is delivered to the device, it is received only by the one who is wearing the glove.

Although various mock-ups of the device were successfully tested with normal and deafblind people, an experimental study, which aim is to verify the effectiveness of the system in real-life situations, is actually in progress. Details and results of the experiment will be provided.

Regarding the software, we found that it can be improved by adding routines that enable the system configuration to be adaptive: parameters should auto-adjust according to the evolution of the user. In fact, a lot of effort during the evaluation of DB-HAND was spent in a constant calibration work because subjects adapted to the device so fast that most of the time was absorbed by tuning the device configuration.

6. ACKNOWLEDGMENTS

We express our gratitude to Professor Fiorella De Rosis (Department of Informatics of University of Bari – Italy) for sustaining this research project with kindness and patience. We are also thankful to the association “Lega del Filo d’Oro” for providing essential information about the deafblind community and real examples of communication with Malossi alphabet which helped to design the system and to improve its implementation. Acknowledgements go to non-profit institutes and organizations, which aid is fundamental for deafblind people, for providing detailed documentation about the state of the art in the field.

7. REFERENCES

[1] Lega del Filo d’Oro. http://www.legadelfilodoro.it.

[2] National Institute for Mental Health in England, Department of Health – UK Government. "Mental Health and Deafness -Towards Equity and Access. Department of Health Publications, 2005. http://www.wirralpct.nhs.uk/document_uploads/Goverment_Publications/mhdeafguid.pdf.

[3] Department of Health – UK Government. “A Sign of the Times - Modernising Mental Health Services for people who are Deaf. Department of Health Publications, 2005. http://www.dh.gov.uk/en/Consultations/Closedconsultations/DH_4016951.

[4] K.R. Henry, F. Richard. “Tactile trasducers”. GB patent registry n° CA2179559.

[5] D.B. Gilden, B. Smallridge. “Touching Reality: A Robotic Finger-spelling Hand for Deaf-Blind Persons”, Proceedings of “Virtual Virtual Reality for handicapped people” Conference, 1993.

SyncDecor: Communication Appliances for Virtual Cohabitation

Hitomi Tsujita
Department of Computer Science, Graduate School of Humanities and Sciences, Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan
g0220529@edu.is.ocha.ac.jp

Koji Tsukada
National Institute of Advanced Industrial Science and Technology
1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan
tsuka@acm.org

Itiro Siio
Department of Computer Science, Graduate School of Humanities and Sciences, Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan
siio@acm.org

ABSTRACT
Despite the fact that various means of communication such as mobile phones, instant messenger and e-mail are now widespread; many romantic couples separated by long distances worry about the health of their relationships. Likewise, these couples have a greater desire to feel a sense of connection and synchronicity with their partners than traditional inter-family bonds. In many prior research projects, unique devices were developed that required a level of interpretation which did not directly affect one's daily routine - and therefore were more casual in nature. However, this paper concentrates on the use of common, day-to-day items and modifying them to communicate everyday actions while maintaining a sustained and natural usage pattern for strongly paired romantic couples. For this purpose, we propose the "SyncDecor" system, which pairs traditional appliances and allow them to remotely synchronize and provide awareness or cognizance about their partners - thereby creating a virtual "living together" feeling. We present evidence, from a 3-month long field study, where traditional appliances provided a significantly more natural, varied and sustained usage patterns which ultimately enhanced communications between the couples.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *input devices and strategies, user-centered design, prototyping.*

General Terms
Design, Human Factors

Keywords
Awareness, Communication, Synchronization

1. INTRODUCTION
Although various means of inexpensive communication such as mobile phones, video phones, instant messenger (chat) systems, and e-mail are available, many romantically involved couples, separated by long distances, don't feel they adequately "keep in touch".

In sociology there is a principle called "Bossard's Law" - we tend to marry (or date) someone who lives or works 20 miles from where we live or work. This means that a long-distance romantic relationship is hard by its very nature. In this paper, we define "long-distance" as the minimum separation distance required to cause difficulties within a romantic relationship which would not occur if both couples could meet on a regular, frequent and as needed basis.

In the study area of remote communication, this matter is widely recognized. There have been a number of papers discussing the enhancement of awareness between persons separated by great distances. However, these systems reported differences in expectation and therefore emotional gain depended on the family member involved. For example, Peek-A-Drawer [1] focused on supporting communication between a grandparent and grandchild. It described that the frequency of usage and the acceptance of the system where the grandparent actively used the system but the grandchild did not. However, to the question "Did you feel closer to the other person because of the system?" would elicit two different responses depending on the person. For example, from the parents of a married son, they would reply positively. Conversely, the daughter-in-law, asked the same question, had a distinctively different negative response.

Compared to family members living apart, we believe that romantically involved couples, separated by long distances, have a very similar strong motivation to communicate and bond. Our paper talks about the unique situation where romantically involved couples want more interactive, impactful yet natural mechanism which enables a more connected communication environment and hence warmer relationship.

Continuing from our previous research [2], we investigate the system more thoroughly including after the participants "graduated" from the fun/novel stage. Additionally, two additional devices are introduced and their detailed usage results as well as various unexpected/serendipitous uses beyond its traditional/normal application were reported. Finally, this field test also collected numerous detailed system logs as well as participant journals which were analyzed for the findings during a longer three month period.

2. SyncDecor
The basic concept of the SyncDecor system involves the synchronization of pairs of daily appliances such as lights, trash

boxes, and TVs - that are located at a distance from each other to create a virtual "togetherness" experience. For example, when a person turns on his/her light, the light of his/her partner also gets turned on at the same brightness or when a person throws away garbage, the lid on his/her partners trashcan would also move. If this couple were living together, these actions would happen naturally on a daily basis. Therefore, to simulate this experience, this system eliminates the need to engage in special actions such as sending an e-mail and therefore leads to a natural and sustained use. However, since this system is linked to one's daily routine, at times, it may be perceived as intrusive. Furthermore, it may lead to instances where one may curtail the use of a particular device, such as the lamp or TV, based on concern for the other partner. Nevertheless, even with these hurdles, couples who yearn for a richer, more connected and stronger relationship will overcome these hurdles to enjoy better communication by augmenting traditional means such as cell phones and e-mail. In effect, just like a relationship where the couples live together, this system creates an environment where the relationship grows stronger through the concern for one another.

We developed four prototype systems based on what most people interacted on a daily basis: SyncLamp, SyncTrash, SyncAroma and SyncTV.

2.1 SyncLamp

A light source, such as a lamp, is an appliance that is an essential part of our daily life and reflects our activities. Light can also reflect our "presence", "state" and even feelings. Using SyncLamp, when a person controls the brightness of his/her lamp, his/her partner's lamp also changes to the same brightness.

2.2 SyncTrash

The disposal of trash can also reflect not only the "presence" of an individual, but our "activity" in the form of starting/finishing actions (e.g., eating). SyncTrash is a system for sharing the states (i.e. open and close) of the lids of trash boxes. When a person opens the lid of his/her trash box, the lid of the other distant trash box opens.

2.3 SyncAroma

SyncAroma is a system for synchronizing smells between couples and to transmit his/her partner's "feeling" and "state" through an alternative, non-visual medium.

2.4 SyncTV

SyncTV is a system for sharing a common TV channel where a person selects a channel to watch, the TV channel of the other person will also change to the same channel. From there, they will have common topics that may initiate other means of communication such as e-mailing or telephoning.

3. System Architecture

The system architecture of the SyncDecor is described in Figure 1. In this example, House A and House B each have a PC with a SyncDecor system attached. Each PC includes middleware software running on Ruby which controls the X10, Phidgets and IR servers. These two remote PC's are connected over the internet via a central web based server which handles connection management, filtering and logging.

With the SyncLamp and SyncAroma device, an X10 controller is used. The SyncTrash system consists of a pair of trash boxes with servo motors and foot switches. The servo motor is equipped on the side of a trash box for opening/closing the lid and is connected to a computer with a Phidgets Servo device. The foot switch connects to the computer via a Phidgets Interface Kit. The SyncTV system utilizes a USB based PC IR transceiver. The X10, Phidgets and IR transciever all have accompanying server components.

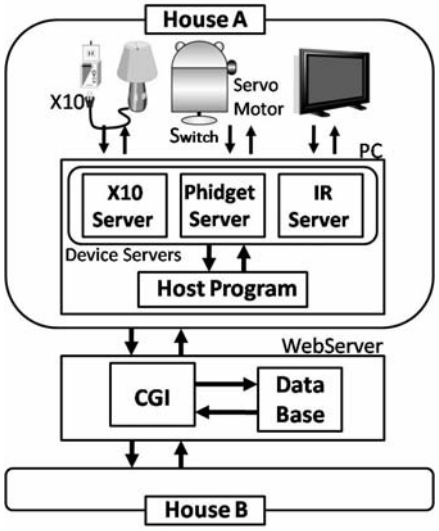


Figure 1. Overview of the SyncDecor architecture.

4. Field Test

In the previous paper, we carried out a field test with the SyncLamp and SyncTrash devices over a period of seven months. The participants were a male (29-year-old office worker) and a female (24-year-old graduate student) living in different cities. The distance between the cities was about 600 km. They had been living apart for three years. We installed a pair of SyncLamp and SyncTrash devices in their rooms (see Figure 2).

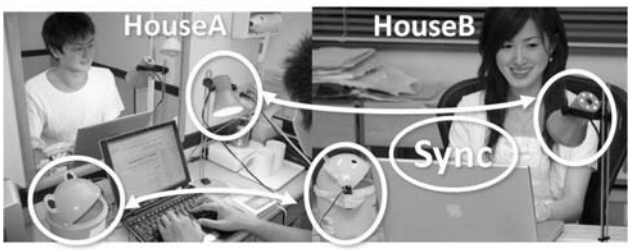


Figure 2. Field test image.

Based on the feedback from the initial field test, we then carried out this field test over a period of three months with the two additional couples (for a total of three couples – six participants).

In this section, first we describe the aim of the field test and approach of the SyncDecor system. Afterwards, we describe the results and discussion.

4.1 The aim of field test

The aim is to reveal the following.

Did the current SyncDecor devices support enhanced communication?

Did the effects and feeling of SyncDecor depend on the type of SyncDecor device?

What other kinds of SyncDecor device is better suited for supporting enhanced communication?

4.2 How the field tests were conducted

We first surveyed the participants using a questionnaire before installing the SyncDecor devices. Basic information such as age, occupation, daily schedule and type of relationship were collected. In addition, we also asked about their daily communication habits.

Next, we installed the SyncDecor devices in their rooms and asked the participants to keep a daily journal to provide feedback on the SyncDecor devices. Separately, we recorded detailed system logs of the field test. In this test, we used the SyncLamp, SyncTrash and SyncAroma devices.

The first relationship was the same from the initial field test. They had almost the same living cycle and habit. The main means of communication were via mobile phone (once or twice per day) and e-mail (once or twice per day).

The second relationship was a male (24-year-old graduate student) and a female (24-year-old graduate student) living in different cities about 1800 km apart. They had been living apart for three years. They had roughly the same living cycle and habit. The main means of communication were via mobile phone (once per day) and e-mail (several times per day). Overall, they kept in frequent contact with each other using these methods.

The third relationship was a male (25-year-old office worker) and a female (24-year-old graduate student) living in different cities about 570 km apart. They had been living apart for two years. They had different living cycle and habit. The main means of communication were via e-mail (once or twice per day). They were not in frequent contact with each other.

4.3 Observation

The results of field test revealed the following.

All participants actively used the SyncTrash device for casual communication. For example, they would open and close the trash box repeatedly to attract their partner's attention. Using the SyncDecor system, the couples felt a certain “warmth” which then often triggered the participants to initiate other means of communication such as e-mailing or telephoning. Often times, they used SyncDecor as a "Good Morning" greeting and woke their partner up by opening and closing the trash box repeatedly. Some sample journal entries included comments such as: "*When I called him, he was sleeping. So I opened and closed the trash box to try and wake him up, but he didn't wake up. At this point, I felt a little angry.*" "*I opened and closed the trash box and tried to wake her up. When she woke up, I was happy to get her attention.*" "*I woke up because he opened and closed the trash box. Honesty, I was a little perturbed.*" During the initial experimental period, the participants regarded the SyncDecor system as novelty devices for explicit communication. However, after the early stages of field tests, they regarded them as a daily appliance with implicit communication capability.

Since the SyncDecor system used familiar, everyday objects, the participant's family also took part in communications. In one journal entry, "*He got home early and turned on the lamp. I was not at home but my family noticed that the light was turned on and sent me a message stating his early arrival.*" In this case, he (the partner) didn't send a message to her about his early arrival. However, the family sending her a message about his early arrival added a different level of closeness to the relationship. While this was a rare use case, the subject's family became part of the relationship – which, in the past, was typically the case before the advent of modern communications technology.

Each member of the couple could feel the daily activities (i.e. disposing of the trash, turning off a lamp, going to sleep) of the other. Moreover, we had many instances where couples guessed their partner's state by the movements of the SyncDecor system as in this quote: "*I felt a bit of hesitation about using the trash box because I came home late. However, after I used it, I didn't receive feedback from her and assumed she must be sleeping*". The effects of SyncDecor system also depended on the participant’s lifestyle. In this sampling, the male participant usually lived alone in a small apartment; the female lived with her family in a large home. As a result, the male participants were more sensitive to the movements or activities of a SyncDecor device. In the case where SyncTrash was installed in the female's room shared with her sister, her partner was more concerned about her sister than her. As a result, the male participant avoided unnecessary opening and closing of the trash box. In the example where a male participant’s parent stayed at his home from an extended period of time (one month), the female participant, who normally used the SyncDecor system quite frequently, noted her curtailed use of the system in deference to the parent’s visit.

Usage of the SyncDecor system also depends on the participants’ living schedules. For example, a couple having different schedules didn't get many chances to show each other's “live” activities such as seeing their lamps and trash box change state. As a result, they developed an alternate use for the SyncDecor system. For example, the male participant turned the lamp on when he left for work. When the female's participant woke up and saw the lamp, this extra action or “thought” itself made the female participant happy. Afterwards, when she left home and turned the lamp off, he also felt certain "warmth" when returning knowing she instinctively took that “extra” step for each other.

The mood and demeanor of the person affect how the SyncDecor system is used. For example, during the field test, all three couples experienced some level of quarrel. All three male participants tried to mend the relationship by using the SyncDecor system. Eventually, it was determined that the SyncDecor system was effective in minor tussles, but wasn’t effective (even counterproductive) during serious fights. This was noted in the following journal entry: *"I was still in bad a mood, but when he tried to improve the situation using the SyncDecor system, it went from bad to worse."*

Finally, the frequency of use depended on each devices. The SyncTrash was used most frequently for explicit communication because the lid of trash boxes changes more dynamically and is transient in nature. On the other hand, participants didn't use the SyncAroma device as frequently because they didn’t have the habit of using an aroma pot or the habit of initiating smell.

Figure 3. SyncLamp, SyncTrash and SyncAroma installed in a participants' room.

4.4 Discussion

First, we answer, did the SyncDecor system have any effect on the romantically involved participants by enhancing communications? Based on post research survey of the participants measuring traditional communications (i.e. number of phone calls and e-mails) before and after the SyncDecor system was installed, four participants said there was no significant change. Two participants specifically mentioned that their initial communications increased mainly to confirm the proper functionality of the SyncDecor device. However, based on the questionnaire regarding whether or not they thought more about the other person, five participants said that their feelings for the other had increased. Within this group of five, several mentioned that they became more cognizant of the others and started wondering what the other was doing - including even hesitating to use certain SyncDecor devices so as not to bother the other person. The remaining one participant mentioned that they thought less of the other. However, this was actually the result of the SyncDecor system providing feedback letting the person know when the other was at home or not – leading to reassurances about the persons wellbeing and hence less worry and therefore further thought.

Based on the results, we feel that the communications between distant couples were enhanced through the user of the SyncDecor system.

Next, we will discuss the difference in how the various SyncDecor devices were used and felt. In figure 4, we show results from the log data obtained from the server during a 3-month span. The three lines in the graph describe the total SyncTrash, SyncLamp and SyncAroma usage/request from the six participants.

Compared to the other device, we found that the SyncTrash device was used the most. We believe that this is because the device was actually used on a daily basis for disposing of garbage. In the diaries of the participants, it was even noted that the SyncTrash device was used explicitly for initiating other forms of communication. However, after the initial novelty of the device wore off, the usage leveled off to a more natural day-to-day usage pattern. This was within our expectation, proving how a natural device allows for natural usage and doesn't let it be forgotten or fade completely from usage once the novelty wears off.

Based on the survey, four of the participants felt that the SyncTrash device was the most useful. The other two devices (SyncLamp and SyncAroma) usage was lower than the SyncTrash device. In this experiment, the SyncLamp device was desk lamp provided to the participants. We later found that certain participants were not in the habit or too busy to use a desk and therefore a desk lamp. Based on this observation, it clearly shows that something which is not a day-to-day object for the participants leads to lower overall usage rates. If however, the light source was something more day-to-day (i.e. room light), the results were most likely different. Nevertheless, two other participants felt that the SyncLamp device was the most useful communication tool. The reason behind this was explained as the SyncLamp device not being transient in nature (i.e. either stays on or off) compared with the SyncTrash device.

The SyncAroma device, four participants mentioned in their survey that they used the device based on its novelty, but that their interest quickly waned due it not being a normal day-to-day action.

Based on this observation, it can be concluded that the participants had a higher usage rate of a device if it was a normal day-to-day object which did not require proactive effort above-and-beyond natural usage patterns.

Finally, based on the participants survey, we would like to discuss what other devices would be better suited for this type of remote communication. Based on the participant's responses, ideas included a warmth synchronized bed, synchronized open/closing curtains and synchronized audio/TV. The last idea, synchronized audio/TV was most popular with three nominations. The explanation behind this included wanting to "fight" over TV channels and/or have a common discussion topic to alleviate loneliness. Furthermore, since the SyncDecor system recorded all synchronization transactions, several participants who had different living schedules wanted the ability to view past activity logs. Therefore, when both parties were on different schedules, there was an increased desire for seeing what actions had taken place. With the SyncDecor system, the participants had an expectation of togetherness. Therefore, perhaps making the log data available can help alleviate that.

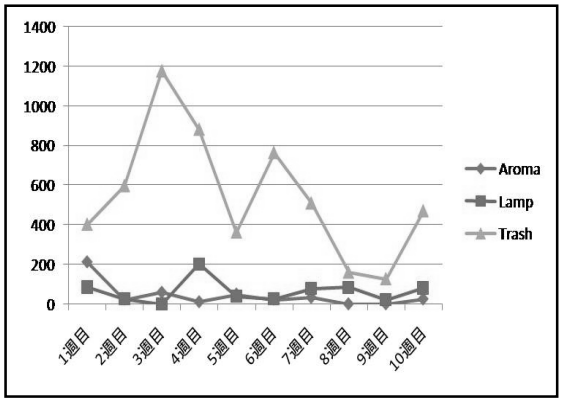


Figure 4. Usage graph of the various devices over a 10-week period.

5. References and Citations

Many research projects have explored the issue of remote awareness. Digital Family Portrait [3] is one of several electronic picture frames that can display the daily activities of family members who live far from their families. For example, it could be used to display the daily activities of an elderly person who lives far from his family. Feather, Scent, and Shaker [4] are elegant design based systems that enable long-distance couples to communicate. MeetingPot [1] is a device that can inform people

of a coffee break, in a common office area, by using the aroma of coffee. Physical awareness proxies [5] convey a remote user's (mainly co-workers or laboratory members) availability, using a tangible interface. Tangible Bits [6] enables users to be aware of background bits at the periphery of human perception using ambient display media such as light, sound, airflow, and water movement in an augmented space. Building Flexible Displays for Awareness and Interaction [7] described a set of flexible ambient devices that can be connected to any available information source and that provide a simple means for people to move from awareness into interaction. In these examples, the devices were designed for asymmetric, one-way communication, which separate the user sensing portion from the information presenting function, thus having no immediate or natural relationship between the user's action and the corresponding remote display. These devices are more passive in nature and only enhance awareness of weaker feeling and ties. We propose devices for symmetric, bi-directional (two-way) communication that combine both the sensing of user action or situation with a correspondingly similar information presentation. In doing so, we support and motivate communications between romantically involved couples, separated by long distances.

LumiTouch [8] is a pair of photo frames, and ComSlipper [9] is a pair of slippers to indicate the activities of a partner who lives far away. ComTouch [10] converts a pressing force to the vibration of the corresponding ComTouch device. Lover's Cup [11] is a communication tool for drinking-together interaction between long-distance couples. The bed [12] is a bed environment that creates the virtual existence of a person (who lives far away) in a bed. inTouch [13] is a pair of communication devices with cylindrical rollers that rotate synchronously. These investigations were optimized more towards communication mechanisms that are more "passive" or casual in nature. Our paper talks about situations where romantically involved couples want more interactive, impactful yet natural mechanisms which enable a more connected/realistic communication environment and hence warmer relationship. Moreover, SyncDecor tries to reflect a person's actions directly onto the remote devices. Our design is based on the synchronization of familiar, everyday objects, without modifying their original function. SyncDecor system can create a virtual "togetherness" experience.

6. Conclusion

We have described the SyncDecor system, which pair remotely installed appliances and electronics so they may synchronize with each other. The objective of this is to create a virtual "togetherness" that enables the couple to share their daily activities with ease through subtle awareness of each other's actions. We built four prototype systems - SyncLamp, SyncTrash, SyncAroma and SyncTV and had three, long distance, romantically involved couples using these devices in a normal, day-to-day setting collecting numerous logs and usage diaries. Based on this usage, we determined the unique ways "feelings" were conveyed through the SyncDecor system as well as the different ways the various devices were utilized within them.

Finally, since we presented a system that leveraged familiar commonplace items, it did not require any extra training or interpretation to use. This allowed for participation beyond the principal romantic parties involved and created instances of spontaneous interaction (and provided additional findings) from other individuals (i.e. family members).

7. ACKNOWLEDGMENTS

This work was funded in part by the Information-Technology Promotion Agency, Japan (IPA).

8. REFERENCES

[1] Siio, I., Rowan, J., Mima, N. and Mynatt, E. Digital Decor: Augmented Everyday Things. In Graphics Interface 2003, PP. 155-166, June 11-13 2003.

[2] Tsujita, H., Siio, I., and Tsukada, K. 2007. SyncDecor: appliances for sharing mutual awareness between lovers separated by distance. Ext. Abstracts CHI 2007, ACM Press (2007), 2699-2704.

[3] Rowan, J. and Mynatt. E.D. Digital Family Portrait Field trial: Support for aging in place. Ext. Abstracts CHI 2005, ACM Press (2005), 521-530.

[4] Strong, R. and Gaver, W. Feather, Scent, and Shaker: Supporting simple intimacy. In Proc. of CSCW 1996, ACM Press (1996), 29-30.

[5] Kuzuoka, H. and Greenberg, S. Mediating awareness and communication through digital but physical surrogates. Ext. Abstracts CHI 1999, ACM Press (1999), 11-12.

[6] Ishii, H. and Ullmer, B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In Proc. CHI 1997. ACM Press (1997), 234-241.

[7] Elliot, K. and Greenberg, S. Building Flexible Displays for Awareness and Interaction. Video Proceedings and Proceedings Supplement of the UBICOMP 2004.

[8] Chang, A., Resner, B., Loerner, B., Wang, X. and Ishii, J. LumiTouch: An emotional communication device. Ext. Abstracts CHI 2001, ACM Press (2001), 313-314.

[9] Chen, C.-Y., Forlizzi, J. and Jennings. P. ComSlipper: An expressive design to support awareness and availability. Ext. Abstracts CHI 2006, ACM Press (2006), 369-380.

[10] Chang, A., O'Modhrain, M.S., Jacob, R.J.K, Gunther, E. and Ishii, H. ComTouch: Design of a vibrotactile communication device. In Proc. of Symposium on Designing Interactive Systems, 2002, pp. 312-320.

[11] Chung, H., Lee, C.-H. J. and Selker, T. Lover's Cups: Drinking interfaces as new communication channels. Ext. Abstracts CHI 2006, ACM Press (2006), 375-480.

[12] Dodge, C. The bed: A medium for intimate communication. Ext. Abstracts CHI 1997: ACM Press (1997), 371-372.

[13] Brave, S. and Dahley, A. inTouch: A medium for haptic interpersonal communication. Ext. Abstracts CHI 1997, ACM Press (1997), 363-364.

Toward Haptic Mathematics: why and how

C. Bernareggi
Dipartimento di Scienze dell'Informazione
via Comelico 39/41
20135 Milano, Italy
bernareggi@dsi.unimi.it

A. Marcante, P. Mussio, L. Parasiliti Provenza
DICo - via Comelico 39/41
20135 Milano, Italy
+39 02 50316290
{marcante, mussio, parasiliti}@dico.unimi.it

Sara Vanzi
DICO - via Comelico 39/41
20135 Milano, Italy
anze@fastwebnet.it

ABSTRACT
Understanding a mathematical concept, expressed in a written form, requires the exploration of the whole symbolic expression to recognize its component significant patterns as well as its overall structure. This exploration is difficult for visually impaired people whether the symbolic expression is materialized as an oral description or a Braille expression. The paper introduces the notion of Haptic Mathematics as a digital medium of thought and communication of mathematical concepts that adopts the nomenclature and language of Mathematics and makes its expressions perceptible as sets of haptic signals. As a first step toward Haptic Mathematics, the paper presents a system adopting an audio-haptic interaction whose goal is to enable visual impaired or blind people to reason on graph structures and communicate their reasoning with sighted people. The paper describes a first system prototype and some preliminary usability results aimed at evaluating the effectiveness of the proposal.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Haptic I/O, Interaction styles.* H.1.2 [**Models and Principles**]: User/Machine Systems – *Human factors.*

General Terms
Design, Experimentation, Human Factors, Theory.

Keywords
Haptic, Multimodal Interactive Systems, Blind Users.

1. INTRODUCTION
Notation is widely recognized as "a tool of thought" in reasoning and communication [4],[1], [5]. However, perceiving a notation – i.e. recognizing the expressions in a notation through human perception capabilities – is a critical problem, which has not received enough attention, to the best of our knowledge. In [4], Iverson has identified the properties a scientific notation should embody [1]: it should combine the executability and universality ensured by programming languages with the properties offered by mathematical notations, i.e. (i) ease of expressing constructs arising in problems; (ii) suggestivity; (iii) ability to subordinate detail; (iv) economy; (v) amenability to formal proofs. Among these, properties (i), (ii) and (iii) depend on human perception and abilities. According to these properties, the notation is characterized as the set of perceptible forms of symbols and their relations aimed at composing a perceivable message. Different abilities of perception require adopting different notations.

In the case of mathematical notation, a concept is usually expressed in a written message, e.g. a formula or a graph. Interpreting the written message requires the exploration of the whole symbolic expression to recognize its component significant patterns as well as its overall structure. This exploration may require several phases in which attention is focused on the whole structure or on its components and back. The process is difficult for visually impaired users whether the symbolic expression is materialized as an oral description or a Braille expression: it can be difficult to perceive the above-mentioned properties of expression easiness, suggestivity and ability to subordinate detail. The oral description does not permit to grasp the whole symbolic expression: it just tells the sequence of symbols according to the reading direction and it does not permit to come back from whole to parts and back. A Braille expression may support the possibility to stay and come back on a significant component but requires a 2D static print, and it is not possible to update or annotate it in an interactive way. The problem is stressed when graph structures have to be studied and built: understanding a graph structure needs also the localization of nodes and edges in the space.

Building on these remarks, the paper proposes an approach based on audio-haptic interaction whose goal is threefold:

1. to allow a visually impaired person to explore a graph based structure reconstructing from local perception the whole structure, thus permitting her/him to focus on the whole and to recognize its significant subparts;
2. to allow a visually impaired person to build a graph expressing the desired constructs;
3. to allow an interactive mathematical reasoning and communication among a community of sighted and visually impaired persons.

The approach introduces the notion of Haptic Mathematics as a digital medium of thought and communication of mathematical concepts that adopts the nomenclature and language of Mathematics and makes its expressions perceptible as sets of haptic signals. We intend haptic signal as signal produced by technologies that interfaces the user via the sense of touch by

AVI'08, May 28-30, 2008, Napoli, Italy.

applying forces, vibrations and/or motions to the user, such as dynamic tactile display or a force feedback device.

A system for building and exploring graph structures according to the Haptic Mathematics approach, has been designed and early prototyped and tested. This system reinforces haptic interaction with audio and visual feedback: audio signals help to locate nodes and their relations; visual feedbacks are needed for improving communication with sighted users.

The paper introduces related work in the haptic interaction fields and then defines, in Section 3, Haptic Mathematics and its reasons. A model for audio-haptic interaction system is briefly presented in section 4. Section 5 presents the prototype and the preliminary usability proofs aimed at evaluating the effectiveness of the proposal. Section 6 concludes with future work.

2. RELATED WORK

In recent years, the notion of "haptic" has been associated with mathematics as related to the multimedia and multimodal tools to support the study of mathematics for visually impaired or blind people. Actually, making, exploring and understanding graphical representations are challenging activities for blind people. Since technical drawings are crucial in scientific reasoning, most blind people are prevented from straightforwardly accessing scientific knowledge. The main hindrances are partly due to the limitations of non-visual perception and partly to the inadequacy of tools for making and exploring non-visual representations of graphics. Much work has been undertaken to represent some categories of drawings in non-visual formats, especially through tactile and audio-tactile representations, as well as through multimodal interactive systems (MISs). Tactile images are usually a simplification of 2D drawings, which are embossed on paper to be perceived by touch. In order to increase the quantity and to improve the quality of details in tactile representations, Vanderheiden proposed audio-tactile images [10]. A tactile image is placed over a touch sensitive tablet so as non-speech audio cues and speech messages can be linked to hot spots and listened on demand by pressing the fingertip on the tablet. Both tactile and audio-tactile images are suitable for exploring 2D drawings. The process of making or editing a tactile or audio-tactile drawing is not achieved through direct manipulation of the drawing (e.g. a new drawing is generated and embossed on paper) and it is asynchronous with respect to exploration. Furthermore, visually impaired individuals can partially control the preparation of an image to be embossed for tactile understanding. In order to combine non-visual manipulation and exploration of drawings in a consistent interaction paradigm as well as to improve non-visual understanding of sophisticated drawings, MISs were studied and designed as the subsequent development of visual interactive systems [2]. Up to now, MISs have been proven to be effective especially to explore line graphs (e.g. function diagrams) [11], to understand bar graphs [6], to create and manipulate bar graphs [7]. All of these systems are focused on representing 2D diagrams, even if the workspace of most haptic devices is a 3D one. Fritz et al. proposed a method for haptic rendering of 3D data plots [3], nonetheless the third dimension is mainly used to locate additional haptic cues or for comparing diagrams. MISs have been designed also to present UML diagrams [8]. The TEDUB system implements an exploration paradigm for UML diagrams by using a force feedback joystick and speech and sound messages.

3. HAPTIC MATHEMATICS: WHY

A new notation that may exploit also non-visual senses for mathematical concepts employed in scientific reasoning is needed in the case where visual perception is limited or lacking. Scientific reasoning is strongly related to reading, understanding and manipulating text, mathematical expressions and graphical representations. Due to limitations of non-visual perception, visually impaired people face challenges in accessing scientific knowledge. Non-visual text understanding does not pose relevant difficulties. The exploration of text visual rendering can proceed either from left to right or browsing the tree structure. Speech and tactile tools can straightforwardly implement these exploration strategies. On the contrary, mathematical expressions present challenging issues. In order to comprehend how mathematical expressions can be non-visually read and understood, it is useful to compare visual reading and tactile and speech exploration. A model for visual understanding of mathematical expressions was proposed by Stevens [9]. This model works as follows. A mathematical expression is visually scanned by the reader, whose gaze may rest upon the expression for a while. A mental representation of the surface structure of the expression is formed. This representation is checked for understanding and a syntactic analysis is achieved before executing transformations. The initial part of this process poses difficulties for a visually impaired reader as it implies the existence of an external memory (e.g. paper, a blackboard or a screen), which permanently displays the representation of the mathematical expression. The transient nature of the speech signal does not provide a permanent representation of the structure to read: consequently, it is impossible to get different views at a mathematical expression and to directly access specific portions (e.g. to immediately read a numerator or an exponent, etc.). Reading by touch is a more active style of exploration. The external memory is not absent, but it is extremely reduced with respect to the one accessible by sight. It is generally made up of a line of about forty Braille cells. Although rightward, leftward, upward and downward movements update the external memory with new portions of the mathematical expressions, at any one time, a small portion of it can be accessed. Moreover, under the fingertips, at most two Braille cells can be perceived at any one time. Consequently, it is difficult to get an overall tactile glance at the whole expression or at its sub-expressions, so planning the reading process is far more difficult than by sight. Exploration problems are more severe in non-visual understanding of graphical representations. Images are visually analyzed hierarchically, from the overall structure down to the fundamental elements. Aggregations of basic elements are isolated instantly by sight and mutual relations are singled out. This global comprehension of a graphical representation enables planning of different exploration paths, which lead the reader to actively understand concepts described by the image. Through haptic and auditory perception small portions of a graphical representation can be perceived locally (e.g. on a tactile image or by moving the stylus of an haptic device) and mutual relations are increasingly identified as soon as new basic components are perceived. Therefore, exploration strategies can be hardly planned and understanding tends to be slow and difficult. The MIS being introduced in the following sections aims at providing basic audio-haptic primitives and interaction paradigms to enable creation, manipulation and exploration of graph structures. Thanks to the use of the Phantom haptic device a 3D workspace is available. Hence, 3D structures are represented as they are, with

no need for perspective technique, which are hardly usable in non-visual exploration. Moreover, the third dimension is also employed to simplify the perception of some graph structures whose planar description is very complex in the number of edges.

4. HAPTIC MATHEMATICS: HOW

As a first step toward Haptic Mathematics, we have designed a MIS – specifically, an audio-haptic interactive system – for graph creation and exploration that represents graphs and graph components – *nodes* and *edges* – in the 3D-space according to the Haptic Mathematics approach, i.e. permitting the interaction with them through haptic input and output signals (reinforced with audio and visual feedbacks for locating graph components in the space and improving communication with sighted people). The system allows users to interact with nodes and edges in the 3D-space as *virtual entities*, defined as virtual, dynamic, open, non-stationary systems [2].

The 3D virtual space, where graphs are set out, is tangible and discrete; it consists of *volume elements*, known as *voxel*s. The audio-haptic system we have designed represents graph nodes through voxels and graph edges as set of voxels connecting two graph nodes in the 3D-space. We also have the empty space composed by all those volume elements that do not belong to the graph. Based on these observations, we define a virtual entity voxel to manage the interaction with a volume element that can be a graph node or an empty element in the space. Furthermore, we define the virtual entity edge as an ordered pair of voxels representing two nodes in the graph. We model the 3D-workspace, where creating a new graph or modifying an existing one, as a graph itself – a grid graph N×N×N – to avoid visually impaired users from getting lost in the 3D-space. The grid nodes are represented by N^3 voxels in the space modeled themselves through the virtual entity voxel. They are the locations where a graph node can be placed. The grid edges are sets of voxels that permit to guide the user in the workspace exploration from a grid node to another one along three different orthogonal directions. Entities voxels are the atomic virtual entities our audio-haptic system is composed of. The system itself is a (composite) virtual entity – denoted ve_{MIS} – that is not a component of any other ves. It is composed of the virtual entity workspace, which models the grid-workspace, in the graph creation phase, and the target graph, in the graph exploration phase. In both cases, virtual entity workspace is composed of entities voxels (at least one) and edges (zero or more). The virtual entity ve_{MIS} is also composed of entity emptyspace, which models all the volume elements that do not belong to the graph, modelled by the voxels in the state *no-perceptible*. Finally, virtual entity proxy models the correspondent in the virtual 3D-space of the haptic device. Each virtual entity in the system reacts to input events that are caused by user actions, i.e. the physical operations the user performs on the perceivable manifestation of the system through the available input devices. The output events are physical events generated by the machine through the available output devices. According to our Haptic Mathematics approach, the input and output events are haptic, in that they are captured and shown through haptic devices, which interface the user via the sense of touch, by applying forces, vibrations and/or motions to the user, to permit, through haptic perception, reasoning and conversation on graph concept. In addition to haptic signals, we also take into account visual and audio signals as output events.

5. TOWARD HAPTIC MATHEMATICS

A system to realize Haptic Mathematics has been developed according to the model presented in the previous section. The system is composed by two different working environments: creation and exploration environment. Creation environment enables the user to move in a 3D space, to place and perceive nodes, to delete nodes and to connect nodes with edges. The exploration environment is meant to explore a graph, namely a set of labeled or not labeled nodes connected with labeled or not labeled edges. Creation environment was implemented. The workspace is made up of a 9×9×9 haptic grid (see a grid portion in Figure 1). The intersection points between lines in the grid are the locations where a graph node can be placed. Haptic interaction was modeled so as to constrain the user's movements either to lines or intersection points. When the user leaves a point, a force attracts the haptic device stylus to the nearest point. Therefore, the user is guided along the grid by the force feedback, thus avoiding from getting lost in the 3D workspace.

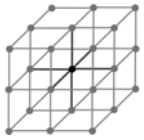


Figure 1. A portion of the 9×9×9 haptic grid.

Nodes can be placed in the grid by pressing a button on the stylus of the haptic device. An edit box enables the user to give a name to the node. Nodes can be perceived through haptic feedback since a vibrational effect is triggered when a node is met. In creation environment haptic feedback is reinforced with speech messages. Especially, coordinates for the current position and node names can be read by speech output on demand.

Up to now the system prototype has been evaluated with respect to three major usability problems: (i) how to represent grid elements so as to be perceived through haptic feedback; (ii) how to represent graph nodes and edges as haptic elements; (iii) how to enable users to navigate in the grid and identify graph nodes. The technique used to evaluate the system consisted in the observing form called "think aloud", where the user is asked to perform a task and talk about what s/he is doing as s/he is being observed. For each task two variables were monitored: the execution time and the number of faults. In order to address the first 2 problems, two experiments were set out with 2 blind users expert in computer science and with the haptic device. The results of these experiments led to implement a system prototype to be tested with blind users inexperienced of haptic interaction. In the first experiment, 2 haptic representations of the geometric characteristics of the grid were compared:

- grid points represented as haptic spheres and grid connections as haptic cylinders. The stylus of the haptic device could be moved either inside a sphere or inside a cylinder;

- grid points represented as haptic points and grid connections as haptic lines. The stylus of the haptic device was constrained to points or could be moved along lines.

The users were asked to move the haptic stylus in the grid and tell the observer whether a grid point or a connection was touched. The first representation posed problems because users were not able to follow a straight direction inside a cylinder. They could

say when a grid point was reached no sooner than 30 sec and after touching at least two connections leaving the grid point. With the second representation, users were able to move in the grid and say that a grid point was reached in 5 to 10 sec without loosing the right direction. This experiment affected the implementation. OpenHaptics library makes available two touch models: the contact model which enables the user to move the stylus wherever in the 3D space and touch the surface of the shapes and the constraint model which constrains the stylus to shapes through attraction forces. The second model was employed to represent the grid. The second experiment aimed at comparing two haptic representations of graph nodes: (i) as haptic spheres, and (ii) as haptic points perceivable through a vibrational effect. The same users were asked to follow a path and say when graph nodes were perceived. The users confused spheres as grid points in a 9*9*9 grid. Instead, they were able to identify all of the nodes represented through a vibrational effect. Hence, a vibrational effect was implemented to represent graph nodes. Finally, a third experiment was set out with 5 blind users not experienced with the haptic device in order to assess movement in the grid, graph nodes placing and graph nodes perception. The users were asked to perform six tasks. Three tasks concerned movement in the grid: to reach a grid point adjacent to the current one, to touch six grid points on the vertices of a cube and to follow a certain path to find a target position. Based on our observations with experienced users, the execution time expectation for each task was at least of one minute, whereas they performed the tasks successfully in 10 to 50 sec. Two tasks concerned the identification of graph nodes. In the first task users were asked to say aloud when a node was found along a line and in the second task users were asked to count the nodes along a certain path. The execution time expectation was again about 1 minute. Both tasks were executed in 5 to 50 sec. Nonetheless, problems were remarked in perceiving graph nodes: the vibrational effect was regarded as misleading when two contiguous nodes were found. Moreover, all the users made at least one mistake to find out nodes along the vertical direction. The last task concerned placing graph nodes. Users were asked to place 3 graph nodes along a certain path. The task took about 30 sec. No faults were observed.

6. CONCLUSIONS

In this paper, we have presented the audio-haptic system for graph creation and exploration, designed and developed in order to allow visually impaired or blind people to reason on graphs and to permit the conversation among communities of sighted and visually impaired people. This MIS represents a first step toward haptic mathematics. The results provided by the evaluation tests give insight on the use of haptic tools in the exploration of mathematical structures and suggest the enhancement of the creation environment through speech messages and non-speech cues (e.g. with vibration effects). Some preliminary experimentation of the exploration environment with expert users has been undertaken. The results indicate that further implementation and evaluation work is needed, especially as for guided exploration and understanding of graphs complex in the number of edges. As far as communication between sighted and non-sighted users is concerned, we are designing a set of experiments based on adaptive camera positioning in a 3D virtual environment.

7. ACKNOWLEDGMENTS

We would like to thank the members of KAEMaRT laboratory at Politecnico of Milano, especially prof. Umberto Cugini and prof. Monica Bordegoni. The present work is partially funded by the 12-1-5244001-25009 FIRST grant of the University of Milan and by the Italian PRIN 2006 PUODARSI project (Product User-Oriented Development Based on Augmented Reality and interactive Simulation). The work is also supported by @Science consortium (ECP-2005-CULT-038137) and the Library of Computer Science of University of Milan.

8. REFERENCES

[1] Cajori, F. A History of Mathematical Notations, Volume If, Open Court Publishing Co., La Salle, Illinois, 1929.

[2] Fogli, D., Marcante, A., Mussio, P., Parasiliti Provenza, L., and Piccinno, A. 2007. Multi-facet Design of Interactive System through Visual Languages, in F. Ferri (Ed.) "Visual Languages for Interactive Computing: Definitions and Formalizations", HERSHEY PA: IGI Global, 2007, 174-204.

[3] Fritz, J. P., Way, T. P., and Barner, K. E. 1996. Haptic Representation of Scientific Data for Visually Impaired or Blind Persons. In Proceedings of the Eleventh Annual Technology and Persons with Disabilities Conference, California State University, Northridge, LA, CA, 1996.

[4] Iverson, K. E. 2007. Notation as a tool of thought. SIGAPL APL Quote Quad 35, 1-2 (Mar. 2007), 2-31.

[5] Marcante, A., Mussio, P. 2006. Electronic Interactive Documents and Knowledge Enhancing: a Semiotic Approach, the Document Academy, October 13-15, 2006, UC Berkeley, Berkeley, CA, USA

[6] McGookin, D. K., and Brewster, S.A. 2006. MultiVis: Improving Access to Visualisations for Visually Impaired People. In extended proceedings of CHI 2006 (Montreal, Canada), ACM Press.

[7] Mcgookin, D. K. and Brewster, S.A. 2006. Graph builder: Constructing non-visual visualizations. In ICAD 2006 (London, UK).

[8] Petrie, H., Schlieder, C., Blenkhorn, P., Evans, G., King, A., O'Neill, A.-M., Ioannidis, G.T., Gallagher, B., Crombie, D., Mager, R., and Alafaci, M. 2002. TeDUB: a system for presenting and exploring technical drawings for blind people. In K. Miesenberger, J. Klaus and W. Zagler (Eds.), LNCS 239. Heidelberg: Springer Verlag

[9] Stevens, R.D. 1996. Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People. PhD Thesis, Dept. of Computer Science, University of York.

[10] Vanderheiden, G. C. 1989. Nonvisual alternative display techniques for output from graphics-based computers. Journal of Visual Impairment and Blindness 83, 8, 383-390.

[11] Yu, W., and Brewster, S. A. 2003. Evaluation of multimodal graphs for blind people. Universal Access in the Information Society, 2, 2, 105–124.

The Need for an Interaction Cost Model in Adaptive Interfaces

Bowen Hui
Dept. of Computer Science
University of Toronto
bowen@cs.utoronto.ca

Sean Gustafson, Pourang Irani
Dept. of Computer Science
University of Manitoba
{umgusta1,irani}@cs.umanitoba.ca

Craig Boutilier
Dept. of Computer Science
University of Toronto
cebly@cs.utoronto.ca

ABSTRACT

The development of intelligent assistants has largely benefited from the adoption of decision-theoretic (DT) approaches that enable an agent to reason and account for the uncertain nature of user behaviour in a complex software domain. At the same time, most intelligent assistants fail to consider the numerous factors relevant from a human-computer interaction perspective. While DT approaches offer a sound foundation for designing intelligent agents, these systems need to be equipped with an *interaction cost model* in order to reason the impact of how (static or adaptive) interaction is perceived by different users. In a DT framework, we formalize four common interaction factors — information processing, savings, visual occlusion, and bloat. We empirically derive models for bloat and occlusion based on the results of two users experiments. These factors are incorporated in a simulated help assistant where decisions are modeled as a Markov decision process. Our simulation results reveal that our model can easily adapt to a wide range of user types with varying preferences.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Miscellaneous; I.2.11 [**Artificial Intelligence**]: Intelligent agents

General Terms

Interaction models and techniques, User interaction studies

Keywords

Information processing, visual occlusion, bloat, perceived savings

1. INTRODUCTION

Software customization has become increasingly important as users are faced with larger, more complex applications. For a variety of reasons, software must be tailored to specific individuals and circumstances [14]. Online and automated help systems are becoming increasingly prevalent to assist users identify and master different software functions [11]. In this paper, we focus on *adaptive interfaces* where the user's preferences over interface attributes (e.g., location, transparency, perceived savings of interface widgets) determine how the system customizes the interface. Our objective is to develop adaptive interfaces that maximizes the user's ease of interaction with the system.

Many decision-theoretic (DT) approaches have been applied to develop assistants that provide intelligent help for different users (e.g., [11, 1, 6, 3, 2, 13]). These approaches typically try to help the user accomplish a task more efficiently by using machine learning techniques to estimate user-specific information, such as the user's current task, whether the user needs help, or how frustrated the user is with the system. At the same time, every system action has a value (i.e., cost or benefit) that may be perceived differently depending on the user or the circumstance. In support of DT approaches, Horvitz proposed that a central principle in designing intelligent systems is the ability to evaluate the *utility* of system actions "in light of costs, benefits, and uncertainties" [10]. Following these approaches, we adopt a DT framework to design an agent that makes rational decisions about its customization under uncertainty.

In order to model the utility of system actions, we need to directly account for the impact that the system's customization actions have on the user. Since an interface is a composition of widgets, we design the system to adapt the interface by changing the attributes of individual widgets. We refer to such changes as *system actions*, which an intelligent agent may decide to take. However, different actions have different consequences: an adaptive interface that hides unused menu items may be preferable for one user because it saves him from scanning unnecessary functions (i.e., savings from processing extraneous items), but the same behaviour may be detrimental to other users who prefer to see all available functions (i.e., high tolerance to bloat). Furthermore, system actions have effects beyond immediate consequences. For example, a user who likes unused menu items hidden may find it annoying when he needs to use one of those functions in the future (i.e., cost of re-discovery). These consequences are, in fact, ways that a user determines the level of satisfaction with a software. Therefore, to quantify the impact of system actions, we identify interaction factors that are relevant to intelligent interfaces and formalize the costs and benefits of adaptive actions using an *interaction cost model*.

The benefits of developing an explicit interaction cost model are two-fold. From an HCI perspective, quantitative models enable designers to compare a variety of interaction mechanisms on the same grounds and predict the performance and satisfaction that new users experience with these mechanisms. From an intelligent systems perspective, the interaction cost model provides a way for the system to evaluate the utility of its own actions *before* adapting the interface. In developing DT systems, we employ the interaction cost model to evaluate the impact of different adaptive actions. By adopting the formalism that takes *user types* as parameters in the interaction cost model [13], the agent is able to quantify the im-

AVI '08, May 28-30, 2008, Napoli, Italy

pact of its actions with respect to specific types of users, and thus, adapting its overall behaviour toward specific user types.

In Section 2, we describe the approach of an intelligent system in a DT framework. Our focus is on modeling the utility of system actions. For this purpose, we identify four common interaction factors for adaptive interfaces — information processing, savings, visual occlusion, bloat — and formalize their cost models. In an effort to derive a quantitative model for occlusion and bloat, we conduct two experiments to explore the relevant parameters and structure in Section 3. Using these experiment results, we implement our interaction cost model in a simulated help system that adds or deletes unused menu items. The simulation is implemented as a *Markov decision process* (MDP) [17]. Our results show that the system is able to adapt its behaviour to different types of users. While usability studies are needed to confirm our simulation results, this work suggests that using a DT framework to model interaction benefits and costs is a formal, general, and useful approach.

2. DECISION-THEORETIC FRAMEWORK

In designing an adaptive interface system, we view the system as an intelligent agent that reasons about the impact that its actions have on the user. Considering adaptive menus in the context of interface bloat as an example, the agent observes which menu items have been selected, evaluates the (long term) utility of hiding one or more unused menu items, and carries out the action that is optimal for the user (i.e., (un)hiding menu items or doing nothing). In general, there are uncertainties in assessing the utility of system actions — how much faster is it for the user to search in that menu after hiding unused items, how tolerant is the user with respect to bloated interfaces, or how likely, and at what frequency, is the user going to require a hidden menu item in the (near) future? These questions illustrate the need to quantify two parts of the customization problem: (i) the costs and benefits of actions with respect to relevant interaction factors (e.g., "how fast", "how tolerant"), and (ii) the likelihood of events (e.g., the probability that the user is highly tolerant to bloat, the probability of a hidden menu item being executed in the near future). Since the focus of this work is utility assessment, we will assume probability estimation is feasible in the system.[1] Section 2.1 explains the concept of utility and its relation to user preferences. In Section 2.2, we turn to the discussion of adaptive interface systems and relevant interaction factors. Section 2.3 formalizes our interaction cost model and explains how it is used in the DT framework.

2.1 Objective Value versus Subjective Utility

Utility theory is used to systematically quantify the total costs and benefits of decision outcomes: if a person has higher utility in one situation than another, that person prefers the former situation over the latter. In designing intelligent systems, we want to know the utility of adaptive actions in a way that reflects the user's preferences over possible interaction mechanisms. Intuitively, our goal is to quantify the utility of specific system actions with respect to the factors that influence the user's interaction experience. Note that utility is subjective in nature since it reflects individual preferences. Therefore, our goal is to determine the perceived utility given an interaction setting (i.e., in terms of system actions and application states). To do this, we first define an interaction cost model that specifies the *objective value* of an interaction setting. Then, we introduce user characteristics that influence individual preferences for interaction, in terms of their objective quantities. Lastly, we define a parametric utility function that maps the objective value and user characteristics to a *subjective utility*. When computing the utility of actions to evaluate which one is best, the system uses this utility function by "plugging in" the necessary parameters based on the current state and action. Since this function defines subjective utility, the system's reasoning process chooses the action that best satisfies the user's interaction preferences.

[1]Indeed, the user modeling literature provides a suite of machine learning techniques that can be used to estimate user information quickly (e.g., see [13]).

2.2 Impact of Intelligent Actions

Different interface designs serve different purposes. Generally speaking, there are two main objectives in intelligent assistance: (i) to minimize user effort in task completions, and/or (ii) to maximize the ease of interaction during application use. In desktop applications, many kinds of system actions can be implemented to (potentially) achieve these objectives. Examples include: doing mundane work on the user's behalf (e.g., auto-completion), moving widgets to another location for more convenient selection (e.g., adding, moving, deleting widgets), changing the delivery of widgets (e.g., via animation), changing the presentation of widgets (e.g., level of transparency), sending reminders (e.g., using a text balloon), making suggestions (e.g., via a toolbar), asking questions explicitly (e.g., via a dialog box), etc. Each of these actions come with associated benefits and costs. For example, adding a frequently executed item to the menu can increase selection convenience at the risk of inducing more bloat. Among the many interaction factors proposed in the literature, we focus on the following:

- *information processing*: cost of evaluating a set of items
- *savings*: manual effort that would have otherwise been required
- *occlusion*: cost of displaying widgets in the user's workspace
- *bloat*: cost of displaying excessive functionality

For a detailed rationale of our choices, please see [12]. We present a formal model for these interaction factors in Section 2.3.

2.3 An Objective Interaction Cost Model

Since processing and savings are well-studied interaction factors, we adopt the existing quantitative models. Specifically, the cost of processing is linear for naive users [9] and logarithmic for expert users [8, 15]. To combine the two models, we define $proc = f(Expertise, Len)$, where $Expertise$ is either naive or expert, Len is the number of items to process, and f is linear for naive users and logarithmic for experts. To model savings, we adopt the GOMS-KLM model [5] that quantifies user effort in terms of the mode used to carry out an event, such as menu selection using the mouse versus keyboard shortcuts. We define the objective savings as $quality = Num * GOMS(Mode)$, where Num is the number of events and $GOMS(Mode)$ is the effort required for that mode.

Both occlusion and bloat are often mentioned in the design literature but, to our knowledge, there are no formal attempts to model them. For this reason, we conduct experiments to explore the parameters and structure of a quantitative model. As a result, we obtained $o = f(Opac, B)$ as objective occlusion, where $Opac$ is the level of opacity of an occluding widget, and B is whether the user's immediate focus is occluded. For objective bloat ("excess"), we obtained $xs = f(Unused)$, where $Unused$ is the difference between the number of functions shown and used. We refer the reader to Sections 3.1 and 3.2 for the definition of these functions and the corresponding experiments.

3. EXPERIMENTS AND SIMULATIONS

We conduct experiments to empirically derive quantitative mod-

els for occlusion and bloat. The analysis of each experiment investigates the relevance of the tested parameters and empirically derives a functional form. While occlusion and bloat have largely been neglected in experimental studies, our results show that both factors cause an interaction effect. This indicates the importance of modeling these two factors in our interaction cost model.

Both experiments had 12 volunteer participants from a graduate computer science pool, all of whom have a good command of written English and no motor control deficiencies.

3.1 Occlusion

The purpose of this experiment is to derive an objective model of occlusion in the context of intelligent assistance. We simulated a typing task by focusing on the task of typing a single letter in a sentence. Each trial consists of the user typing the highlighted letter (i.e., the target), ignoring or dismissing a pop-up box (varied in 4 parameters defined below), and then typing a second highlighted letter. We measured the time between the two typed letters in each trial. A screenshot of this program is shown in Figure 1.

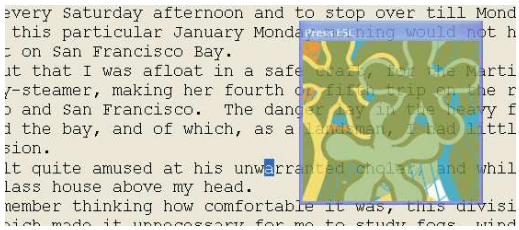


Figure 1: Screenshot showing a pop-up dialog box of size 200×200 pixels at 80% opacity in a typing task.

Each trial varied in 4 parameters of the dialog box: direction, size, opacity ($Opac$), and proximity — yielding a total of 480 configurations. In addition, we logged the intersecting area between the dialog box and the target letter. The measured task completion time in each trial is a function of these 6 variables.

To create a simpler model, we used factor analysis to determine the most important variables. We used ANOVA to determine whether each set of user data came from separate distributions, and the F-test to determine the complexity of the model. As a result, we found occlusion is best explained by a non-linear function: $o = f(B, Opac)$, where B is an indicator to denote the presence of overlap between the dialog box and the target, and $Opac$ is defined above. When $B = 0$, $o = c_0$, and when $B = 1$, the best approximation is a cubic function $o = c_3 Opac^3 + c_2 Opac^2 + c_1 Opac + c_0$ for half of the users and a linear function $o = c_1 Opac + c_0$ for the other half, where $c_0, .., c_3$ are empirically derived constants for individual users.

3.2 Bloat

The purpose of this experiment is to derive an objective model of bloat in the context of intelligent assistance. We designed a menu selection task with an interface that has the same menu structure as Microsoft Word but using abstract menu labels. This experiment has 4 conditions varying in the number of menu items shown ($Shown$): 18, 62, 107, and 152, out of a total of 152 possible menu items. In all the conditions, we fixed the number of menu items used ($Used$) to 15. The target items in the selection task are randomized across conditions. In each trial, participants follow an instruction (e.g., Fruits → Papaya) and select the target menu item. We measured the successful selection time of target menu items. A screenshot of the program is shown in Figure 2.

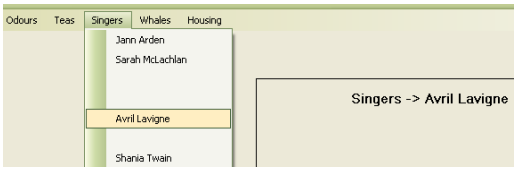


Figure 2: Screenshot showing the target menu item and instructions on the right. Notice this menu has many empty slots.

Each participant carried out 15 trials per condition. With 4 conditions, each participant carried out a total of 60 trials in the experiment. We counterbalanced the order of blocks using a size 4 Latin square. The measured selection time in each trial is a function of $Shown$ and $Used$. Conceptually, we defined $Unused$ as the number of items shown but not used. Using this definition, we used ANOVA and an F-test and found that bloat is best approximated as a linear function $xs = c_1 Unused + c_0$ for most users, and as a quadratic function $xs = c_2 Unused^2 + c_1 Unused + c_0$ for 1 user, and as a cubic function $xs = c_3 Unused^3 + c_2 Unused^2 + c_1 Unused + c_0$ for 1 user, where $c_0, .., c_3$ are empirically derived constants for individual users.

3.3 Markov Decision Process

To put the interaction cost model to use, we designed a system that adapts menus in simulation. The first step in the design is to identify the relevant interaction factors for this domain. Given the objective cost models of these factors (as defined in Section 2.3), we introduce user characteristics and define the overall subjective utility function. This function is used in the system to evaluate the goodness of its actions. In the simulation, we assume we know the user characteristics and model the customization problem as an MDP[2]. In this way, the agent optimizes its adaptive actions with respect to the user's preferences over repeated interaction with the system. When an MDP is solved, we obtain a *policy* that maps (application and user) states to an optimal action. For a detailed introduction to MDPs, the reader is referred to [4].

The possible actions of this system are to add a menu item, delete a menu item, or do nothing. For simplicity, we use bloat and savings in defining this system's interaction cost model[3]. These two factors are relevant because the system can remove or introduce items that offer potential savings. To compute the subjective utility of system actions involving these factors, we use the following functions:[4] $savings = f(Quality, N, D, F, I)$ — represents the perceived savings of the resulting interface, given the objective quality of savings, how much help the user needs (N), how distracted the user is in general (D), how frustrated the user is with the system (F), and whether the user generally likes to work independently (I); $bloat = f(XS, T, D)$ — represents the perceived bloat of the resulting interface, given the objective excess of bloat, the user's tolerance toward bloat (T denotes whether users are *feature-keen* or *feature-shy* [16]), and how distracted the user is with more functions available (D). Finally, the overall utility of an action is the weighted sum of $savings$ and $bloat$.

[2]In reality, this problem should be modelled as a *partially* observable MDP because we cannot know the user with complete certainty. However, since machine learning techniques are available for learning the user, we assume we can observe the user here.

[3]In general, the cost of processing, interruption, and disruption also play a role in adaptive menus.

[4]Due to a lack of space, we refer readers for a detailed account of these models elsewhere [13, 12].

In the simulation, we discretize the user variables to be binary and tertiary (e.g., F has 3 values, representing the user being highly, somewhat, and not at all frustrated). With 5 user variables, the system's utility function accounts for a total of 162 user types. In addition, the MDP dynamics are defined to reflect changes to the interface (adding/deleting) can distract and frustrate the user. In this way, the system does not risk taking adaptive actions when dealing with highly distracted/frustrated users.

We conducted two simulation runs. The first one investigates the effect of bloat and the second one explores the system's adaptability toward various user types in the model. At any point in time, the adaptive menu can have 1 to 6 items shown, and the system policy suggests to add an item, delete an item, or do nothing, based on the menu state and the user's type. In the first simulation, we defined a constant value for savings and focus only on bloat. A qualitative description of the results is presented in Table 1, where the number of menu items shown is categorized as "few" (less than half), "many" (more than half), or "any" (any value between 1 to 6).

Distractibility	Tolerance	Shown	Policy
low/medium	keen	any	add
high	keen	few	add
low	shy	many	delete

Table 1: Results showing the effect of bloat.

Generally, we see that the system adds items for feature-keen users, even when they are highly distracted because an addition offers enough savings to tradeoff the cost of annoying the user. For all other combinations, the system opts to do nothing.

In the second simulation, we re-introduced savings to compare the adaptive behaviour toward different user types. When the user's frustration and independence levels are low and the neediness level is high, we expect this type of user to be most receptive to help. We define this user type as our "best case". Analogously, we define the "worst case". The qualitative results are shown in Table 2.

Case	Distractibility	Tolerance	Shown	Policy
best case	low	keen/shy	any	add
best case	medium/high	keen	any	add
worst case	low	keen	any	add
worst case	low	shy	many	delete
worst case	medium	shy	many	delete

Table 2: Results showing the effect of user types.

For the best case user type, the system tends to suggest adding an item because these users are receptive to adaptive help. For the worst case user type, the system is much more conservative and only adds an item for low distractibility and feature-keen users. The system deletes items when many are shown for feature-shy users who are not highly distractible. For all other combinations, the system opts to do nothing in fear of distracting the user by changing the interface. Note that there are 160 user types "between" the best and worst cases. These results show that the system is able to adapt to many different user types.

4. CONCLUSIONS

In this paper, we proposed a decision-theoretic approach to account for the varying user preferences with software interfaces. We modeled four interaction factors, that when combined, result in an interaction cost model that forms part of a utility function used to explain different interaction preferences. Our interaction cost model is highly flexible so that formal models can be refined simply by changing the corresponding formula in the model. Additionally, our implementation shows that designers can pick and choose the interaction concepts from the framework that are relevant for their application. By modeling the costs and benefits of various interaction factors, intelligent systems can reason the impact of its actions and optimize its behaviour for different users with varying interaction preferences.

5. REFERENCES

[1] D. Albrecht, I. Zukerman, and A. Nicholson. Pre-sending Documents on the WWW: A Comparative Study. In *Proc. of IJCAI*, pp. 1274–1279, 1999.

[2] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie, and A. Mihailidis. A decision-theoretic approach to task assistance for persons with dementia. In *Proc. of IJCAI*, pp. 1293–1299, 2005.

[3] T. Bohnenberger and A. Jameson. When policies are better than plans: decision-theoretic planning of recommendation sequences. In *Proc. of IUI*, pp. 21–24, 2001.

[4] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of AI Research*, 11:1–94, 1999.

[5] S. Card, P. Moran, and A. Newell. *Psychology of HCI*. Hillsdale, NJ: Erlbaum, 1980.

[6] C. Conati, A. Gertner, and K. VanLehn. Using Bayesian networks to manage uncertainty in student modeling. *Journal of UMUAI*, 12(4):371–417, 2002.

[7] K. Gajos, M. Czerwinski, D. Tan, and D. Weld. Exploring the Design Space For Adaptive Graphical User Interfaces. In *Proc. of AVI*, pp. 201–208, 2006.

[8] W. Hick. On the rate of gain of information. *Journal of Experimental Psych.*, 4:11–36, 1952.

[9] A. Hornof and D. Kieras. Cognitive modeling reveals menu search is both random and systematic. In *Proc. of CHI*, pp. 107–114, 1997.

[10] E. Horvitz. Principles of mixed-initiative. In *Proc. of CHI*, pp. 159–166, 1999.

[11] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumière Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Proc. of UAI*, pp. 256–265, 1998.

[12] B. Hui. A Survey of Interaction Phenomena. Technical report, Dept. of Computer Science, Univ. of Toronto, 2007.

[13] B. Hui and C. Boutilier. Who's Asking for Help? A Bayesian Approach to Intelligent Assistance. In *Proc. of IUI*, pp. 186–193, 2006.

[14] B. Hui, S. Liaskos, and J. Mylopoulos. Requirements Analysis for Customizable Software. In *Proc. of RE*, pp. 117–126, 2003.

[15] R. Hyman. Stimulus information as a determinant of reaction time. *Journal of Experimental Psych.*, 45:188–196, 1953.

[16] J. McGrenere, R. Baecker, and K. Booth. An evaluation of a multiple interface design solution for bloated software. In *Proc. of CHI*, pp. 163–170, 2002.

[17] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, NY, 1994.

Theia: Open Environment for Multispectral Image Analysis

Vito Roberto
Dipartimento di Matematica e Informatica
Università di Udine, Italy
vito.roberto@dimi.uniud.it

Massimiliano Hofer
Nucleus s.n.c., Udine, Italy
max@nucleus.it

ABSTRACT

Preliminary results of Theia, a software system for multispectral image visualization and analysis are presented. A new approach is adopted, based on modern design techniques and better tuned to the recent advancements in hardware. A careful implementation in the C++ language addresses the issues of time efficiency, openness to personalizations and portability by exploiting the advances of Open Source technologies. Experimental tests on multispectral images have given promising results towards the use of the system as a dynamic, interactive interface to massive data visualization, mining and processing.

Categories and Subject Descriptors

I.4.0 [**Image Processing and Computer Vision**]: General—*Image processing software*

General Terms

Multispectral analysis, image processing environment

Keywords

Multispectral, Hyperspectral, Image Processing, Visualization, Interactive Interfaces, Object Oriented Design, Open Source

1. INTRODUCTION AND MOTIVATIONS

The analysis of multispectral images is becoming a central issue in a number of research and managing tasks, such as environmental planning, medical diagnosis, archaeological survey, surveillance for both military and civilian applications. Problems arise from the data acquisition by heterogeneous sensory systems; the massive data sets to be handled; the need for efficient encoding in image transmission and archiving; the need for efficient algorithms for data visualization, mining and filtering.

As a consequence, a number of challenging topics are to be faced by the designers of automated systems. Commercial products are available like ENVI [1], or freeware like MultiSpec [2], which are the outcome of long-term research and development projects, and so are mature enough to address a large number of analysis tasks.

However, some of the available software frameworks have been designed for computers that couldn't afford to manipulate massive data sets efficiently. As a matter of fact, most computers can now process gigabytes of data in a second and load large data sets in the RAM. We take advantage of this state of affairs, which potentially influences the design of novel software systems; new opportunities can be explored towards the realization of interactive visual interfaces, with real-time dynamic processing of considerable amounts of information.

On the other hand, the fast growth and differentiation of applications suggests to adopt well-posed criteria of software design – inspired to modularity, readability and flexibility - offering the developer the opportunity to re-use existing modules and personalize the system to specified needs.

This paper presents preliminary results of Theia, a research project aimed at the design of an open software environment for multispectral image analysis, adopting the object-oriented approach. The paper is structured as follows: the next section contains an introduction to the system architecture, with details on the organization of modules and data flow. Section 3 reports its practical implementation and preliminary results. Section 4 contains our conclusions and perspectives for future work.

2. THEIA: STRUCTURE AND CONTROL

The main goals of the project are: clean object-oriented design; portability among significant client Operating Systems; extendability with new or customized components; flexibility in combining and re-using components; good performance in processing massive data sets; high interactivity with the user.

Theia addresses two basic functionalities, image visualization and processing. The whole system employs two cross-platform libraries: QT4 [3], v.4.3.3, a framework for GUI development distributed under GPL license v. 2.0 or 3.0; and LibTIFF [4], v.3.8.2, to read and write TIFF [6] files, under the X11/MIT license. A simplified scheme of the Theia class diagram has been reported in Figure 1. We also provide several classes for basic computations (not shown).

The 'filter' is a central concept in our architecture; it is meant to be an independent module. Looking more closely, it is a set of three abstract classes: the plugin – interface

AVI '08, 28-30 May , 2008, Napoli, Italy.

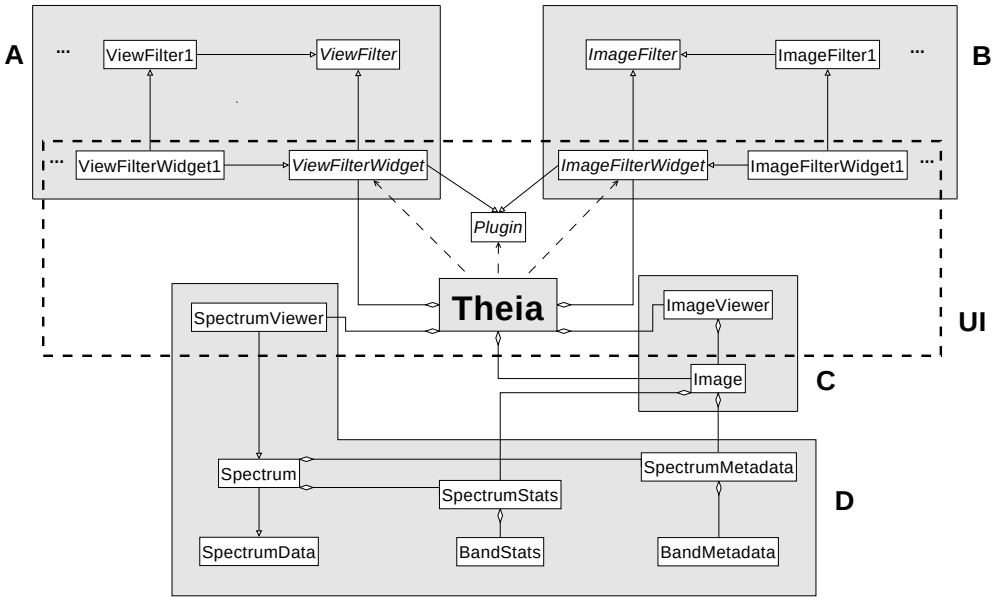


Figure 1: The Theia class diagram. Classes have been grouped into four blocks: A) Visualization: framework for real-time rendering; B) Framework for processing; C) Image handling; D) Spectral manipulation. The classes within the dashed rectangle handle the user interface, while those outside perform computations. User-interface and computation classes have been kept strictly separate throughout the project.

to exchange parameters with the main GUI; a processor to trigger the numerical computations and provide results; a widget component to visualize the graphical objects needed to set the parameters of the processor. Two kinds of processors have been designed, an ImageView and an Image filter, charged of visualization and numerical processing, respectively. A scheme of the interfaces of the ImageView has been reported in Figure 2.

As far as the data flow is concerned, in Figure 4 we report an example of user interaction with Theia, aimed at adjusting a visualization parameter.

3. IMPLEMENTATION AND RESULTS

Theia has been implemented on Linux for x86-64 platform; the C++ language has been preferred to Java™ because of its higher efficiency in processing massive data sets; in addition, it allows a more effective use of the O-O and generic paradigms, by offering solutions such as multiple inheritance, virtual base classes and templates.

One of the most effective solutions adopted to achieve efficiency in rendering is the 'attentional' processing, i.e., only visible data subsets are selectively processed, in such a way that interactivity and dynamic processing of the system are enhanced, and the performance does not depend on the loaded data size.

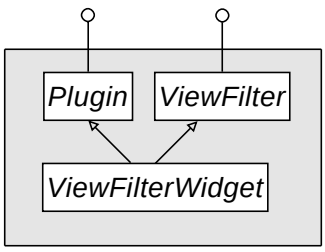


Figure 2: Abstract base classes of a visualization filter.

Besides the basic operations, a limited number of tasks have been implemented to explore the system performance: – Visualization: multispectral-to-RGB image mapping; pixel classification according to a distance measure with respect to reference spectra; difference-based false color rendering; – Processing: multispectral-to-multispectral image mappings; moving average filtering; band selection.

Tests have been carried out on a notebook equipped with 3 GB RAM and a CPU with a 2 GHz clock. Data were acquired by a MIVIS (Multispectral Infrared and Visible Imaging Spectrometer), providing images in 102 wavelength bands between 433 and 12700 nanometers, with 755x4000 pixel resolution, 16–bit sample precision, and a load of 587 MB per image. Theia currently operates on multichannel TIFF [6] and BSQ [5] files with ENVI [1] metadata encoding.

In all tests performed so far Theia exhibited a remarkable time efficiency: with a visible portion of 755x768 pixels, most visualization filters require a few tenths of a second for a complete refresh; much less, for the partial renderings needed while panning. The most CPU-intensive filters — e.g, false color segmentation – need at most 1.5 seconds for the first rendering, without activating multithreaded computations.

Due to the software tools that have been chosen, the user interface is reconfigurable and basic components can be cast by drag-and-drop on the working set.

An interesting feature of Theia is its openness: a developer can readily introduce additional modules. A well-defined set of interfaces allows for interaction between components. The main GUI knows how to deal with plugins: manages events, mouse ownership, drag-and-drop

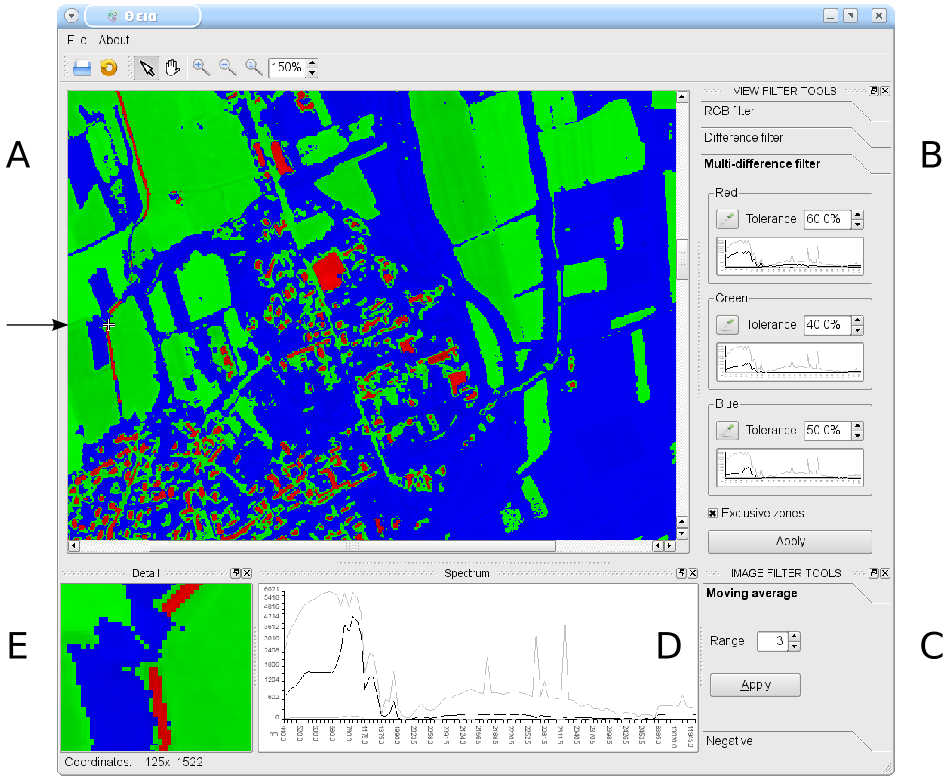


Figure 3: A screenshot of Theia showing the false color segmentation of three types of terrain. MIVIS data of the Friuli area (North-eastern Italy) have been processed. Five main panels are shown. A) The segmented image, output of the filter; B) The view filter configuration panel: a pixel has been selected (see the arrow on the left) and labelled with a color (red, green, blue) according to a distance measure with respect to the three spectra taken as reference, and reported on the panel; on the latter, each spectrum is complemented by two curves, reporting the minimum and maximum spectral values observed over all bands; horizontal scales are in nanometers; C) multispectral filtering panel (not active); D) Detailed spectral data of the selected pixel; E) Magnified image region surrounding the pixel, with coordinates (125, 1522).

of basic data, switching between components and driving their placement and visualization. It also receives signals

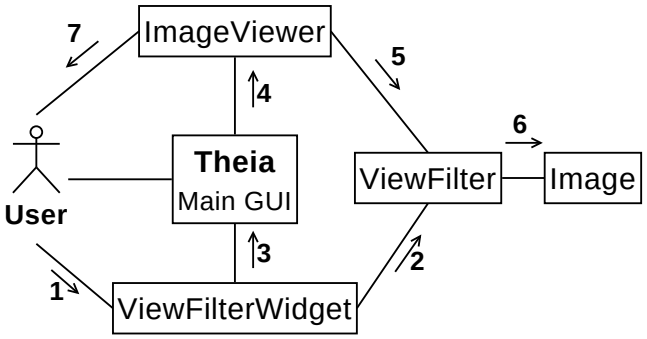


Figure 4: Message exchanges to adjust a visualization parameter. 1.–The user interacts with the filter GUI (Widget); 2.–The latter sets the filter parameters; 3.–The same notifies the main GUI that a refresh is needed; 4.–The GUI notifies to reprocess the visible data subset; 5.–The visualization filter is charged of the computation; 6.–The same accesses the data; 7.–The RGB image is served to the user.

requesting operations to be performed. A new filter can be developed without modifications to the main program, and in principle can be linked at run-time from a separate, shared library. Moreover, all basic components to process images and show data in the GUI are fully re-usable by any plugin component.

Portability is another issue addressed in our project. Not being coded in Java, Theia might undergo significant limitations for re-compilation on platforms with different data encodings (32-bit vs. 64-bit, for example); in addition, no standard graphics libraries are available, unlike in Java. Such problems have been taken in due care in the design of the framework, so that critical pieces of code are parameterized or encapsulated. Dependency on OS APIs has been avoided by limiting any external reference to the standard C++ library, QT4 and LibTIFF.

4. CONCLUSIONS AND PERSPECTIVES

We have designed, implemented and preliminarily tested Theia, a software system devoted to the analysis of multispectral images. Our main concerns were object-oriented design; high performance, to interactively process hundreds of megabytes of data; portability on heterogeneous platforms; openness, to develop personalized applications.

The preliminary results of our research work demonstrate that all such goals are within reach.

A well-posed O-O software design ensures a modular, readable and flexible code; two cross-platform libraries are included, QT4 and LibTIFF. The system as a whole provides a sound platform for developers of specialized applications.

Satisfactory performance has been obtained after a – still restricted – number of visualization and processing tasks. Computational efficiency has been achieved by adopting a number of solutions. Firstly, data processing modules have been kept physically separate from the other ones throughout the system; secondly, processing itself is accomplished in a selective way, i.e., only on the visible portion of the loaded data set; finally, a number of features of the C++ language have been exploited to optimize the performance, such as template specialization and inline method expansion for basic data types.

Portability is ensured by a careful implementation and tests have been conducted on Linux and Windows®. We also verified that adding new filters to the system, or linking additional modules from separate libraries, are both readily affordable tasks.

Our results encourage further research work and suggest that new approaches are possibile to multispectral processing GUIs, towards a fully interactive user experience.

Work is in progress to extend the system functionalities. Besides adding new filters, efforts are currently devoted to adding components to extract several types of ROI. The latter are to be used to restrict elaborations for further processing and as supports for spectral data repositories. Moreover, we plan to expose the same funtionalities to a script language, in order to record and apply the work on larger sets of images. We also plan to better exploit multiple processors, when available on current hardware platforms.

Our efforts were devoted so far to the development and test of the visualization and processing modules, rather than a structured approach to the GUI design and evaluation. This is a further direction to be followed in order to increase the usefulness of the project for real-world applications.

Theia cannot be compared with well-established software products for multispectral image analysis, and a considerable amount of work is needed to reach an adequate maturity. Rather, after significant development and systematic tests, Theia is likely to remain an open platform for tailoring and testing specialized applications.

5. REFERENCES

[1] ENVI ITT Visual Information Solutions http://www.ittvis.com/envi/index.asp

[2] MultiSpec Purdue Research Foundation http://cobweb.ecn.purdue.edu/~biehl/MultiSpec/

[3] QT4 TrollTech http://trolltech.com/products/qt

[4] LibTIFF http://www.libtiff.org/

[5] Perrizo, W., Ding, Q., Ding, Q., and Roy, A. On Mining Satellite and Other Remotely Sensed Images Data Mining and Knowledge Discovery (2001), pp.33-40 http://www.cs.ndsu.nodak.edu/~ding/publications/DMKD01.pdf

[6] Adobe Systems Incorporated TIFF Revision 6.0 Tagged Image File Format 1992 http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf; accessed December 15, 2007

The Multi-Touch SoundScape Renderer

Katharina Bredies, Nick Alexander Mann, Jens Ahrens,
Matthias Geier, Sascha Spors and Michael Nischt
Deutsche Telekom Laboratories / Technische Universität Berlin
Ernst-Reuter-Platz 7, 18. Floor
10587 Berlin, Germany
katharina.bredies@telekom.de

ABSTRACT

In this paper, we introduce a direct manipulation tabletop multi-touch user interface for spatial audio scenes. Although spatial audio rendering exists for several decades now, mass market applications have not been developed and the user interfaces still address a small group of expert users. We implemented an easy-to-use direct manipulation interface for multiple users, taking full advantage of the object-based audio rendering mode. Two versions of the user interface have been developed to explore variations in information architecture and will be evaluated in user tests.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Graphical User Interfaces; H.5.5 [**Sound and Music Computing**]: Systems

1. INTRODUCTION

More and more spatial sound reproduction systems are used in practice. They differ in the particular reproduction technique used and in the speaker layout. Traditionally, audio is produced and transmitted on a per reproduction channel basis and hence for a particular speaker layout. In order to handle the variety of sound reproduction systems, object-based audio is a promising approach. In this case, the audio objects are transmitted together with side information; the local terminal then renders the sound for the local speaker layout using a suitable spatialization algorithm.

This object based approach to spatial audio opens up the freedom of local user interaction with the audio scene. As sounds can be handled as discrete objects, it also offers new possibilities for audio editing tools and interaction techniques. However, currently spatial audio tools mostly address expert users. Solving technical problems has been given priority over developing appropriate digital interface designs until now. Current tools often stick to well-known visual metaphors of physical devices and reproduce them on screen. Such interface solutions are normally used before an appropriate standard for new interaction modes is established. However, they don't exploit the extended options that object-oriented sound reproduction offers. Furthermore, as soon as the technology will reach a mass market in telecommunications or entertainment, there will be a need for more intuitive interfaces.

The development of the SoundScape Renderer user interface was an opportunity to explore new interaction techniques for object-based spatial sound scenes, enabling direct manipulation and collaborative interaction. We built a Frustrated Total Internal Reflection (FTIR) based multi-touch table similar to the one presented in [8]. Building on an existing mouse-based previous iteration of the interface, we implemented two versions for the table: One that focused on so-called "gestures" and multi-touch input, heavily relying on the idea of direct manipulation. The other version emphasizes the collaborative aspect and provides individual menus for each user, assembling the available information and functions in one area. In the following, we describe the hardware setup, the development of the graphical user interface for the spatial sound renderer software and the two versions of the interface that we designed for user testing.

2. RELATED WORK

A lot of interactive tabletop devices for musical expression have been developed as music controllers in the past years, both as multi-touch and tangible devices. Popular examples are the reactable[10] and the audiopad[12], both of which use physical pucks as interactive objects. Examples for multi-touch interfaces are the LEMUR sensor pad [9] and the synthesis interface for the original FTIR table presented in [4]. They provide an overview of audio-related visualizations, gestures and interaction techniques that range between physical and virtual space. On the other hand, spatial mappings have been exploited in domains where they are inherent, like landscape or interior architectural planning scenarios[2, 14]. This approach is also very obvious for controlling spatial sound sources. However, only few research has been dedicated to multi-touch tabletop devices for spatial sound scenes.

The ISS Cube [13] uses a surround sound system to create spatial audio scenes. The sources are represented as acrylic pucks that can be put into the scene space and be moved around freely. In the Audiocube [1] system, the sounds are represented as cubes. Each side of the cube plays a different sample once it is put down on the table. Both systems are developed for exhibitions and offer only reduced functionality like spatial positioning, volume and sample changes. Also, the scenes are spatially restricted and do not support

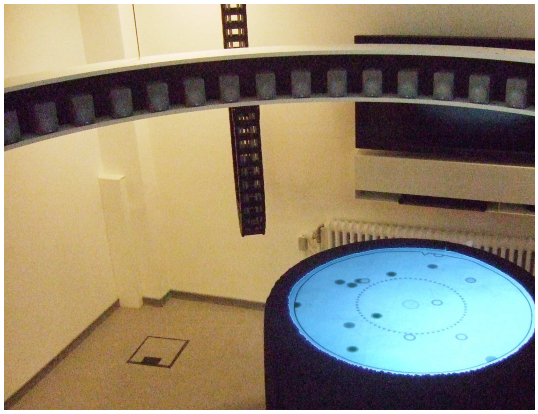


Figure 1: The wave field synthesis lab room with the circular speaker array and the multi-touch table in the center.

animation of sound sources.
The IOSONO system [6], on the other hand, uses a Wacom pen tablet and virtual representations to work on spatial sound scenes. Developed at Fraunhofer Institute for Digital Media Technology (IDMT), it adresses expert users and refers to established audio software interfaces. Furthermore, there seems to be no multi-touch functionality implemented on the tablet, although the mapping is similar to a tabletop device.
For our purpose, the tangible tabletop interfaces provide a very comprehensive mapping, but do not offer the required flexibility. Real-time audio applications are a good source for existing and well-working multi-touch and "gesture" input possibilities. From other applications with spatial mappings, we could also learn about the particularities and problems with spatial layouts and refer to an established repertoire of multi-touch inputs.

3. HARDWARE SETUP

We used the Frustrated Total Internal Reflection technique introduced to multi-touch sensing in [8] to built a custom-made, round-shaped tabletop multi-touch device. The FTIR technology seemed appropriate because of its relatively low costs and nonetheless sufficient performance for multi-touch purposes. We decided in favor of a round shape to avoid that the table would have a particular orientation, to fit it into the circular wave field synthesis speaker system, and to allow for an undetermined number of users (see Fig.1). For the spatial mapping, the horizontal orientation of the display seemed very appropriate. It would also cause less fatigue during use.
In contrast to many other audio tabletop applications, we decided against using tangible parts like markers. Without tangible components, we preserved more liberty and reduced the danger of accidentally changing the setup. We also had no restrictions regarding the amount and placing or intersection of the sources. Besides, we could include animation in our interface.
The deployed software dealing with concurrent touch inputs was developed in house as an open source library built upon the $Java^{TM}$ Platform. Besides a small native part for the camera driver grabbing the input image sequence, the software is organized into two independent modules. The first one is responsible for determining and describing the touch inputs. This is done by labeling connected components in a gray-scale image, whereby each of them corresponds to a single touched area. The other module focuses more on interaction. It contains a set of composable manipulators to be linked with the user interface. For example, it allows to move, scale and rotate a single widget with an unlimited amount of touches contributing. However, in this project it was not truly used, because of the $Flex^{TM}$ based user interface, which receives touch states accordingly to the TUIO Protocol [11]. Although we knew that it is not perfectly suitable for a multi-touch table as its focus lies on rigid physical object markers, we welcomed the simplicity and it proved good enough for our purposes.

4. SOUNDSCAPE RENDERER

The SoundScape Renderer (SSR) is a software framework for real-time rendering of spatial audio using a variety of algorithms [5]. At its current development stage it is able to render virtual acoustic scenes using either Wave Field Synthesis (WFS), binaural rendering or Vector Based Amplitude Panning (VBAP). It provides, amongst other key features, a graphical user interface and a network interface to interact in real-time with the auditory scene. Multiple clients can connect to a central SSR and modify the scene and system parameters. Thereby any type of interface or tracking system can be connected easily and control the SSR. The multi-touch GUI was implemented as a Flash Shockwave application that includes all the necessary functionalities to connect to a SSR and interact in real-time with the auditory scene.

5. SSR GUI DEVELOPMENT

For the binaural rendering mode of the SSR, we had designed a graphic user interface for demonstration purposes. So we already had a set of requirements for the user interface and informal experience concerning its usability.
As an initial step, we conducted a short informal video inquiry, asking people to perform some crucial functions that we wanted to implement while recording their movements. It turned out that they stick closely to the interaction style with mouse and keyboard they were already familiar with. But we also recognized some patterns of how a couple of basic actions were performed (like rotating and scaling). Those were, on the other hand, geared to the interaction with physical objects, or simply known from other touch-based devices. However, the test subjects reported dissatisfaction about their own creative output and ascribed it to not being prepared and not having the time to reconsider their expressions. We therefore took the video inquiry as an inspiration rather than as a strict requirement.
We then decided upon correspondent multi-touch input for the SSR interface. One particular problem was to find ways to display information in the scene and on the sources. As GUIs for spatial audio are not widespread, we had few conventions to stick to. So we borrowed some interactions from existing audio editors as well as from tangible audio interfaces like home stereos. Besides, we referred to the various examples of musical tabletop devices[1, 10, 9, 12] and some multi-touch literature [14, 3].
We also conducted paper prototyping with the first designs

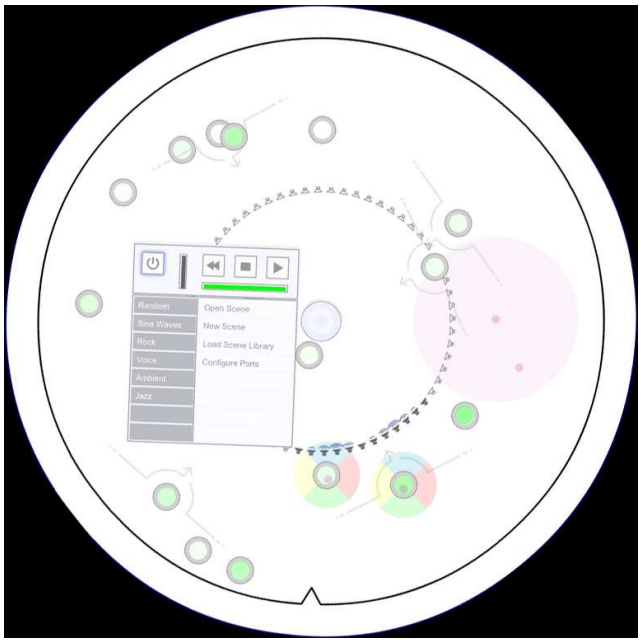


Figure 2: Version A of the SSR multi-touch GUI. The center application menu, the scene translation halo and the source context shortcuts are visible.

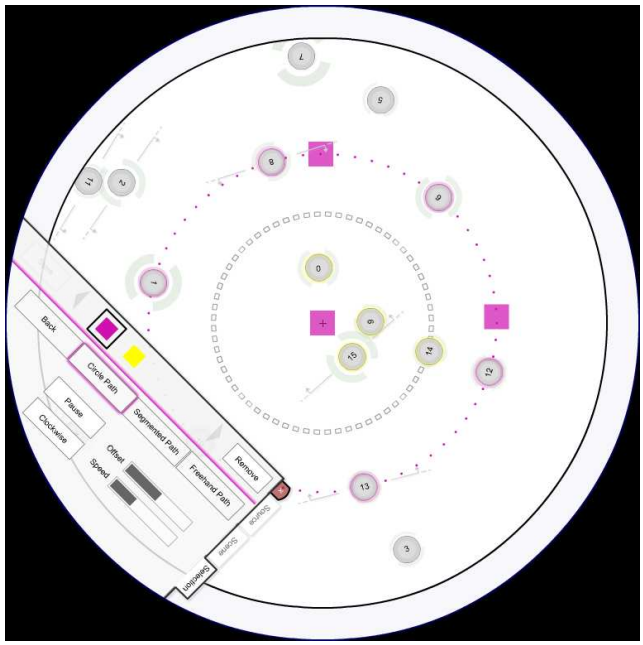


Figure 3: Version B with the edge menu and some sources animated to follow a circle path.

to find inconsistencies and logical gaps in an early stage. Here, we checked the size of the graphic representations especially important for touch input. According to the paper prototyping results, we implemented the graphic user interface for the table.

6. RESULTS

The preliminary result of our development are two versions of the multi-touch SSR user interface. They apply two different extremes of interface logic that we came upon during the information architecture design and sketching phase. The first one (that we will refer to as version A), inherits the organization of the predecessor mouse-based interface for the SSR and enhances it with multi-touch input. The second one (called version B) focuses on making the interface appropriate for multiple users. Both versions provide sound sources as single objects, a surrounding space that we named the ”scene space”, and a menu with application-related functions.
In the following, both versions will be described in detail.

6.1 Version A

Version A organizes the features directly in situ. All functionality and information belonging to a particular source is attached to it. For this version, we used buttons as well as simple finger movements that we refer to as ”gestures”. The term ”gesture” in a multi-touch context is not related to the expressive gestures used in human face-to-face communication. It is used to describe familiar and conventional hand movements for some particular task: e.g., turning the hand with some fingers pressed on the surface can be interpreted as a ”gesture” for rotation. Those movements partially derive from interaction with physical objects. A common set has already been patented (see [7]).

Each sound source has its own four-part shortcut menu that is arranged around the core. It consists of three toggles (to mute and solo and to display the information panel) and a gesture slider for the volume. The user can simply tap the toggle shortcuts to perform the allocated action. To change the volume, he has to put one finger in the shortcut area and perform a scaling gesture (see Fig.2). The volume level is then indicated as a green arc around the source, increasing and decreasing with the volume. The intensity of the sound each source emits is represented as a green circle in the source center that gets brighter and darker. An arrow at the rim of the table indicates the reading direction and justification for the shortcut menu. Touching the rim moves the arrow to the respective place and readjusts all source menus.
If the user touches the scene space, a pink halo appears around his touch. He can scale and rotate the whole scene by placing another finger in this area and moving it relative to the first one. Moving the first finger will move the whole scene. The translations from several users are added up against each other.
Pressing the center of the table a menu appears with features for the whole application. The user can jump to another preset audio scene, adjust the volume, pause and play the audio rendering.
In this version, the most important functions are accessible through gestures. We tried to make the interface as ”direct” as possible, so the features would be easy to find. Anyway, one clear drawback of gesture input is that it is not made visible in the interface; there is no visual hint on what kind of gestures could be performed. So we stuck to a very small set of gestures that we assumed to be the most common, relying on our short video inquiry.

6.2 Version B

In version B, interaction with the sound sources in the

scene space was reduced to positioning, selecting, deselecting and grouping. Tapping on several sources one after another would store them in a preliminary group selection that was always highlighted in blue. Using the rim menu, the groups could then be stored permanently. Tapping into the void would deselect all sources. The user can also drag lassos around several sources to select them. We kept two gesture shortcuts: Drawing a lasso around one pressed-down finger would select all sources; drawing a lasso around two pressed fingers selects all stored groups. The emitting sound of a source is visualized by pulsating transparent arcs around it that get bigger the louder the source sound is. The reading direction for the type is always oriented towards the edges, so the label of a source changes direction when the source is moved over the table.

The whole functionality - source information and type, scene features, application features and group features - was relocated in a rim panel. Several panels could open up in front of each user (see Fig.3). The panel contains three tabs: The first one is showing the source menu, the second one the preset scene selection and scene features, and the third one grouping and animation features.

The source menu displays all information and functions allocated to the source. The user can pick the source in numerical or alphabetical order from a top menu bar, then change the volume and source type on the panel. The selected source is highlighted with a pulsating blue halo in the scene space. However, it is not possible to allocate a source selected in the scene space to a particular panel.

The scene menu tab shows a number of preset scenes as well as the scene master volume, scale and source file.

In the third tab, the user can store his selection in permanent groups and apply some animation behavior, like following a circle path or moving between a number of points. Again, particular groups can only be selected in the menu.

In this version, the connection between the individual panel and a single source is much weaker than in version A, but it offers more detailed interaction, like assembling, storing, changing and animating groups.

7. CONCLUSIONS

In this paper, we described the development process of a multi-touch user interface for a spatial sound reproduction software. We produced two versions of the GUI, working with different visualizations and with different foci. We believe that these two versions address a general problematic of interfaces with spatial layouts and that the results can be applied to different domains of simulation and planning. The ongoing work will therefore aim at further refining the interface, especially assimilating the functionality of both versions while purifying their different approaches. Additionally, we are interested in exploring the usability of both approaches with regard to the information architecture.

Although one can assume the strengths and weaknesses of each version, they have not been evaluated in user testings so far. We will therefore conduct formal user tests to evaluate the actual advantages and drawbacks of each version. Besides, we use the table as a testbed not only to reveal usability requirements for audio software tools, but to observe emergent user behavior for new application areas.

8. REFERENCES

[1] Audite. Audiocube. `http://www.audite.at/en/projects_audiocube.html`.

[2] E. Ben-Joseph, H. Ishii, J. Underkoffler, B. Piper, and L. Yeung. Urban simulation and the luminous planning table bridging the gap between the digital and the tangible.

[3] H. Benko, A. D. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272, New York, NY, USA, 2006. ACM.

[4] P. L. Davidson and J. Y. Han. Synthesis and control on large scale multi-touch sensing displays. In *NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression*, pages 216–219, Paris, France, France, 2006. IRCAM — Centre Pompidou.

[5] M. Geier, J. Ahrens, A. Möhl, S. Spors, J. Loh, and K. Bredies. The soundscape renderer: A versatile framework for spatial audio reproduction. In *Proceedings of the DEGA WFS Symposium*, Ilmenau, Germany, Sept. 2007. Deutsche Gesellschaft für Akustik (DEGA).

[6] I. GmbH. Iosono. `http://www.iosono-sound.com/`.

[7] J. G. Greer, W. Westerman, and M. Haggerty. Multi-touch gesture dictionary, January 2007.

[8] J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, New York, NY, USA, 2005. ACM.

[9] Jazzmutant. Lemur. `www.jazzmutant.com`, 2004.

[10] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, New York, NY, USA, 2007. ACM.

[11] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza. Tuio - a protocol for table based tangible user interfaces. In *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France, 2005.

[12] J. Patten, B. Recht, and H. Ishii. Interaction techniques for musical performance with tabletop tangible interfaces. In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, page 27, New York, NY, USA, 2006. ACM.

[13] Something. ISS cube. `http://www.somethingonline.org/index.php?main=done&sub=iss_cube`.

[14] M. Wu and R. Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 193–202, New York, NY, USA, 2003. ACM.

Demos Session

Interactive Visual Interfaces for Evacuation Planning

Gennady Andrienko Natalia Andrienko Ulrich Bartling

Fraunhofer Institute IAIS (Intelligent Analysis and Information Systems)
Schloss Birlinghoven; 53754 Sankt Augustin, Germany, +49 2241 142486

{gennady.andrienko, natalia.andrienko, ulrich.bartling}@iais.fraunhofer.de

ABSTRACT

To support planning of massive transportations under time-critical conditions, in particular, evacuation of people from a disaster-affected area, we have developed a software module for automated generation of transportation schedules and a suite of visual analytics tools that enable the verification of a schedule by a human expert. We combine computational, visual, and interactive techniques to help the user to deal with large and complex data involving geographical space, time, and heterogeneous objects.

Categories and Subject Descriptors

H.1.2 [User/Machine Systems]: Human information processing – Visual Analytics; I.6.9 [Visualization]: information visualization.

Keywords

Visual Analytics, geovisualization, transportation planning, task-centered visualization design, coordinated multiple views.

1. INTRODUCTION

In time critical situations, software tools automating some of people's activities or suggesting solutions to problems are of great benefit. However, machine-generated solutions can generally be used only after a verification and validation by a human expert, who takes the responsibility for the decisions made. Hence, the expert needs tools that enable effective reviewing of these solutions in the shortest possible time. Although visualization plays a great role here, large amounts of information cannot be efficiently examined without the involvement of computational techniques for analysis and summarization.

We have developed a software system to support civil protection services in planning evacuation of people from disaster-affected areas. The system includes a module that automatically builds transportation schedules and a suite of techniques enabling the inspection of the schedules by a human planner. To handle large amounts of data, we integrate interactive visual displays with computational techniques for data transformation, according to the paradigm of visual analytics (Thomas and Cook 2005, Keim 2005). This distinguishes our approach from the usual tools (e.g. ILOG 2007, TurboRouter 2007, Fagerholt 2004).

AVI'08, 28-30 May , 2008, Napoli, Italy

In (Andrienko et al. 2007), we described the main features of the automated schedule builder and presented our task-centered design of the tools for schedule examination. We also demonstrated the appropriateness of the tools for the task by an example of schedule analysis. In this presentation, we focus on the display manipulation techniques, coordination between different views, and dynamic transformations of the data.

2. VISUAL ANALYTICS TOOLS

2.1 The data to be examined

In an emergency evacuation, it is necessary to schedule the transportation of many people from multiple sources (original locations) to multiple destinations (shelters). There may be diverse categories of people such as general public, disabled people, and critically sick or injured persons. These categories need to be handled differently, which includes the selection of proper destinations and proper types of vehicles as well as proper timing of the transportation.

The input data for the evacuation planning include (1) the sources of the endangered people, (2) the numbers and categories of these people, (3) the latest allowed departure times per place and category; (4) possible destinations and their capacities, by people categories; (5) types of vehicles and their capacities for the people categories they are suitable for; (6) available vehicles and their initial locations. The automated schedule generator produces a collection of *transportation orders* assigned to the vehicles, where each order specifies one trip of a vehicle: source and destination locations, start and end times, and the category and number of the people to be delivered. One schedule may consist of hundreds of orders. A human planner cannot examine each order individually, especially under time-critical conditions. Hence, the information needs to be presented to the planner in a summarized form adequate to the purpose of detecting possible problems (e.g. people remaining in the sources, time limits exceeded, etc.) and understanding their reasons.

2.2 Dynamic aggregation

To provide a summarized representation of the data while enabling the planner to focus on various subsets, we combine interactive filtering of the data with dynamic aggregation. The user may set one or more data filters of different types: by people category, by time interval, by source, and/or by destination. The aggregation is applied to the portion of the data that have passed through the filters and immediately re-applied when the filters change. For this purpose, several types of *dynamic aggregators* are created. A dynamic aggregator is a special object linked to a number of data records and able to derive certain statistical summaries from those records which satisfy current filters. These summaries are presented on visual displays, and the aggregators

are responsible for updating the displays when the filters change. Different types of aggregators are attached to individual locations (e.g. counters of remaining people in the source locations and counters of used and free capacities in the destinations), to pairs of locations (trip aggregators), or to the entire territory (e.g. aggregator of people by states and calculator of the vehicle use).

2.3 Visualization and user interaction

A transportation schedule is a complex construct involving geographical space, time, and heterogeneous objects (people and vehicles) with states and positions varying in time. All this information cannot be appropriately presented in a single display. Our toolkit includes several coordinated views presenting different aspects: (1) a summary view of the transportation progress over time (Figure 1), which also serves as a direct manipulation interface to the time filter; (2) a map display showing the situation on a user-selected time interval (Figures 2, 3); (3) a source-destination matrix presenting summarized data for pairs of locations, which serves as a direct manipulation interface of the filter by source and/or destination; (4) a Gantt chart providing a detailed view of the distribution of the trips over time. All the views are dynamically updated when the user changes current filters: selects an item category, a time interval, a source, and/or a destination. In our presentation, we are going to demonstrate how the tools enable detection of possible problems and investigation into their reasons.

3. CONCLUSION

To support efficient examination of large transportation schedules involving multiple geographical locations and diverse categories of transported items and types of vehicles, we combine interactive visual displays with dynamic aggregation and summarization of the data. This research is conducted within the integrated EU-funded project OASIS – Open Advanced System for Improved Crisis Management (IST-2003-004677, 2004-2008; http://www.oasis-fp6.org/). We have presented out tools to potential users, professionals in civil protection and crisis management, who expressed their high interest and wish to have such tools at their service. Next year, the users will test and evaluate the tools in the course of the trials of the entire OASIS system, which will take place in two European countries.

4. REFERENCES

[1] Andrienko, G., Andrienko, N., Bartling, U. 2007. Visual Analytics Approach to User-Controlled Evacuation Scheduling, In Proceedings of the IEEE VAST 2007 Conference (Sacramento, CA, USA, Oct.2007), 43-50

[2] Fagerholt, K.2004. A computer-based decision support system for vessel fleet scheduling - experience and future research, Decision Support Systems, 37(1), 35– 47

[3] ILOG 2007. ILOG Transport PowerOps, available at URL: http://www.ilog.com/products/transportpowerops/, last accessed: 20 March 2007

[4] Keim, D. A. 2005. Scaling visual analytics to very large data sets, presentation at Visual Analytics Workshop, June 4th, 2005, Darmstadt, Germany, http://infovis.uni-konstanz.de/index.php?region=events&event=VisAnalyticsWs05

[5] Thomas, J.J. and Cook, K.A. (editors) 2005. Illuminating the Path. The Research and development Agenda for Visual Analytics, IEEE Computer Society

[6] TurboRouter 2007. TurboRouter software. Schedule visualization, available at URL: http://www.marintek.sintef.no/TurboRouter/visualisations.htm, last accessed: 20 March 2007

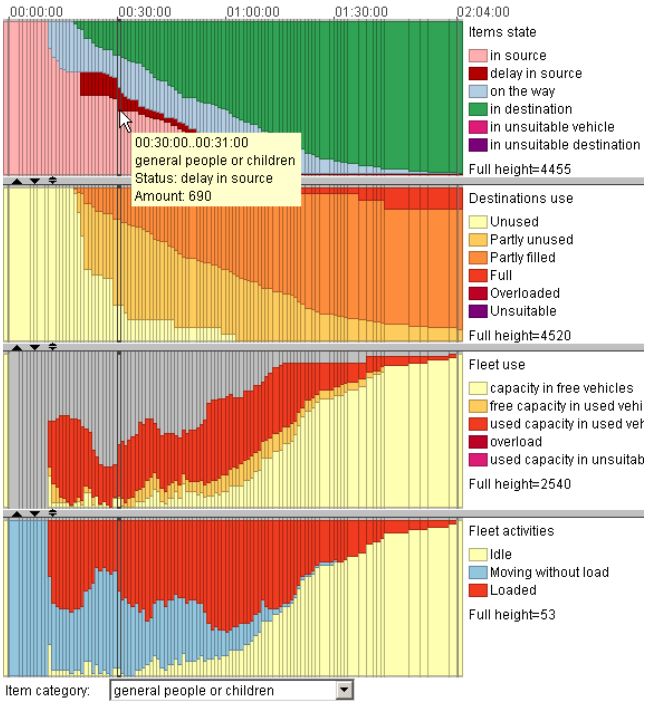


Figure 1. The summary view of the transportation progress.

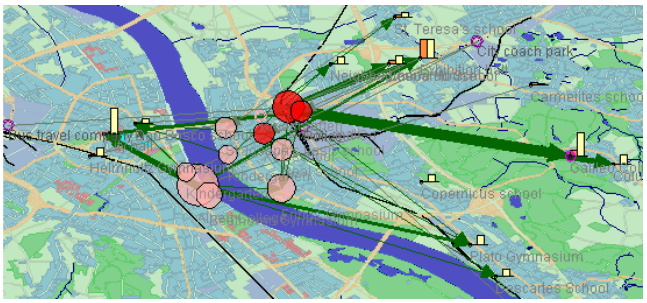


Figure 2. A fragment of the map view.

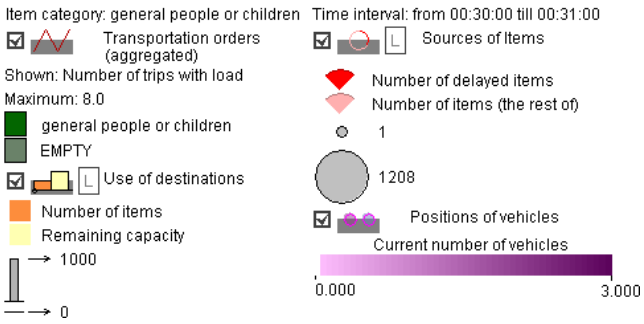


Figure 3. The legend of the map shown in Figure 2.

Supporting Visual Exploration of Massive Movement Data

Natalia Andrienko Gennady Andrienko

Fraunhofer Institute IAIS (Intelligent Analysis and Information Systems)
Schloss Birlinghoven; 53754 Sankt Augustin, Germany, +49 2241 142486

{natalia.andrienko, gennady.andrienko}@iais.fraunhofer.de

ABSTRACT

To make sense from large amounts of movement data (sequences of positions of moving objects), a human analyst needs interactive visual displays enhanced with database operations and methods of computational analysis. We present a toolkit for analysis of movement data that enables a synergistic use of the three types of techniques.

Categories and Subject Descriptors

H.1.2 [User/Machine Systems]: Human information processing – Visual Analytics; I.6.9 [Visualization]: information visualization.

Keywords

Movement data, trajectory, movement patterns, movement behavior, visual analytics, exploratory data analysis, visualization, interactive displays, cluster analysis, aggregation.

1. INTRODUCTION

Thanks to the recent advent of inexpensive positioning technologies, data about movement of various mobile objects are collected in rapidly growing amounts. Potentially, these data are a source of valuable knowledge about behavioral and mobility patterns. To gain an understanding of these patterns, an analyst needs a visual representation of the data, which is the most effective way to support human perception, cognition, and reasoning. However, purely visual methods of analysis (e.g. Hägerstrand 1970), even being enhanced with interactive techniques (Andrienko et al. 2000, Kraak 2003, Kapler and Wright 2005), are not scalable to large datasets. Such methods need to be combined with database operations and computational analysis techniques helping to handle large amounts of data. Some approaches have been suggested recently. Forer and Huisman (2000) and Dykes and Mountain (2003) summarize movement data into surfaces, but this is not suitable for analyzing routes. Buliung and Kanaroglou (2004) envelop bunches of trajectories and compute the central tendency, which works well for similar and close trajectories. Laube et al. (2000) combine visualization with data mining methods oriented to specific types of patterns.

In (Andrienko et al. 2007) we have presented a framework and a toolkit for analysis of movement data based on a synergy of visualization, database operations and computations. Here, we focus on the visual and interactive components of the toolkit.

AVI'08, May 28-30, 2008, Napoli, Italy.

2. MAKING SENSE FROM POSITION SEQUENCES

Movement data acquired by position tracking usually lack any semantics. The records basically consist of time stamps and coordinates. In particular, there are no explicitly defined trips with specified origins and destinations and no semantically identifiable places. To understand the data, an analyst should be able to link them to his/her prior knowledge and interpretable information from other sources. Visualization is essential for this purpose.

2.1 Finding significant places

One important task in analysis of movement data is to extract and interpret the places of stops. Our toolkit supports this task in the following way. First, the positions of stops with user-specified minimum duration are extracted from the database. Second, a spatial clustering tool is applied to find groups of spatially close positions, which indicate repeatedly visited places. Third, the results are shown on a map where the positions are marked by colored point symbols (each cluster receives a unique color). The map provides the spatial context and thereby helps the analyst to interpret the places. Additional help may come from temporal histograms showing the distribution of the stops within temporal cycles (daily, weekly, etc.). Thus, the histograms in Figure 1 show the frequencies of stops of a personal car for minimum 3 hours by days of the week (A) and by hours of the day (B). The colored bar segments represent the results of the spatial clustering of the stops. It is vividly seen that the stops of the "blue" cluster occur only on the working days and mostly in the morning times. The stops of the "red" cluster occur all days and mostly in the evenings. A plausible conclusion is that the "blue" cluster of positions is situated near the place where the person works and the "red" cluster is near person's home.

2.2 Extracting trips and exploring the routes

The sequence of position records representing the movement of an entity needs to be partitioned into subsequences corresponding to trips. The notion of trip may be application- and goal-dependent. Our toolkit allows the users to divide data in several ways: by stops, by spatial gaps, by temporal cycles, and by places of interest. The division is done by means of database operations. After that, repeated trips and typical routes may be detected by means of clustering, which groups together trips having something in common, depending on the distance function chosen (e.g. closeness of the origins and destinations, similarity of the routes). The analyst may select one or a few clusters and refine

them by re-applying the clustering tool with a different distance function or different parameter settings. To represent massive movements on a map display, we apply computational summarization of moves. The algorithm is described in (Andrienko et al. 2007). The results look as shown in Figure 2.

2.3 Exploring movement dynamics

3D views where two dimensions represent space and one represents time (Hägerstrand 1970) are good for exploring the speed of movement and its variation over time. This approach can be used even for multiple trajectories if they do not intersect (in particular, if they follow the same route). One of the distance functions in our toolkit groups trajectories by similarity of the routes and similar dynamics of the movement. Selected clusters can be explored and compared in a 3D view as shown in Figure 3, which is quite legible despite the number of trajectories displayed.

3. CONCLUSION

Interactive visual displays play the key role in supporting sense-making from movement data but are insufficient when the data are large. Our framework combines visualization with database operations and computations. The generic database techniques enable handling large datasets and are used for basic data processing and extraction of relevant objects and features. The computational techniques, which are specially devised for movement data, aggregate and summarize these objects and features and thereby enable the visualization of large amounts of information. The visualization enables human cognition and reasoning, which, in turn, direct and control the further analysis by means of the database, computational, and visual techniques.

The reported work has been partly funded by EU in the project GeoPKDD - Geographic Privacy-aware Knowledge Discovery and Delivery (IST-6FP-014915; http://www.geopkdd.eu).

4. REFERENCES

[1] Andrienko, G., Andrienko, N., Wrobel, S. 2007. Visual Analytics Tools for Analysis of Movement Data. ACM SIGKDD Explorations, 9 (2), (in press)

[2] Andrienko, N., Andrienko, G., Gatalsky, P. 2000. Supporting Visual Exploration of Object Movement. In Proc. Working Conf. Advanced Visual Interfaces AVI 2000 (Palermo, Italy, May 2000), ACM Press, 217-220, 315

[3] Buliung, R.N., Kanaroglou, P.S. 2004. An Exploratory Data Analysis (ESDA) toolkit for the analysis of activity/travel data. Proceedings of ICCSA 2004, LNCS 3044, Springer, Berlin, 1016-1025

[4] Dykes, J. A., Mountain, D. M. 2003. Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications, Computational Statistics and Data Analysis, 43, 581-603

[5] Forer, P., Huisman, O. 2000. Space, Time and Sequencing: Substitution at the Physical/Virtual Interface. In Information, Place and Cyberspace: Issues in Accessibility (Eds: Janelle, D.G., Hodge, D.C.), Springer, Berlin, 73-90

[6] Hägerstrand, T. 1970. What about people in regional science? Papers of the Regional Science Association, 24, 7-21

[7] Kapler, T., Wright, W. 2005. GeoTime information visualization, Information Visualization, 4(2), 136-146

[8] Kraak, M.-J. 2003. The space-time cube revisited from a geovisualization perspective. In Proc. 21st Int. Cartographic Conf. (Durban, South Africa, Aug. 2003), 1988-1995

[9] Laube, P., Imfeld, S., Weibel, R. 2005. Discovering relative motion patterns in groups of moving point objects. Int. J. Geographical Information Science, 19(6), 639–668

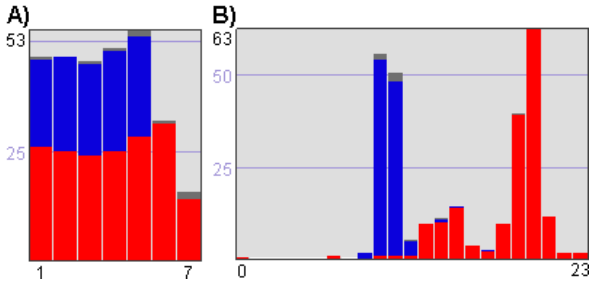


Figure 1. Temporal histograms show results of clustering of stop positions.

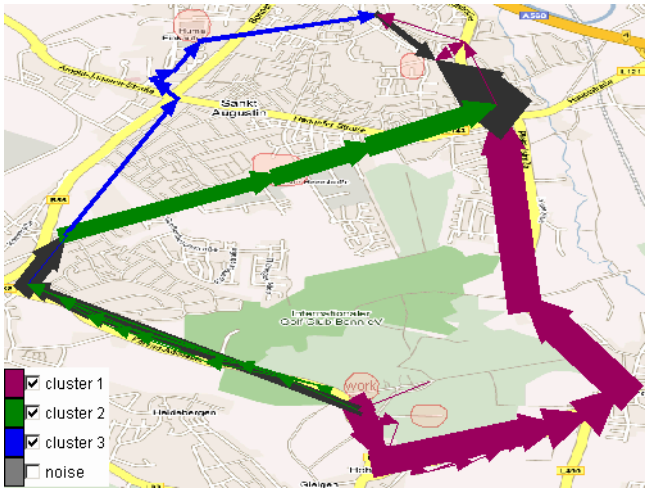


Figure 2. Three clusters of trips are represented in a summarized form.

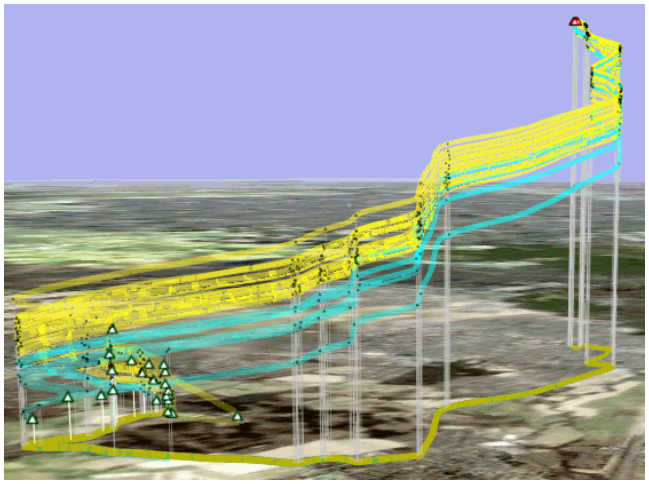


Figure 3. Two clusters of trips follow the same route but differ in the dynamics.

Scenique: A Multimodal Image Retrieval Interface*

Ilaria Bartolini, Paolo Ciaccia
DEIS
University of Bologna, Italy
{i.bartolini, paolo.ciaccia}@unibo.it

ABSTRACT

Searching for images by using low-level visual features, such as color and texture, is known to be a powerful, yet imprecise, retrieval paradigm. The same is true if search relies only on keywords (or *tags*), either derived from the image context or user-provided annotations. In this demo we present Scenique, a multimodal image retrieval system that provides the user with two basic facilities: 1) an *image annotator*, that is able to predict keywords for new (i.e., unlabelled) images, and 2) an integrated query facility that allows the user to search for images using *both* visual features and tags, possibly organized in semantic *dimensions*. We demonstrate the accuracy of image annotation and the improved precision that Scenique obtains with respect to querying with either only features or keywords.

Keywords

Multi-structural Databases, Semantic Dimensions, Visual Features.

1. INTRODUCTION

The advent of digital photography has enormously increased the demand of tools for effectively managing huge amounts of color images. Among such tools, those providing similarity-search functionalities are essential if one wants to provide users with the possibility of looking for images whose visual content is similar to a given, so-called *query*, image. Even if this content-based approach can be completely automatized, it is known to yield imprecise results because of the semantic gap existing between the user subjective notion of similarity and the one implemented by the system.

The alternative to content-based retrieval is to look for images by using text-based techniques. Towards this end, several solutions have been proposed in recent years, such as the image search extensions of Google[1] and Yahoo[2], which consider the original Web context (e.g., file name, title, surrounding text) to infer the relevance of an image, as well as systems like flickr[3], which rely on user-provided *tags*. However, in both cases, the accuracy of the results is highly variable, since it heavily depends on the precision and the completeness of the manual annotation process (in the case of flickr) and it is completely uncorrelated with respect to the visual image content (in the case of Google and Yahoo).

In this demonstration we present Scenique (Semantic and ContENt-based Image QUErying), a multimodal image retrieval system whose major aim is to provide users with an *integrated* query facility that allows images to be searched by means of *both* visual features and semantic tags, thus taking the best of the two approaches. The model of Scenique is based on the multi-structural framework proposed in [2]. In particular, each image is viewed as a set of regions, from which color and texture can be automatically extracted, and a set of tags. Tags can be organized in so-called (classification) *dimensions*, which take the form of *tag trees*. Each dimension, such as `location`, is thought to be as a particular coordinate to describe the content of an image. When dimensions are defined by the user, Scenique predicts tags for each specified dimension.

Searching for images in Scenique can take three basic forms, as better explained in the following: content-based only, tag-based only, and integrated. In the demo we will show how the quality of retrieval depends on the chosen query modality.

2. ARCHITECTURE AND PRINCIPLES

The Scenique architecture is mainly composed by a *Feature DB* storing color and texture feature vectors that are automatically extracted from images, and by a *Tag DB* which stores the current tags defined for each image. A tag occurrence is actually a specific node in a *tag tree*, each tag tree representing the organization of tags for a specific *dimension*. As an example, the tag `animal` could be a node in the tag tree of the `subject` dimension. Note that, in principle, the same tag can appear in different tag trees, which allows to discriminate between the different usages and/or meanings different tag occurrences can have. For instance, the tag `Italy` might appear as a node for the `location` dimension (used to organize photos according to the place they have been shot) as well as a node in the `sport` dimension (which only applies to photos related to sport events).

By default, each tag is initially a node of a generic, unstructured, `default` dimension. User-defined dimensions

*This work is partially supported by a Telecom Italia grant.
[1]Google image: http://images.google.com/
[2]Yahoo image: http://images.search.yahoo.com/
[3]flickr: http://www.flickr.com/

AVI '08, May 28-30, 2008, Napoli, Italy

can be defined to fit specific needs. For instance, in order to organize photos according to their main `subject`, a corresponding dimension can be defined and structured by creating the nodes `person` and `animal`; then the node `animal` can be split into the three nodes `mammal`, `bird`, and `fish`. The node `mammal` can be further specialized into nodes `bear`, `horse`, etc.

Scenique is based on the multi-structural framework [2], that consists of a set of objects, together with a *schema* that specifies a classification of the objects according to multiple distinct criteria (i.e., the dimensions). In such a way, the user can define several dimensions, with the aim to organize images from different points of view, and, at query time, browse images through the tag trees as well as formulate composed tag-based queries. This is exemplified in Figure 1, where the dimensions `subject` and `place` are conjunctly used to look for "sea animal" images. Within the

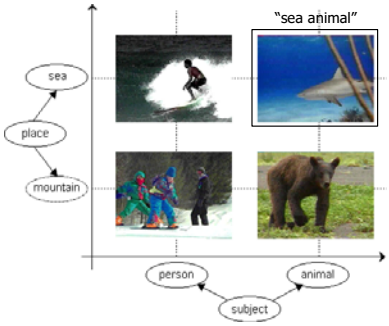


Figure 1: A compound search based on the `place` and `subject` dimensions.

reference model, a set of operations, such as the *meet* (or logical `AND`) and the *join* (or logical `OR`) are defined. In this way, the user formulates compound queries by means of logical expressions (e.g., `(sea AND animal)`).

Queries submitted by the user are managed by the *Query Processor*, which supports three main query modalities: *content*-based (C), *tag*-based (T), and *content&tag*-based (CT), respectively. With modality C, the user is looking for images that are similar, from a visual features point of view, to a specific query image. In particular, the Query Processor provides support for k nearest neighbor (k-NN) queries: Given a query image, it ranks images according to a specific similarity criterion and returns the k images with highest similarity score. Queries of type T are formulated using the available dimensions. In the simplest case, the retrieval is based on the resolution of user-provided logical expressions, that relies upon the exact match between selected tags and image associated tags. More interesting queries are derived when the *parent-child* relations between nodes of the tag trees (e.g., "the bear is a mammal") are exploiting by the Query Processor to improve the quality of the results. By supposing that the user is looking for `bear` images, the result provided by the Query Processor might include not only images with the tag `bear`, but also images annotated with the tag `asiatic_brown_bear`, because, in this case, the Query Processor takes the advantage of the relation "the asiatic_brown_bear is a bear". With the same aim, lexical ontologies, such as WordNet[4], can be exploited instead of user-defined dimensions. This allows to deal with the case when provided keywords do not belong to any dimension. Finally, with *content&tag*-based queries, the Query Processor combines C and T modalities by returning images in the intersection of both the C and T results first, followed by images in the T list only and, finally, by images in the C result only.

The user can also take advantage of the *Annotator* component of Scenique to obtains tags for unlabelled images, for each specified dimension, so as to properly characterize their semantic content. Here we summarize the main idea of the image annotation process (for a complete description, please refer to [1]). Annotation is modelled as a nearest neighbor problem on image regions. The set R containing the k-NN regions of each region of a new image is first determined. The initial set T of tags for the new image equals the tags included in images containing regions in R; each tag in T is also given a frequency score f. However, tags in T might include unrelated, or even contradictory, terms. To overcome such limit, we exploit the pairwise *term correlation* by associating to each couple of tags a correlation score c. In particular, we reduce the cardinality of T by combining the scores f and c. To this end, we build an undirected and weighted graph G whose nodes correspond to tags in T with the highest values of f, whereas the weights are the f values. An edge between two nodes is added if their correlation score c exceeds a fixed threshold value. Starting from the graph G, we derive the set of final tags that are both *affine* to the new image and that share a *semantic correlation* among themselves by determine the maximum subset of fully connected nodes.

3. DEMONSTRATION

Let us illustrate a usage example of Scenique. In our system each image is automatically segmented into a set of homogeneous regions which convey information about color and texture features. Each region corresponds to a cluster of pixels and is represented through a 37-dimensional feature vector. With respect to regions comparison the Bhattacharyya metric is used. In the demo we will show results obtained on an image database of annotated images extracted from the Corel image collection.

First of all, the user builds several dimensions by means of the graphical tag tree editing functionalities offered by the GUI. Then she formulates the *content&tag* query "`(sea AND animal)`" by also supplying to the system her favorite *fish* image. Scenique returns images according to the integration rule above described. Depending on her preferences, the user can refine the tags associated to the returned images or assign new ones to images coming from the content-based retrieval only. Finally, for a new photo, she is interested in annotating it. Among the terms predicted by the system, each one associated to the proper dimension, the user can refine them by deleting wrong tags and/or by adding missing terms, depending on the precision of the provided result.

4. REFERENCES

[1] I. Bartolini and P. Ciaccia. Imagination: Accurate Image Annotation Using Link-analysis Techniques. In *AMR 2007*, Paris, France, July 2007.

[2] R. Fagin, R. V. Guha, R. Kumar, J. Novak, D. Sivakumar, and A. Tomkins. Multi-structural Databases. In *PODS 2005*, Baltimore, USA, June 2005.

[4] WordNet: http://wordnet.princeton.edu/

Multimodal User Interfaces for Smart Environments: The Multi-Access Service Platform

Marco Blumendorf, Sebastian Feuerstack, Sahin Albayrak
DAI-Labor, TU-Berlin
Ernst-Reuter-Platz 7, D-10587 Berlin
{Marco.Blumendorf, Sebastian.Feuerstack, Sahin.Albayrak}@DAI-Labor.de

ABSTRACT
User interface modeling is a well accepted approach to handle increasing user interface complexity. The approach presented in this paper utilizes user interface models at runtime to provide a basis for user interface distribution and synchronization. Task and domain model synchronize workflow and dynamic content across devices and modalities. A cooking assistant serves as example application to demonstrate multimodality and distribution. Additionally a debugger allows the inspection of the underlying user interface models at runtime.

Categories and Subject Descriptors
H.5 [**Information Interfaces and Presentation**]: User interfaces; D.2.2 [**Software Engineering**]: Design Tools and Techniques-*User Interfaces*; H.1.2 [**Models and Principles**]: User/Machine Systems-*Human factors;* H.5.2 [**Information Interfaces and Presentation**]: User Interfaces-g*raphical user interfaces, interaction styles, input devices and strategies, voice I/O.*

General Terms
Design, Human Factors

Keywords
Model-based user interfaces, runtime interpretation, Smart home environments, ubiquitous computing, multimodal interaction, human-computer interaction, interface design, usability

1. INTRODUCTION
Ambient environments comprising numerous networked interaction devices challenge interface developers to provide approaches that exploit these new capabilities. In this paper we describe an approach that addresses the need to adapt the interface to the environment. A runtime system, utilizing user interface models supports multimodal interaction and user interface distribution. The next section gives an overview of the developed system, followed by the description of an example, demonstrating the features.

AVI'08, May 28-30, 2008, Napoli, Italy.

2. THE MULTI-ACCESS SERVICE PLATFORM
The Multi-Access Service Platform (MASP) is a runtime system we created to address deployment and runtime issues when developing interaction in smart environments. The system therefore focuses on multimodal applications and follows a model-based approach. Based on a user interface model, the system allows controlling multiple user interfaces and is able to deliver the partial UI artifacts to different devices supporting different interaction modalities. Based on the runtime interpretation of the model the MASP is able to synchronize the distributed parts of such user interfaces (UIs).

The underlying user interface model is based on the ideas of the Cameleon Reference Framework [2] and similarly to UsiXML [5] separates multiple levels of abstraction. A task- and domain model define the workflow and dynamic data of the application, providing the basic information required for the interaction. The actual user interface is defined via templates providing final UI code (i.e. HTML and VoiceXML).

The task tree [4] defines the application workflow using the Concurrent Task Tree (CTT) notation [6]. Similar to [3] this allows assembling user interfaces based on the enabled task set. Objects, related to the identified tasks are defined as domain model allowing the exchange of information between tasks and with the backend. An object store holds the defined objects as dynamic content at runtime and thus provides access to the actual information for front- and backend. The connection to involved backend services is defined by a service model used to call the required backend services. The user interface is defined via multiple monomodal velocity (http://velocity.apache.org) templates associated with each task. These templates also incorporate the dynamic information from the object store. The selection of the active templates is carried out based on the active interaction tasks. The utilization of multiple monomodal interface templates allows forming a multimodal user interface. Interactions received via one of the modalities are interpreted and mapped onto domain object manipulations or task completions. In combination with interaction channels [1] that can be set up to interaction devices on the fly to render and transport the results of the templates, task completions and object manipulations are reflected in all active presentations, which allows the synchronization of the different monomodal UI parts via the underlying model.

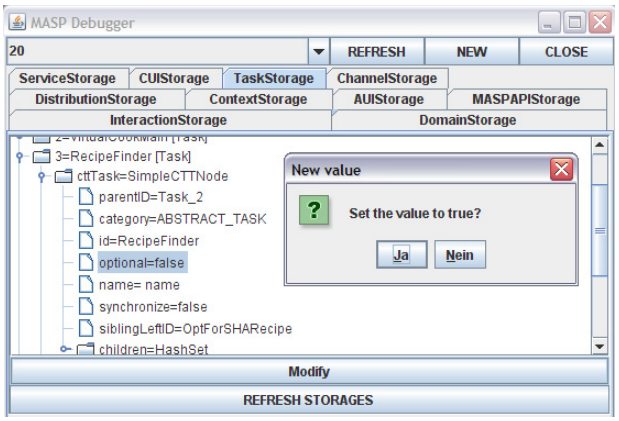


Figure 1: The MASP Debugger

Utilizing models at runtime also allows the inspection of the state of the application stored as dynamic part of the models. Figure 1 shows the debugger that can be used to browse through the actual state of the application. The tool connects to the runtime system and allows to inspect and alter the models for prototyping or direct manipulation of the running application. To further evaluate our approach we built an application using the multimodal interaction and distribution capabilities the described approach provides.

3. THE COOKING ASSISTANT

We developed a cooking assistant (CA) (Figure 2) as example application, to evaluate our runtime system. The CA has also been deployed as part of an ambient living testbed setup at the DAI-Labor at the TU-Berlin in the Service Centric Home project (www.sercho.de). The CA is based on three interaction steps. First the user selects a recipe, from the results of a search according to criteria given by the user. Afterwards an interactive dialog queries the user about what ingredients are available. Based on this information a shopping list is generated. Finally the CA guides step by step through the cooking process.

The whole application can be controlled via mouse, keyboard, touchscreen or voice and feedback from the system is provided via a graphical user interface as well as via voice output. The combination of the different modalities is determined based on the availability of the required interaction resources. Thus, the interactive querying of the availability of the ingredients can either be done via voice or via the graphical user interface. However, as the user has to move freely around in the kitchen, using voice interaction seems to be more appropriate in this case. Once the shopping list has been generated, the user can migrate the list to a mobile device using the distribution feature of the MASP. This allows to continue interaction during shopping, by marking the bough ingredients. Once shopping is done, the user indicates that, and seamlessly continues with the cooking assistant. The CA then guides step by step through the cooking process (Figure 2) and the user is able to control kitchen devices (e.g. turn on the oven) and request additional explanations in form of a video for each step. Device and video control as well as navigation between steps are possible via voice or the graphical user interface. Ingredients and step details are presented visually and via voice output. Voice input can be realized either via speaker dependent dictation or via speaker independent recognition. A small chat style interaction application allows text

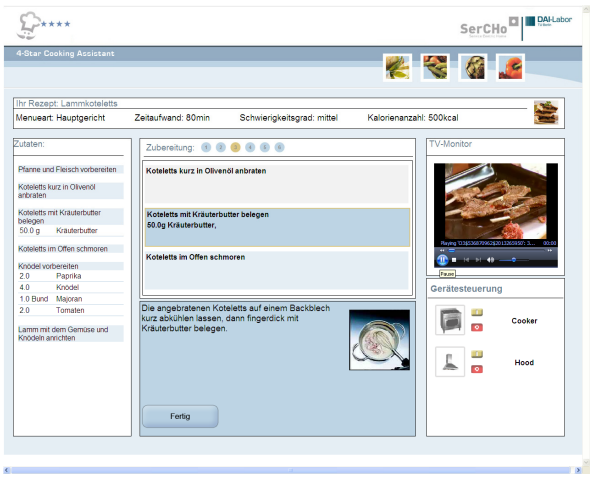


Figure 2: The graphical user interface of the cooking aid

input via dictation or the keyboard, e.g. to realize Wizard of Oz experiments.

The cooking assistant serves as example to demonstrate multimodal interaction based on voice and speech via the MASP. It shows how different channels and modalities can be added and removed on the fly. The shopping list scenario illustrates the capability to distribute the developed user interfaces across multiple devices while keeping the different parts synchronized.

4. ACKNOWLEDGMENTS

We thank the German Federal Ministry of Economics and Technology for supporting our work as part of the Service Centric Home project in the Next Generation Media program.

5. REFERENCES

[1] Blumendorf, M., Feuerstack, S. Albayrak, S. Multimodal user interaction in smart environments: Delivering distributed user interfaces. *European Conference on Ambient Intelligence: Workshop proceedings*, 2007.

[2] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 2003.

[3] Clerckx, T., Vandervelpen, C., Luyten, K., Coninx, K. A task-driven user interface architecture for ambient intelligent environments. In *Proceedings of IUI '06*.

[4] Feuerstack, S., Blumendorf, M. and Albayrak, S. Prototyping of multimodal interactions for smart environments based on task models. *European Conference on Ambient Intelligence: Workshop proceedings*, 2007.

[5] Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., and López-Jaquero, V. Usixml: A language supporting multi-path development of user interfaces. In EHCI/DS-VIS, volume 3425 of Lecture Notes in Computer Science, 2004.

[6] Paternò, F. *Model-Based Design and Evaluation of Interactive Applications*. Springer 1999.

Interactive Shape Specification for Pattern Search in Time Series

Paolo Buono
Dipartimento di Informatica
Università di Bari
Via Orabona, 4, Bari, Italy
+39-080-5442239
buono@di.uniba.it

Adalberto Lafcadio Simeone
Dipartimento di Informatica
Università di Bari
Via Orabona, 4, Bari, Italy
+39-080-5442299
simeone@di.uniba.it

ABSTRACT
Time series analysis is a process whose goal is to understand phenomena. The analysis often involves the search for a specific pattern. Finding patterns is one of the fundamental steps for time series observation or forecasting. The way in which users are able to specify a pattern to use for querying the time series database is still a challenge. We hereby propose an enhancement of the SearchBox, a widget used in TimeSearcher, a well known tool developed at the University of Maryland that allows users to find patterns similar to the one of interest.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: Graphical user interfaces (GUI).

General Terms
Languages.

Keywords
Interactive visualization, interactive system, information visualization, visual querying.

1. INTRODUCTION
Time series analysis involves the use of algorithms and tools that allow users to better understand a phenomenon. Time series can be defined as an ordered sequence of measurement that describes a phenomenon. A user may perform analyses on time series in order to describe or to explain a phenomenon or to perform forecasting [1]. In both cases it is important to use algorithms that associate some behavior.

TimeSearcher is a time series visualization tool, recently updated to version 3.0 [1, 2]. This tool allows the user to interactively search recurring patterns in its data [3]. To accomplish this task, users can draw a box (the *SearchBox*) enclosing the pattern of interest, then through a contextual interface they can start the search. We made informal user testing in order to understand how much usable the interface was. Based on the observations made we improved the SearchBox widget.

AVI'08, May 28–30, 2008, Naples, Italy.

2. TIME SERIES QUERYING
The search in TimeSearcher is composed by three steps. The user observes the time series and when (s)he finds an interesting pattern s(he) selects it and, by using the SearchBox, similar patterns are found. The users can tune the pattern search by interacting with the ToleranceHandle placed on the side of the box [3]. The ToleranceHandle sets the similarity degree. The process can be iterated. During the usability study we found that users would have liked to perform the pattern search starting from scratch, instead of searching first for an anomaly. In order to provide users with a widget that allows them to specify first a pattern shape and then start the search we improved the SearchBox with some interesting features.

3. SHAPE SPECIFICATION
Our goal was to allow users to specify the shape of a pattern in order to start the search in a time series. In literature there are some works that allow users to specify a query pattern. Don et al. [4] defined a set of typical patterns based on their shape; WizTree [5] allows users to specify the pattern behavior, after having transformed the series into an equivalent representation. Chortaras [6] permits to sketch the shape of the pattern, which is used as the input for finding similar patterns.

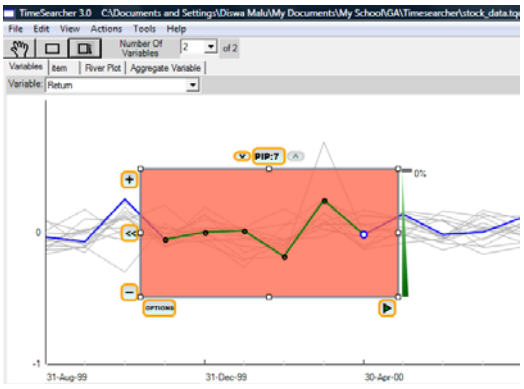


Figure 1 In the new SearchBox, the user can change the pattern in order to define the query pattern.

4. THE ENHANCED SEARCHBOX
As shown in Figure 1, as soon as the SearchBox is drawn on the interface, the time points belonging to the portion of the time series enclosed by the boundaries of the box are highlighted by colored disks. These disks will then be used as handles of the line segments to use for the purpose of pattern matching. The line formed by these handles is highlighted (in dark green). By

clicking and dragging the mouse over each disk, the user can then proceed to adjust the shape of the pattern. Users can also adjust the position of the active disk (the white one) by clicking on the *plus* and *minus* buttons located at the left corners of the box. Figure 2 shows the SearchBox after the user has dragged the white disc in order to decrease the third value of the pattern. The possibility to change a pattern was very appreciated by the users since they are not obliged to scan the dataset in order to find something to search for. Nevertheless, the users were still not completely satisfied because they could be interested to a specific pattern and they would like to depict it from scratch, eventually starting from a known shape. In order to satisfy this need we were inspired by Featurelens [4], a tool oriented to the analysis of text that uses a visualization technique similar to the one used in TimeSearcher.

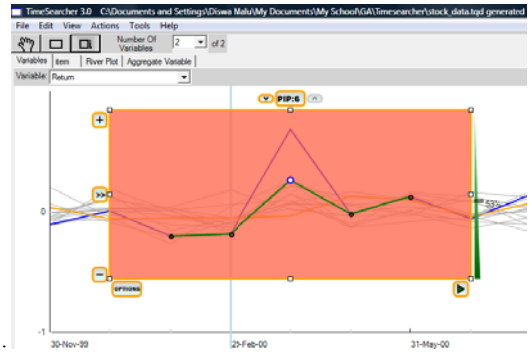


Figure 2 The user dragged the white disc into another position in order to change the original shape of the pattern.

We added the ShapesButton, a button that opens up a panel on the left side of the SearchBox containing several characteristic patterns (Figure 3). By clicking on it, the user will be presented with a side panel containing nine more buttons. Each one representing a common pattern that the user might be interested in adopting for the particular search that (s)he had in mind. These patterns are, from left to right: average, low peak, high peak (first row), increasing values, step up, step down (second row), decreasing values, valley, plateau (third row).

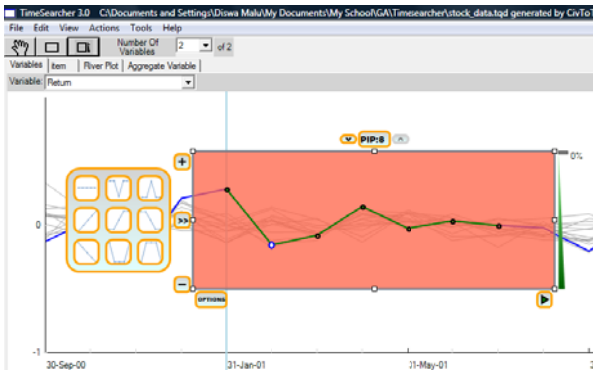


Figure 3 SearchBox with the ShapesButton exploded: the user can choose several pattern shapes in order to perform the query.

When the user moves its mouse cursor over one of these buttons, the pattern in the SearchBox is instantaneously changed to assume the shape shown in the button. This is, however, only a preview, the changes to the actual pattern shape will not be applied until the user decides to click on the relevant button. Moving the mouse pointer outside the bounds of the ShapesButton will cause the pattern in the SearchBox to revert to its original shape.

5. DISCUSSION AND USER FEEDBACK

Our approach in pattern specification shows some similarities with QueryLines [7], a tool developed by Ryall et Al, which also allows user to perform time series querying. In QueryLines, the user can manually draw lines in the display area of the plot to select and sort the results. A query line can be of two types: "soft constraint" and "preference". QueryLines's shape specification features are more geared towards fine-tuning time series querying rather than explicit pattern analysis.

We performed informal tests on this new feature of TimeSearcher and we received positive feedbacks and suggestions. The possibility of manually selecting the pattern of interest was very appreciated by the users; now they need a way to specify null values or uninteresting parts of the shape. In addition, it would be useful to add the possibility to save patterns, whose shape may be of interest. Users could be able to do so by hitting a key combination, once they are satisfied with the shape of the pattern. "Custom" shapes will be accessible by clicking on an appropriate button in the related panel. It should be observed that drawing the SearchBox over the display area is required because the extents of the pattern area have to be known in order to perform the search.

6. REFERENCES

[1] Jank, W., Shmueli, G., and Wang, S., (2006). Dynamic, real-time forecasting of online auctions via functional models. In Proc. 12th ACM SIGKDD (Philadelphia, PA). ACM Press, New York, NY (2006), pp. 580-585.

[2] Hochheiser, H., and Shneiderman, B., (2004). Dynamic Query Tools for Time Series Data Sets, Timebox Widgets for Interactive Exploration. *Information Visualization*, vol. 3, No. 1. (March 2004), pp. 1-18.

[3] Buono, P., Aris, A., Plaisant, C., Khella, A., and Shneiderman, B., (2005). Interactive Pattern Search in Time Series, In Proc. of Visualization and Data Analysis, VDA 2005, SPIE, Washington DC (2005) pp. 175-186.

[4] Don, A., Zheleva, E., Gregory, M., Tarkan, S., Auvil, L., Clement, T., Shneiderman, B., and Plaisant, C., (2007). Discovering interesting usage patterns in text collections: integrating text mining with visualization. In Proc. Information and Knowledge Management, CIKM 2007, pp. 213-222.

[5] Keogh, E. and Kasetty, S., (2003). On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration, *Data Mining Knowledge Discovery*, vol. 7, No. 4. (October 2003), pp. 349-371.

[6] Chortaras, A., (2002). Efficient Storage, Retrieval and Indexing of Time Series Data, Department of Computing, Imperial College of London, UK, 2002.

[7] Ryall, K., Lesh, N., Lanning, T., Leigh, D., Miyashita, H., and Makino, S., (2005). QueryLines: approximate query for visual browsing. In Proc. of Human Factors in Computing Systems (CHI 2005) Short Paper, pp. 1765-1768.

A System for Dynamic 3D Visualisation of Speech Recognition Paths

Saturnino Luz*
Dept. of Computer Science
Trinity College Dublin
Ireland
luzs@cs.tcd.ie

Masood Masoodian
Dept. of Computer Science
The University of Waikato
Hamilton, New Zealand
masood@cs.waikato.ac.nz

Bill Rogers
Dept. of Computer Science
The University of Waikato
Hamilton, New Zealand
coms0108@cs.waikato.ac.nz

Bo Zhang
Dept. of Engineering
The University of Waikato
Hamilton, New Zealand
bz48@waikato.ac.nz

ABSTRACT

This paper presents an interactive visualisation system that assists users of semi-automatic speech transcription systems to assess alternative recognition results in real time and provide feedback to the speech recognition back-end in an intuitive manner. This prototype uses the OpenGL libraries to implement an animated 3D visual representation of alternative recognition results generated by the Sphinx automatic speech recognition system. It is expected that displaying alternatives dynamically will facilitate early detection of recognition errors and encourage user interaction, which in turn can be used to improve future recognition performance.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems; H.5.2 [**User Interfaces**]: Natural Language

General Terms

Human Factors

Keywords

Automatic Speech Transcription, Interactive visualisation, Animated interfaces, Error correction

1. BACKGROUND

Automatic speech recognition (ASR) technology has progressed remarkably in the last decades, evolving from small-vocabulary research prototypes into commercial systems, with applications that range from domain-specific dialogue systems to unconstrained dictation. However, despite being clearly workable in a variety of applications, ASR remains an imperfect technology for which error correction mechanisms need to be carefully designed [1, 5].

The issue of error correction has been extensively studied in the area of spoken language dialogue systems where recognition rates and user acceptance of an imperfect input modality can be improved through clever interaction design and exploitation of domain constraints [2]. In applications such as dictation systems, for which domain-specific constraints will not readily come to the rescue of system designers, user-specific factors can sometimes be brought to bear. Dictation systems usually incorporate on-line training functionality which allows the system to adapt to the user's voice, thereby improving recognition rates above those attained by the baseline system.

When domain and user constraints fail, however, error correction will typically need to be done through an input modality other than speech [5]. This is often the case of speech transcription applications, where domains are unconstrained and speakers vary greatly in voice and accent. Error-correction in such applications has been dealt with by presenting the user with a linear transcript and allowing words to be highlighted, deleted, inserted or modified directly. In this scenario, the role of the ASR module ends once the initial, imperfect transcript has been produced. The user-corrected transcript then becomes the final product of the transcription process. More recently, applications have been proposed which extend the role of the recogniser by allowing it to effect global changes to the transcript as a result of local user feedback [3, 4]. However, these applications still employ a linear text document metaphor to mediate error correction and user feedback.

We have developed a prototype, called DYTRAED (DYnamic TRAnscript Editor), which uses the word lattice produced by the recogniser in order to propagate local error correction through the transcript, but also employs an animated 3D representation of the sentence being transcribed, showing alternative recognition paths as they unfold.

*This work was supported by a Science Foundation Ireland *Research Frontiers* grant.

AVI'08 28–30 May, Naples, Italy

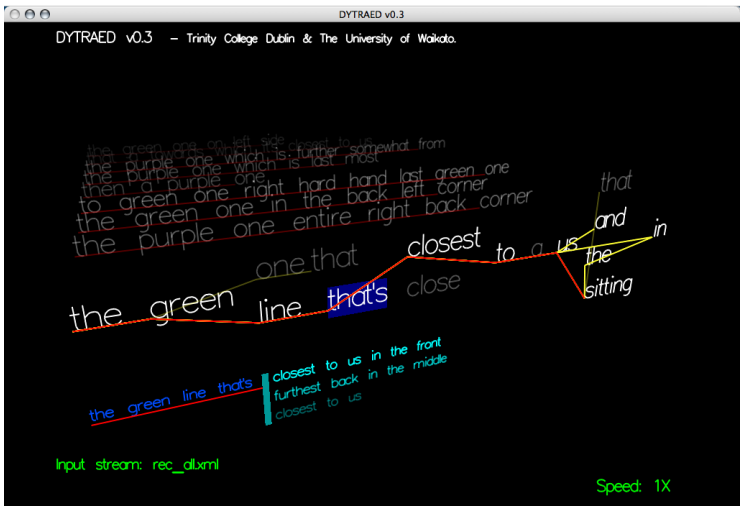


Figure 1: The user interface of DYTRAED

2. THE SYSTEM

The user interface of DYTRAED is shown in Figure 1. The user initially selects an audio source (live or recorded speech) and starts the transcription process. As recognition alternatives are generated by the ASR engine, the words are displayed as edges of a directed graph laid out on the middle-foreground of the application window. The partial recognition alternatives assigned the greatest scores by the ASR decoder are highlighted and connected through red-coloured edges. Lower-score hypotheses are dimmed, and alternatives undergoing active search (the rightmost words on the graph) are highlighted and connected to the choice point through yellow-coloured edges.

The user can pause the animation, increase or decrease its speed, and interact with the transcription graph. If the user clicks on a word (node) the animation stops and completion alternatives based on the buffered word lattice are presented. The user can then either select and alternative, thus accepting the entire sentence and bypassing the remainder of the visualisation for the current utterance, or simply ignore all options and continue to visualise the recogniser's preferred paths. This form of user input is illustrated in Figure 1, where the words next to the vertical bar near the bottom of the screen are possible sentence completions ordered by the likelihood assigned to them by the ASR engine.

As the recognition process ends for a given sentence (either through user selection or due to the system reaching the end of the search on the lattice) sentences move to the background. Old sentences are slowly pushed towards the horizon by newly finished sentences until they disappear completely from view. This form of presentation serves to maintain a degree of context of the transcription task visually available to the user without hindering the necessary focus on the current sentence.

DYTRAED uses Sphinx-4[1] as its speech recognition backend, and OpenGL for 3D rendering and animation. Sphinx encodes hypotheses actively under consideration by the recogniser (i.e. recognition paths that have not been prunned out) as a "word lattice" object containing acoustic scores (from the Hidden Markov Model) and language scores (in the present case, from a 3-gram language model). Our system displays "snapshots" of such structures and provides feedback to the search process through user interaction by, for instance, forcing the recogniser to select a lower-score path that would normally have been pruned out.

[1]http://cmusphinx.sf.net/

3. CONCLUSION AND FUTURE WORK

Informal evaluation suggests that our prototype can potentially increase user performance in ASR-assisted transcription tasks as well as making the experience more enjoyable. User feedback at the moment is restricted to the hypothesis already under consideration. We are currently working on mechanisms to incorporate new hypotheses (e.g. alternative segmentation, out-of-vocabulary words) to the system, along the lines of what has been done in [4].

4. REFERENCES

[1] W. A. Ainsworth and S. R. Pratt. Feedback strategies for error correction in speech recognition systems. *International Journal of Man-Machine Studies*, 36(6):833–842, June 1992.

[2] K. S. Hone and C. Baber. Modelling the effects of constraint upon speech-based human-computer interaction. *Interface Journal of Human-Computer Studies*, 50(1):85–107, 1999.

[3] P. Liu and F. K. Soong. Word graph based speech rcognition error correction by handwriting input. In *Procs. of the 8th Intl. Conference on Multimodal Interfaces*, pages 339–346. ACM Press, 2006.

[4] M. Masoodian, B. Rogers, and S. Luz. Improving automatic speech transcription for multimedia content. In P. Isaías and M. B. Nunes, editors, *Proceedings of WWW/Internet '07*, pages 145–152, Vila Real, 2007.

[5] B. Suhm, B. Myers, and A. Waibel. Multimodal error correction for speech user interfaces. *ACM Trans. Comput.-Hum. Interact.*, 8(1):60–98, 2001.

Perspective Change: A System for Switching Between On-screen Views by Closing one Eye

Fabian Hemmert
TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin
fabian.hemmert
@telekom.de

Danijela Djokic
FH Potsdam
Pappelallee 8-9
14469 Potsdam, Germany
djokic@fh-potsdam.de

Reto Wettach
FH Potsdam
Pappelallee 8-9
14469 Potsdam, Germany
wettach@fh-potsdam.de

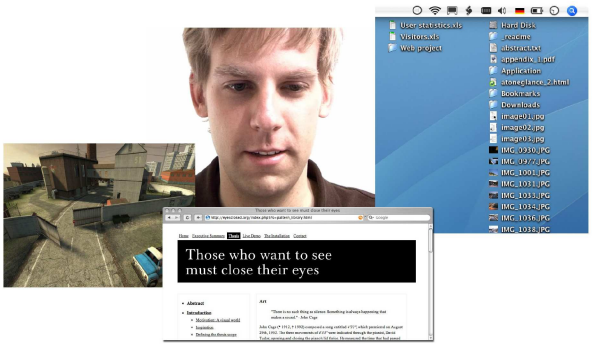


Figure 1: Normal viewing conditions for SN, ZB and DF (from left to right)

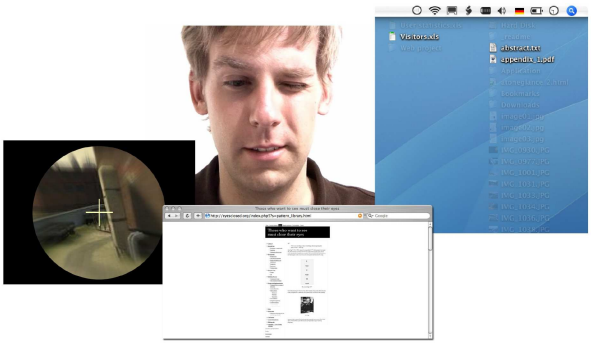


Figure 2: Alternative perspectives for SN, ZB and DF (from left to right), displayed for the duration of the single-sided eye closure

ABSTRACT

This project explores the change of on-screen views through single-sided eye closure. A prototype was developed, three different applications are presented: Activating a sniper scope in a 3D shooter game, zooming out into a overview perspective over a web page, and filtering out icons on a cluttered desktop. Initial user testing results are presented.

Categories and Subject Descriptors

H.5.2 [**Information interfaces and presentation**]: User Interfaces - Input devices and strategies

Keywords

Eye, eyelid, eye closure, perspective change, prototype, screen interface

1. INTRODUCTION

Eye closure has been used as a user input before [3], often as a suspension for a mouse button, often in eyegaze-based systems for disabled people [4]. The obvious problem here is that the eye is a sensory organ, and not an actuator and we are simply not trained or used to *cause something* by closing an eye [6][1].

We are, on the other hand, familiar with the fact that when looked at with only one eye, things look a little different: We experience a slight shift in our perspective, and our field of view is slightly narrowed. One goal of this project was to find out how this principle could be transferred to the HCI context. Different views on the same data are commonly used in software interfaces. Point-of-view changes, as the page view in a word processor; Data changes, as infrastructural data laid over satellite images in a map view; Tool changes, as the "layout" and "code" views in a HTML editor - the spectrum of perspective changes in user interfaces is broad. However, the ways these perspective changes are controlled by the user are not always satisfying. Predominantly, they are controlled through mouse and keyboard actions: While these are agreed-on means, none of the current interaction schemes for on-screen perspective change is really *intuitive* or *direct*[5].

If we could understand how on-screen view changes could be made accessible to computer users in a more natural and effective way, this would enable us to design interfaces that would allow users to get an understanding of more complex matters, faster and better than before. Surely, and as we found out in our own observation, not everybody can close one eye, and only few people can do it effortlessly. Closing one eye is, however, a regular voluntarily accessible muscle

AVI '08, 28-30 May , 2008, Napoli, Italy.

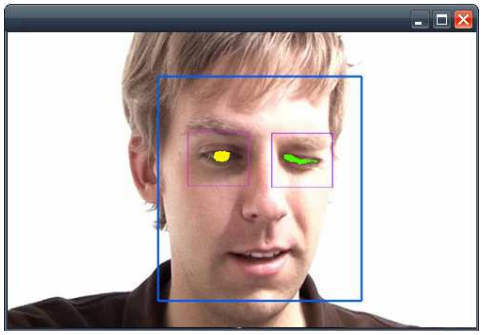


Figure 3: The eyelid tracking software, distinguishing between open and closed eyes

activity, and can be trained as a such. On-screen perspective changes in the human-computer context are activated in a unsatisfying way - we propose to explore the potential of single-sided eye closure as an alternative input modality for this particular issue.

2. SWITCHING VIEWS BY CLOSING AN EYE

We implemented a simple eyelid tracking system, using a face finding software [2] and a standard web cam. The face finding software delivered individual data for every eye, and with image processing, we were able to determine if a pupil was visible in it or not (Fig. 3). The prototype was implemented on two computers, one performing the tracking, and one running the applications. To communicate with the application computer, we enabled the tracking software to generate virtual keyboard events through a Java interface and sent them through a remote control software.

2.1 Initial User Test

Over the design of the three applications presented in the following, a small set of users (4f, 5m, avg. age 25.43 yrs., including both novice users and computer experts) was asked to test the system, and compare it to their experiences from the past.

2.2 Applications

In a iterative process, we developed three applications. Each was tested with the users, and their comments (which we present as quotes) were taken into consideration for the design of the next application.

2.2.1 Sniper Scope (SN)

The first application we implemented was a sniper scope for a first-person shooter game. We modified the game so that the zoom functionality (which was originally controlled by holding down the SHIFT key) can be activated by closing one eye. As soon as the user returns to normal looking, the normal perspective is restored (Fig. 1, Fig. 2). We hypothesized that the eye-based interaction would be more intuitive and also faster than its keyboard-based equivalent. The users in our test were of a different opinion, as they stated that the eye-activated sniper feature was "innovative and cool", but "very exhausting at the same time" - due to that, the overall acceptance was comparably low.

2.2.2 Zoom Browser (ZB)

For the following iteration, we developed a *page view mode* for a web browser. In the prototype, the web page zooms out (fit to the height of the window) when the user closes an eye (Fig. 1, Fig. 2). This enables the user to select a new target area (using the mouse) and zoom in to this area as soon as he returns to normal looking. According to their comments in the interviews, the users liked the functionality. However, also the users in the experimental group asked why this feature couldn't just be controlled "with a simple keystroke" - a question that lead us to the design of the third application. Both of the first applications had a strong character of triggering an event in the computer - something that is usually connected to a keystroke. For the third implementation, we sought a more subtle change.

2.2.3 Desktop Filter (DF)

In the final application, we implemented a perspective change that generated the impression that the screen's contents would *look* different when they were watched with one eye. Our prototype consists of a Mac OS X desktop, cluttered with icons. When the filter is activated, all but the 5 most recently edited files are are faded out (Fig. 1, Fig. 2). The user comments confirmed that it was "easy to remember" and "associate the changed view with the eye action", and the overall acceptance for this application was very high, compared to the other two implementations.

3. CONCLUSIONS AND OUTLOOK

Surely, the user study we conducted does not allow us to draw hard conclusions about the value of the system. However, the last implementation hints to a possible benefit that should be explored. We find this project points into a interesting direction - if subtle changes in the visual presentation, activated in an intuitive way, could help us to improve how we understand visual information in a better way, that would be very beneficial.

4. REFERENCES

[1] C. Evinger, K. A. Manning, and P. A. Sibony. Eyelid movements. mechanisms and normal data. *Invest. Ophthalmol. Vis. Sci.*, 32(2):387–400, February 1991.

[2] B. Fröba and C. Küblbeck. Robust face detection at video frame rate based on edge orientation features. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, DC, USA, 2002. IEEE Computer Society.

[3] M. Kumar and T. Winograd. Guide: gaze-enhanced ui design. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 1977–1982, New York, NY, USA, 2007. ACM.

[4] C. Lankford. Effective eye-gaze input into windows. In *ETRA '00: Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 23–27, New York, NY, USA, 2000. ACM.

[5] A. Raskin. Why modes kill. *http://www.humanized.com/weblog/2006/12/07*.

[6] F. Van Der Werf, P. Brassinga, D. Reits, M. Aramideh, and Ongerboer. Eyelid movements: Behavioral studies of blinking in humans under different stimulus conditions. *J Neurophysiol*, 89:2784–2796, 2003.

Improving citizens' interactions in an e-deliberation environment

Fiorella De Cindio
University of Milan
fiorella.decindio@unimi.it

Cristian Peraboni
University of Milan
cristian.peraboni@dico.unimi.it

Leonardo Sonnante
Fondazione RCM
leonardo.sonnante@rcm.inet.it

ABSTRACT
In an e-deliberation environment it is particularly important to conceive tools and web interfaces able to facilitate social online interactions between citizens and public officers. In this paper we present some choices made in the development of an e-deliberation platform. In particular we will focus on the use of maps to facilitate citizens interaction based on geo-localized discussions, and on the design of an ad hoc interface for online discussion to increase citizens' participation.

Categories and Subject Descriptors
H.5.1 [**Information interfaces and presentation**]: Multimedia Information Systems – *hypertext navigation and maps.*

H.5.3 [**Information interfaces and presentation**]: Group and Organization Interfaces – *Web-based interaction*

General Terms
Design, Experimentation, Human Factors.

Keywords
e-participation, e-deliberation, map-based interaction, web interfaces, web-based social interaction.

1. INTRODUCTION
In 2003 Italy's Ministry for Innovation and Technology issued a "Call for selecting projects to promote digital citizenship (e-democracy)". Ten municipalities in the Lombardy Region (Mantua – the coordinator, Brescia, Como, Desenzano sul Garda, Lecco, Malgesso – as coordinator of a consortium of several small municipalities – Pavia, San Donato Milanese, Vigevano, and Vimercate), some with previous experience managing community networks, presented a project named "*e21* for the development of digital citizenship in Agenda 21" under the scientific coordination of A.I.Re.C., the association for community networking set up in 1996 under a protocol of cooperation between the Lombardy Regional Government and the Department of Informatics and Communication of the University of Milan.

The purpose of the project is to overcome the hindrances to participation typical of local Agenda 21 processes - as described, for instance, by Evans and Theobald [6] - by creating a social environment on a custom-designed, dedicated online deliberation platform. This environment is called Deliberative Community Networks (DCN for short) as it aims at improving the so to say "traditional" community networks by introducing deliberative tools [4]. DCN have been conceived by considering the steps of a typical local Agenda21 process and the deep analysis of several participatory processes, as reported in [1].

DCN are organized in three interrelated spaces: the community space (aimed at facilitating the rise of mutual trust between participants), the deliberation space (that is the core of the participatory system and aims to foster the creation of a shared vision position among the group members) and the informational space (aimed at facilitating sharing and collection of information provided by citizens to support group activities) that is integrated in both the others two spaces. In [2] we present a set of online deliberation tools that seem to be necessary to support complex participatory processes.

However, in the framework of the e21 project, the field analysis conducted in the ten municipalities, the discussions we had with the public officers involved in the project as well as a previous experience done with citizens in Milan [3] have influenced, and slightly modified, the implementation priorities. As a consequence, up to now we have developed the so called **City Map** in the community area, and three deliberative tools: the **Informed discussion** is an enriched forum with facilities for sharing documents that support the discussion and for producing, in a collaborative way, a document that summarizes it; **Certified citizen consultation** allows the promoters of a deliberative process to poll participants, who choose among alternatives come out in earlier deliberation steps; **Online deliberation** structures synchronous debates with a formal protocol (i.e., a set of rules embedded in the software), inspired by the Robert's Rule of Order [8], to assure each one the possibility of presenting her/his positions, and the majority to deliberate.

This paper presents how well known web-based interaction techniques have been applied in this e-participation environment, which is characterized by the need of involving a high number of people (ordinary citizens as well as politicians) that cannot be assumed familiar with online interactions. The next two sections present the City Map and the Informed discussion, with particular attention to apparently minor facilities that our experience and the preliminary field analysis have identified as relevant.

2. A MAP OF THE PARTICIPATION
The city map is the main tool of the community space and represents the map of participation events of the city. This tool allows people to localize all the participation experiences (free

discussions and deliberative processes) in a topographic map. The idea is to allow citizens to "tag" places of the city (streets, public square, and so on) with discussions and documents related to them. In this way it is possible to collaboratively build a representation of the city based on the civic intelligence [9] collected and shared through the discussions.

The Community Network experience shows how difficult it is to manage the knowledge created through public dialogue: communities, as well as blogs, - when successful - produce in a short time a huge amount of cognitive materials (messages in the forums, documents attached to them, posts and comments in blogs, etc.) that grow into a not structured collection of cognitive items, often displayed in a mere linear way [7]. As a consequence, it is more and more difficult for participants to retrieve the information they need, e.g., for joining in a discussion that aims at taking a decision. The city map provides a first answer to these difficulties, by organizing different civic contents on the basis of the direct people's experience of the city. It is worth noting the use of maps is only a *partial* solution because there are theme of relevance that cannot be easily attached to a location (e.g. a general discussion on 'Solutions to solve the traffic problem in Milan"). The concrete guideline we give right now to face with this limit is to attach these transversal themes to the place where the City Hall has its main building. We are now working on a tag-based solution that will organize the system's entire content on a simplified conceptual map based on a users' tag-generated 'folksonomy'.

From a technical point of view, the city map uses the Google Maps© and the related APIs. Every free discussion is represented in the map with a little balloon icon (red if the discussion is started by a citizen, blue if it is started by a local government officer), while every participatory process is represented by a bigger blue balloon icon (blue because only local government can start a participatory process).

3. ENHANCING ONLINE DISCUSSIONS

In order to reduce the barriers that hinder the ordinary citizens' participation in a online deliberative process, instead of adopting one of the existing software for managing discussions, we choose to develop the Informed discussion tool, which includes three distinguished features.

The first feature concerns the visualization of the messages in a thread of discussion. Usually messages/posts in forums and blogs are presented either in strictly chronological order or in indented threads. In the first case, a post which is the answer to a specific previous post does not appear close to it. In the second one, only message headers are usually displayed: when a reader opens a message, its context get lost. To overcome these limits, we have adopted the second alternative, but, thanks to a simple JavaScript, when someone wants to read a message, the body of message is opened (and then closed) within the same web page that hosts the message list, so to maintain the context awareness of the discussion. This solution helps citizens to visualize at a glance in a single web page the nesting of posts and replies, and, if it is the case, to put their own post in the right position.

The second feature aims at improving the rationality of the discussion, by providing an organized visualization of informative resources (documents, links, videos, etc). For every discussion, it exists an informative space that collects all the materials attached to single posts or directly uploaded. Participants can visualize, in the same page, the discussion and its informative materials.

The third feature imports a typical feature of social network environments: it allows citizens to express their (level of) consensus and to flag as relevant posts or informative materials. This is done by assigning a numerical value (from 1 to 4) both to messages and documents. In such a civic environment, allowing people to express agreement provides (technological) support for what Edward [5] calls different *styles of citizenship*: one "stronger," more active, and another apparently "weaker". This may be important for extending participation.

4. CONCLUSION

In this paper we have shortly presented some significant features of the first components of an online deliberation system. These features are thought to increase citizens' participation in online deliberative processes. The feedbacks by users during the current field experiments will provide input for further improvements.

5. REFERENCES

[1] Bobbio, L. 2004. *A piu' voci - Amministrazioni pubbliche, imprese, associazioni e cittadini nei processi decisionali inclusive*. Edizioni Scientifiche Italiane, Napoli. (in Italian).

[2] De Cindio F., De Marco A., and Grew P. 2007. "Deliberative community networks for local governance", *Int. Journal of Technology, Policy and Management*, 7, 2 (2007), 108-121.

[3] De Cindio, F., Di Loreto, I., and Peraboni, C. 2008 **(**forthcoming**).** Moments and Modes for Triggering Participation at a Local Level. In *Urban Informatics: Community Integration and Implementation,* Foth, M. (Ed.). Hershey, PA: Information Science Reference, IGI Global.

[4] De Cindio F. and Schuler D. 2007. Deliberation and Community Networks: A Strong Link Waiting to be Forged. In *Proceedings of CIRN Conference 2007 "Communities and Action"* (Prato, Italy, November 5-7th, 2007).

[5] Edward, A. 2006. Online Deliberative Policy Exercises and Styles of Citizenship: Issues of Democratic Design. Position paper presented at the *DEMOnet Workshop on eDeliberation Research* (Leeds, UK, October 16th, 2006).

[6] Evans, B. and Theobald, K. 2003. "Policy and Practice LASALA: Evaluating Local Agenda 21 in Europe", *Journal of Environmental Planning and Management,* 46(5)781-794.

[7] Macintosh, A. 2006. Argument Maps to Support Deliberation. Position paper presented at the *DEMOnet Workshop on eDeliberation Research* (Leeds, UK, October 16th, 2006).

[8] Robert, H. M. III, Evans, W. J., and Honemann, D. H. 2000. *Robert's Rules of Order Newly Revised*, *10th ed*. Perseus Publishing, Cambridge, Mass.

[9] Schuler, D. 2001. "Cultivating society's civic intelligence: patterns for a new 'world brain'", *Journal of Information, Communication and Society*, 4

"Isn't This Archaeological Site Exciting!": a Mobile System Enhancing School Trips

Carmelo Ardito, Rosa Lanzilotti
Dipartimento di Informatica, Università di Bari, 70125 Bari, Italy
{ardito, lanzilotti}@di.uniba.it

ABSTRACT
Explore! is an m-learning system that aims to improve young visitors' experience of historical sites. It exploits the imaging and multimedia capabilities of the latest generation cell phone, creating electronic games that support learning of ancient history during a visit to historical sites. Explore! consists of two main components: 1) the Game Application running on cellular phones, to be used during the game and 2) the Master Application running on a notebook, used by the game master (i.e. a teacher) to perform a reflection phase, which follows the game. Having the Game Application been described in previous papers, in this work we mainly illustrate the Master Application.

Categories and Subject Descriptors
K.3.1 [**Computer Uses in Education**]: *Collaborative learning*

General Terms
Design, Human Factors.

Keywords
Learning game, mobile system.

1. EXPLORE!
Explore! is an m-learning system that supports middle school students during a visit to an archaeological park with their teachers. It adopts a learning technique called *excursion-game*, whose aim is to help students to acquire historical notions while playing a game on a cell phone and so make archaeological visits more effective and exciting.

The main system components are the *Game Application*, running on standard cell phones Java Micro Edition (J2ME) compatible, used by students during the game; and the *Master Application* running on a PC or a notebook, used by the game master (i.e. a teacher) to perform a reflection phase, which follows the game.

Students play the game in groups of 4 or 5. The Game Application is provided on a phone memory card, which is handed out to each group at the start of the game session. All data exchange takes place between the cell phone and the memory card inside it: no data are transmitted from or to the phone during the actual game, thus reducing communication costs and time.

AVI'08, May 28-30, 2008, Napoli, Italy

The Game Application is developed using Java Micro Edition (J2ME). Three packages are requested, which are currently provided by default in cell phones supporting J2ME: JSR75 (for managing XML files), JSR184 (for visualizing the M3G files containing the 3D models), and JSR234 (for reproducing multimedia). In our trials, the game was executed on a Nokia E70 handset but it has also been successfully run on a Nokia 6630. The game master's notebook is equipped with either Bluetooth or a memory card reader, for the application to collect the logfiles from the groups as they come in at the end of the game. It was developed using the Microsoft .NET framework.

Explore! lets students interact with the 3D reconstructions of historical monuments. They are developed using 3D StudioMax and exported to the M3G file format. If the user executes the Game Application on a phone with little power, the "real" 3D M3G files can be substituted by a sequence of snapshots of the 3D models taken while rotating around the object by 360 degrees in 3D Studio Max. We have evaluated different versions of this image player: the version the users preferred permits a very basic zoom-in and zoom-out from any desirable viewing angle. Two sequences of eight snapshots have been taken from viewpoints differing by 45 degrees, at two different distances from the monument [1].

The main novelty of Explore! is its slim architecture that aims at reducing implementation costs and architectural complexity to absolve the archaeological park from any need to invest in hardware infrastructure. Most middle school students have a cell phone, so we can assume that at least one student in each group will own one. The Game Application has been described in other papers [1, 2]; in the following section we also illustrate the Master Application. It is worth noting that Explore! is applicable to a wider set of historical sites. The way historical information is presented (time, location, modality) is determined by an XML file and can thus be authored in numerous ways and adapted to different archaeological parks. We are currently developing an authoring tool to be used for this purpose. The figures shown here refer to the visit to the archaeological site of Egnathia. The demo will also show a possible implementation for other sites. Future works will include an adaptation of the system for use by families visiting the site with their children.

2. "GAIUS' DAY IN EGNATHIA"
"Gaius' day in Egnathia" is an excursion-game we have implemented in Explore! for a visit to the archaeological park of Egnathia, an ancient city in the Apulia region [2]. "Gaius' Day" is structured like a treasure hunt to be played by a class of students. This type of game is ideally suited to the archaeological park context, with wide spaces where students can move about freely and use their intelligence and imagination to conjure up how life

used to be there, by observing the park and memorizing places, names and functions.

The game consists of three main phases: introduction phase, game phase, debriefing phase. In the *introduction phase,* the game master gives a brief description of the place and explains the game. Groups of 4/5 players are formed: each group impersonates a Roman family that has just arrived in Egnathia. In the *game phase,* each group is given a cell phone and the map. The challenge is to carry out ten missions, visualized on the phone screen one at a time, which require students to walk around and look for the mission target. Players provide their answer to a mission by typing in the place code on the cell phone. After completing the challenge, the group receives "God's gifts": they can explore the 3D reconstruction of the identified places on the phone and visually compare how the places probably once looked with the existing remains (Fig. 1).

Fig. 1 The remains of the Trajan Way (left) and the phone 3D reconstruction of how it probably looked in the past (right).

The *debriefing phase* is a reflection phase, following the true game, in which the acquired knowledge is reviewed and shared among students. Using the Master Application, the game master plays a "collective memory game" where monuments and archaeological objects must be placed in the "right" place. One at a time, a thumb image of the 3D reconstruction of a site element and its name are shown on the left of the screen visualizing the notebook display; the possible positions where the buildings could be placed on the digital map are marked by the letters of the alphabet (Fig. 2). When the students have decided on which letter the element should be placed, the game master or a student will "drag&drop" it onto the letter. If the position is wrong, the system highlights the place with a red oval, a negative acoustic feedback is reproduced and an error message is displayed. Otherwise, the acoustic feedback is positive and a green oval appears on the letter. The 3D reconstruction of the element is now visualized on the screen, and the game master goes back over the concepts that Explore! illustrated during the game phase, so as to analyze in more detail the place at the time of ancient Romans. After all the proposed 3D reconstructions have been correctly placed, the system analyzes the logfiles collected from the cell phones and proclaims the winning group, showing the groups' standings (indicated as "Podium" in Fig. 3). The system can replay the activities of an arbitrary group showing the path they took across the archaeological park (Fig. 3).

Fig. 2 A thumb image of the site elements (left), and the positions where to place the buildings.

Fig. 3 The "Podium" showing the groups' standings (left) and the path a group followed during the game across the archaeological park.

3. ACKNOWLEDGMENTS

The financial support of the Italian MIUR through grant "CHAT" is acknowledged. We thank students M. De Bartolo, M. Giampietruzzi and A. Giannelli, for their contribution to implementing the system.

4. REFERENCES

[1] Ardito, C., Buono, P., Costabile, M.F., Lanzilotti, R., Pederson T. 2007. An Augmented Reality Game on Standard Mobile Phones for Exploring History at Archaeological Parks. R. Meersman et al. (Eds.), LNCS 4805, pp. 357–366. Berlin: Springer (Germany).

[2] Costabile, M.F., De Angeli, A., Lanzilotti, R., Ardito, C., Buono, P., Pederson, T. 2008. Explore! Possibilities and Challenges of Mobile Learning. In Proceedings of CHI 2008 (Florence, Italy, April 5-10, 2008). ACM Press, New York, NY. In press.

MedioVis – Visual Information Seeking in Digital Libraries

Mathias Heilig, Mischa Demarmels, Werner A. König, Jens Gerken, Sebastian Rexhausen, Hans-Christian Jetter, and Harald Reiterer

University of Konstanz
HCI Group
Box D-73, 78457 Konstanz, Germany
+49 7531 883066

{firstname.lastname}@uni-konstanz.de

ABSTRACT

MedioVis is a visual information seeking system that aims to support users' natural seeking behavior, particularly in complex information spaces. To achieve this goal we introduce multiple complementary visualization techniques together with an easy-to-use and consistent interaction concept. Over the last four years, MedioVis was developed in the context of digital libraries following a user-centered design process. The focus of this paper is the presentation of our interaction model and further to give an overview of the applied visualization techniques.

Categories and Subject Descriptors

H 5.2 [Information Interfaces and presentation]: User Interfaces - Graphical user interfaces, Interaction styles, User-centered design

General Terms

Design, Human Factors.

Keywords

Interaction Design, Semantic Zooming, Coordinated Views.

1. INTRODUCTION

Nowadays users of digital libraries are confronted with information that is rapidly growing in quantity, heterogeneity and dimensionality. Therefore, more effective tools are required to facilitate the exploration and search in this information space. We propose MedioVis as a flexible application for the visual exploration of such data that is especially designed for users without prior professional experience in search, retrieval or visualization [2]. The project was launched four years ago and still undergoes iterative development and evaluation cycles. To gain continuous end-user feedback and insights in real interaction behavior, we are running MedioVis for over three years in the media library of the University of Konstanz.

2. SYSTEM CHARACTERISTICS

Due to the increasing complexity of user-accessible information spaces in digital libraries a single visualization is not able to sufficiently cover the various information needs and seeking strategies.

AVI'08, May 28-30, 2008, Napoli, Italy.

Thus, MedioVis offers the possibility to flexibly collocate complementary visualizations to provide several views to different dimensions of the information space. This is realized by multiple coordinated views [1] connected through the technique of snap-together [5] and linking and brushing. Hence, the user is able to define mutual filters in different visualizations to narrow down the data space to a relevant subset. The synchronized visualizations provide the user with instant feedback and a powerful but straightforward filter mechanism. The user can directly manipulate the selection and arrangement of visualizations, by dragging and dropping them from the tool bar onto the desired area in the application window.

An information seeking process is often a combination of several searches. The user must be able to switch between different search paths without losing the afore gathered information. To support this non-linear search strategy, we integrated a tab concept, similar to multiple document interfaces.

Figure 1. MedioVis: Multiple Coordinated Views with HyperScatter (top) and HyperGrid (bottom).

To create a pleasurable and satisfying user experience we developed an attractive and deliberate visual design in cooperation with communication designers. Additionally, we integrated multimedia content (e.g. images, videos, web pages) as well as supportive and natural user interaction concepts (e.g. animated zooming, direct manipulation) to increase the joy of use.

3. VISUALIZATION TECHNIQUES

In consequence of our gathered experiences and evaluation results [2], MedioVis applies multiple visualizations that complement the

features of each other. We intentionally decided to use and combine visualizations that base on well-known and straightforward concepts (e.g. tables, scatter plots). Crucial for the design was the applicability for analytical and browsing oriented ways of data exploration (e.g. dynamic queries or details on demand).

To get an overview over the entire data space at a glance, we introduce the HyperScatter (see figure 1) as a zoomable, two-dimensional scatter plot. It enables the user to explore relations between the data objects along different user-adjustable dimensions and to recognize patterns. Furthermore, the interactive visualization can be used for visual filtering, through animated zooming into a user-defined area of the plot and thereby offers a natural way of query formulation and refinement. Depending on the user's information demand, the HyperScatter also allows progressive access to detail information through continuous semantic zooming [6] into specific data objects. This detail on demand technique, realized by semantic zooming is a general interaction concept of MedioVis. With this technique, we intent to avoid information overload. The user decides through zooming into a region of interest, which information is important in a certain context.

The HyperGrid (see figure 1) applies the zoomable user interface concept on a well-known table visualization. It allows filtering, sorting and selecting of individual data objects, which are presented in columns [4]. Furthermore the HyperGrid enables the user to explore meta data and related external multimedia content (Web 2.0 content like Google Maps, Wikipedia entries etc.) through semantic zooming into table cells. There, it is even possible to access the real data object (e.g. full text in a PDF document, streamed video). Thus, typical problems like "change blindness" or "loss of orientation" are avoided. As a result, the system allows a browsing-oriented discovering of the data space without leaving the context of the table. In addition, the HyperGrid is very appropriate to compare two or more data sets through the structured nature of a table.

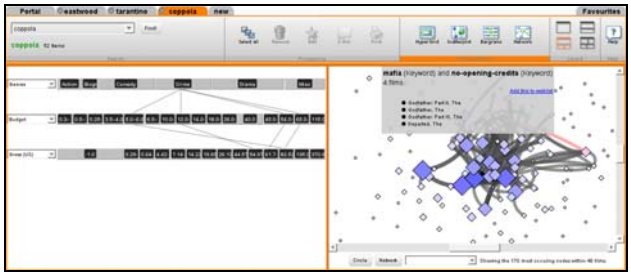


Figure 2. MedioVis: Multiple Coordinated Views with Parallel Bargrams (left) and Network Visualization (right).

With the parallel bargrams (see figure 2), inspired by [7], we provide a different entry point to the data by giving an overview of the attribute space rather than looking at the objects themselves. The amount of data objects with a certain attribute value is mapped onto the length of a section of a bar. By showing several bars beneath each other, the system allows to examine multiple attributes at once. The connecting lines between the bars evolve in a parallel coordinate visualization [3], exposing relationships and characteristic distributions between diverse attributes.

The network visualization (see figure 2) enables the user to analyze relationships between data objects through connecting attribute values (e.g. tags, authors). The network illustrates these values as nodes, where the number of occurrences is mapped to their size and color. The data objects with similar attribute characteristics are represented by connecting edges.

Since a single visualization cannot completely cover all kinds of information needs, MedioVis combines the introduced visualizations to overcome their limitations. The users for example may use sequentially or parallel the HyperScatter, with its good possibility to gain an overview and narrow down the data set and the HyperGrid to explore and compare the remaining data sets. Furthermore, by the applied interaction techniques like linking and brushing the user may gain an enhanced knowledge and deeper understanding of the information space.

4. CONCLUSION

With MedioVis, we offer an innovative visual information seeking system for end-users. To give a satisfying search experience, we faced the challenges of providing different views on the data space and of supporting analytical and browsing oriented exploration strategies through the usage of multiple coordinated visualizations and a consistent and supportive interaction design. The concepts we developed in the context of digital libraries can also be transferred onto other information seeking domains. For example, we successfully applied them on personal information management, virtual museums and image retrieval. MedioVis runs in the library of the University of Konstanz and is available as an open source project[2].

5. REFERENCES

[1] M. Q. W. Baldonado, A. Woodruff and A. Kuchinsky. Guidelines for using multiple views in information visualization. In AVI '00: Proceedings of the Working Conference on Advanced Visual Interfaces, 2000, pp. 110-119.

[2] C. Grün, J. Gerken, H. Jetter, W. König and H. Reiterer. MedioVis - A user-centred library metadata browser. In Proceedings of the 9th European Conference, ECDL, Research and Advanced Technology for Digital Libraries, 2005, pp. 174-185.

[3] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In VIS '90: Proceedings of the 1st Conference on Visualization '90, 1990, pp.361-378.

[4] H. Jetter, J. Gerken, W. König, C. Grün and H. Reiterer. HyperGrid -accessing complex information spaces. In People and Computers XIX - the Bigger Picture, Proceedings of HCI 2005, 2005.

[5] C. North and B. Shneiderman. Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In AVI '00: Proceedings of the Working Conference on Advanced Visual Interfaces, 2000, pp. 128-135.

[6] K. Perlin and D. Fox. Pad: An alternative approach to the computer interface. In SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, 1993, pp. 57-64.

[7] K. Wittenburg, T. Lanning, M. Heinrichs and M. Stanton. Parallel bargrams for consumer-based information exploration and choice. In UIST '01: Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, 2001, pp. 51-60.

[2] http://sourceforge.net/projects/mediovis

End-User Visualizations

Alexander Repenning
AgentSheets, Inc.
6560 Gunpark Dr. Suite D
Boulder, CO 80301
+1 303 530-1773
alexander@agentsheets.com

Andri Ioannidou
AgentSheets, Inc.
6560 Gunpark Dr. Suite D
Boulder, CO 80301
+1 303 530-1773
andri@agentsheets.com

ABSTRACT
Computer visualization has advanced dramatically over the last few years, partially driven by the exploding video game market. 3D hardware acceleration has reached the point where even low-power handheld computers can render and animate complex 3D graphics efficiently. Unfortunately, end-user computing does not yet provide the necessary tools and conceptual frameworks to let end-user developers access these technologies and build their own interactive 2D and 3D applications such as rich visualizations, animations and simulations. In this paper, we demonstrate the Agent Warp Engine (AWE), a formula-based shape-warping framework for end-user visualization.

Categories and Subject Descriptors
I.3.5 [Computational Geometry and Object Modeling]: Hierarchy and geometric transformations

General Terms
Design, Human Factors, Languages

Keywords
Real-time Image Warping, 3D Graphics, End-User Programming.

1. INTRODUCTION
Video games and the Web have been essential drivers of the incredibly rapid evolution of personal computers. Since the 1990s, visualization and networking capabilities of affordable computers have exploded, yet very little of these advancements are accessible to end-user computing. As a response, we created a framework that goes beyond regular animations to create complex visualizations and networked simulations [1]. This framework provides what we call rich *end-user visualizations* that are:

- ***End-User Accessible***. End users should not only be able to select from menus of preexisting visualizations; they should also be empowered to construct their own.
- ***Rich***. To be truly engaging, visualizations need to be rich. Crucial variables, e.g., heart rate and breathing rate in the case of a simulated human being, should be represented in a way that immerses users audio-visually.
- ***Efficient***. To be perceived as smoothly animated, visualizations need to be highly efficient.

AVI'08, May 28–30, 2008, Napoli, Italy.

2. TECHNOLOGY
The Agent Warp Engine (AWE) is a technical framework creating end-user visualizations. With AWE, end users create custom visualizations by defining 2D or 3D shapes with control points that connect to variables through spreadsheet-like formulas. Employing techniques such as shape warping, users can define sophisticated visualizations. Shape warping is a kind of image warping [2, 3].

The best way to illustrate this technology is with a demonstration. We will use examples two examples: 1) Mona Lisa: facial distortions; and 2) Mr. Vetro: a human being breathing.

2.1 Mona Lisa Example
A basic example is applying a shape warp visualization to the well-known image of Leonardo DaVinci's Mona Lisa to evoke different emotional interpretations.

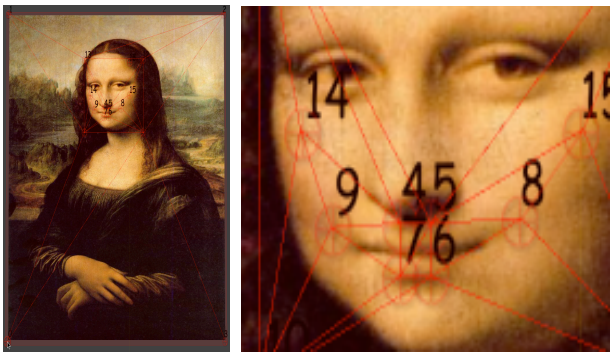


Figure 1: Left: Mona Lisa. Right: detail showing tessellated face (vertices 4, 5, 6, 7, 8, 9, 14, and 15).

The first step for the end user to define a visualization is the image tessellation. AWE includes a mesh-authoring tool that lets end users define tessellation points and triangles. A simple approach to warping our image emotionally is to focus on Mona Lisa's mouth to make her look happy, sad or neutral. A mesh around her mouth (Figure 1, vertices 4, 5, 6, 7, 8, 9) is a starting point. Additional vertices are needed to be able to define triangles covering the entire image. The key to vertex selection is controlling the scope of the desired effect. To change the mouth by moving vertices 4-9, one needs to make sure that the mouth

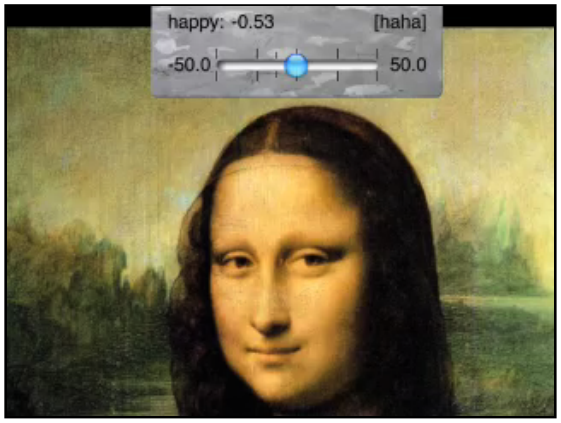
deformation does not influence too much of the remaining image. For instance, if the only other vertices were the corners of the image itself, then moving vertices 8 and 9 up to make Mona Lisa smile would also partially move the rest of the face in an unnatural way. Instead, we define vertices 14 and 15 as fixed points, roughly at the location of the cheekbones.

After defining the mesh, the end user needs to add formulas to vertices to define how the visualization takes place. The AWE mesh-authoring tool automatically creates an XML representation of the Mona Lisa mesh that includes the image reference, a list of vertices, and a list of triangles. The goal is to control Mona Lisa's emotions by adjusting the positions of the left and the right corners of the mouth. Both the x and the y attribute of the vertex are extended by the user from being constants to being formulas:

```
<vertex x="0.520 + 0.0003 * happiness" y="0.712 + 0.0003 *
   happiness" … />
```

Happiness is a user-defined variable. For happiness = 0 we get the original image. For happiness > 0 we get an increasingly happy Mona Lisa by pulling her mouth corners up and out. Finally, for happiness < 0 we have her start to frown by pulling her mouth corners down and together.

The real power of a formula-based shape warp appears when the user sees it attached to a variable controlled by a slider and experiences warping in real time. In the electronic version of the paper, the Movie 1 illustrates that.



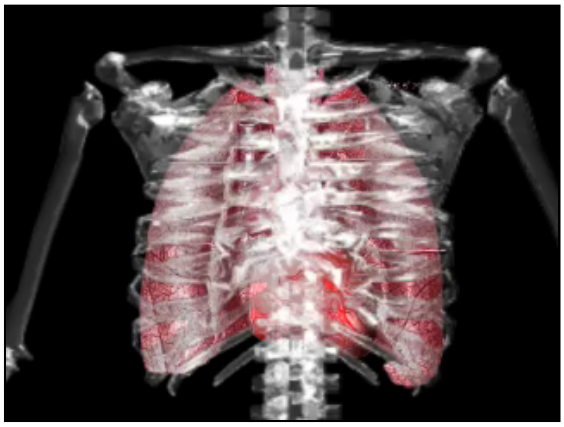
Figure/Movie 1: Changing Mona Lisa's emotions: A single variable called "Happiness" controls an image warp based on a texture map 3D shape warp.

As the user changes the value of the variable through the slider, the shape warp is recomputed and updated on the screen. Displaying the slider and the shape warp is fast; they render at about 400 frames per second on a 1.67 Ghz Mac PowerBook G4 with an ATI Mobility 9700 GPU.

2.2 Mr. Vetro Example

An important goal of this work is to offer refined kinds of visualizations necessary to communicate complex dynamic processes. For instance, in an application called Mr. Vetro, a collective simulation of a human being [1], we need to visualize the function of the heart, the lungs, and the human skeleton. All three systems mechanically interact with each other in complex ways. Inhaling air will change the shape of the lung, which in turn will influence the skeleton. Ribs expand and, in the case of deep breathing, even the position of the shoulders and arms can be influenced. AWE offers a number of visualizations, but the most sophisticated one (called "morph") is specifically designed to implement complex visualization based on shape warping.

User interface output options also include sound. To increase the immersiveness of the visualization we added inhale and exhale sounds that are triggered if the value of the distortion variable begins to increase or to decrease, respectively.



Figure/Movie 2: Mr. Vetro is breathing. Four variables are used to control breathing frequency/intensity and heart beat frequency/intensity. The two frequencies and intensities warp the combined lung and heart. The lung influences the rib cage and even changes the position of the shoulders. Nothing is pre-computed, there is no fixed animation sequence. The movie shows the physiological functions that are computed and visualized in real time.

3. CONCLUSION

Advances in computer graphics make it possible to create a new kind of application with strong visualization, animation and simulation components. The Agent Warp Engine's sophisticated 2D/3D visualizations are accessible to end users. Formula based shape warping is a spreadsheet-inspired end-user programming paradigm that can be employed for a variety of applications in need of end-user visualizations.

4. ACKNOWLEDGMENTS

This work is supported by NIH Grant 1R43 RR022008-02. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Institutes of Health.

5. REFERENCES

[1] Repenning, A. and Ioannidou, A. 2005. Mr. Vetro: A Collective Simulation Framework. In ED-Media 2005, World Conference on Educational Multimedia, Hypermedia & Telecommunications. Association for the Advancement of Computing in Education, Montreal, Canada.

[2] Wolberg, G. Digital Image Warping. IEEE Computer Society Press, 1994.

[3] Wolberg, G. 1996. Recent Advances in Image Morphing. In Proceedings of the 1996 Conference on Computer Graphics International. IEEE Computer Society, Washington, DC, 64.

Agrafo: A Visual Interface for Grouping and Browsing Digital Photos

João Mota Manuel J. Fonseca Daniel Gonçalves Joaquim A. Jorge
Department of Computer Science and Engineering
INESC-ID/IST/Technical University of Lisbon
R. Alves Redol, 9, 1000-029 Lisboa, Portugal
joao_mota_@hotmail.com, mjf, daniel.goncalves, jaj@inesc-id.pt

ABSTRACT

With the growing popularity of digital cameras, the organization, browsing, management and grouping of photos become a problem of every photograph (professional or amateur), because their collections easily achieve the order of thousands. Here, we present a system to automate these processes, which relies on photo information, such as, semantic features (extracted from content), meta-information and low level.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces - Graphical user interfaces (GUI)

General Terms

Design, Human Factors

Keywords

Image grouping, Image analysis, User interface

1. INTRODUCTION

Nowadays, due to the wide dissemination of digital cameras, any ordinary photograph reaches easily a large collection of photos. Since taking photos is almost priceless, people tend to take several similar photos, for later selection of the best one. Additionally, the sharing of photos has become an easier and more global experience, helping the growing of personal collections. This increase in the number of photos demands the need for tools to help users organize and manage their collections in an automatic way. Although, there are some approaches [3, 1] to organize photos, they use time as the main organizing principle tending to disregard other important information about photos that can only be gleaned by their contents. In this paper, we describe the Agrafo system, a visual tool to help users organize collections of photos based on its content and associated information. We use meta-information provided by the digital cameras (i.e. time), semantic information extracted from photos (presence of faces, urban/nature scene, indoor/outdoor picture) and low-level information extracted from the content of images, such as, color and texture. Users can combine several of these criteria to organize their photos, in an interactive and iterative way.

AVI '08, 28-30 May , 2008, Napoli, Italy.

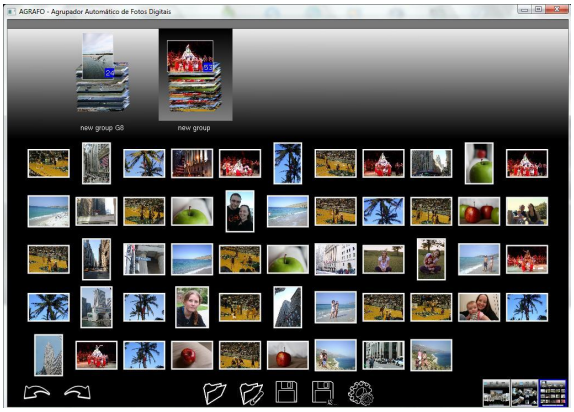


Figure 1: Agrafo user interface.

2. THE AGRAFO SYSTEM

Contrarily to other existing solutions, the Agrafo system allows the grouping and browsing of collections of photos based on their contents, using semantic information and low-level features. The semi-automatic grouping is done interactively and iteratively, through the selection of grouping criteria.

2.1 The Visual Interface

The Agrafo user interface has two main areas (see Figure 1). At the top we can see the resulting groups represented as stacks of photos, with a representative one facing the user. A small number shows the number of photos in the group. The rest of the photos are organized in perspective to give users an overview of the group content. Users can open various groups at the same time, rename groups and perform drag & drop operations to join groups or to order them. Below the group area, the system shows the photos from a group (or groups) arranged according to the selected view. Currently, we have three views available: The grid view, illustrated in Figure 1, the disorganized view (Figure 2-left) and the carousel view (Figure 2-right). Users can drag, rotate, zoom and select photos as they do in any direct

Figure 2: Agrafo visual interface and the set of photos to be grouped.

manipulation application. It is also possible to drag photos to the group area to create a new group or to add them to an existing group. Finally, users can see photos properties provided by the EXIF data and from the filesystem.

Initially, when the user opens a set of photos (e.g. from a folder) they are all placed in the same new group. New groups can be created by selecting a group or group of photos and submitting them to the automatic grouping mechanism. As a result, new groups will be created according. To choose how photos are grouped users use the Grouping Pane illustrated in Figure 3. We can select various criteria and their relative importance, allowing the fine tune of grouping. For instance, it is possible to select the Time, and Indoor/Outdoor criteria, to separate photos from a wedding into groups for the church ceremony, the photos of the spouses in the garden outside the church (taken shortly after, so using time alone would not suffice to tell them apart), and the wedding reception (later that day).

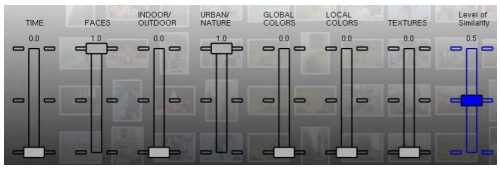


Figure 3: Grouping criteria.

The rightmost bar allows users to select the similarity between elements in a group. The bigger the similarity level, the smaller the number of photos in a group and more groups will be created. What we are saying is that we only want groups with very similar photos.

Figure 4 illustrates two of the four groups created by Agrafo after grouping the collection of photos from Figure 1, using two criteria, Faces and Urban/Nature. The system created four groups: photos with Nature scenes without Faces, Nature photos with Faces, Urban images without Faces and Urban photos with Faces. When users select criteria that can lead to "strange" combinations, such as, Urban/Nature and Indoor/Outdoor, the system only creates those groups that make sense (Indoor, Outdoor+Nature and Outdoor+Urban).

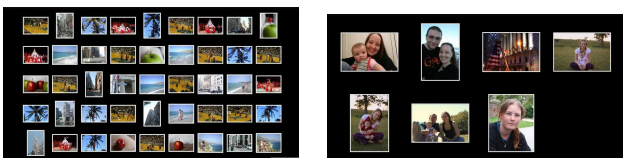


Figure 4: Photos from Nature without Faces (left) and with Faces (right), after grouping by Urban/Nature and Faces.

2.2 Grouping Mechanism

The automated grouping mechanism was implemented using the QTClust clustering algorithm [2]. While this is more computational intensive than the more widely known k-means algorithm, it has the advantage of not require the number of clusters beforehand. In Agrafo, we can not provide that number, because it will depend of the set of photos and of the criteria selected by the user.

The QTClust algorithm requires a distance function to tell how close photos are from each other. The more similar the photos are, the closer they will be. To measure that similarity, our algorithm uses the different criteria specified by the user. It is important to notice that only the specified criteria are used during the clustering operation. Each criterion corresponds to a different dimension, thus by using only the selected criteria we are creating smaller dimension spaces, which optimizes the clustering process.

To optimize performance, our system computes features for each criteria in background and stores those features for next executions of the application.

Currently, Agrafo can group photos using the following criteria: Time, extracted from EXIF metadata; semantic information, such as, presence of Faces, Indoor/Outdoor and Urban/Nature scenes, detected from image content; and low-level features, namely Global color, Local color and Textures, extracted from photos. However, and since our architecture is modular, the inclusion of new criteria is very easy.

3. CONCLUSIONS

In this paper we shortly describe a system to support users in their task of grouping and browsing collections of photos, through the selection of features extracted from images content. It has a simple and easy to use visual interface, which allows automatic and quick grouping of photos.

Currently we are preparing an experimental evaluation of the Agrafo system, to check if the resulting groups are similar to what users will do by hand. To that end we are collecting photos from users, already organized in groups, to serve as test bed.

4. ACKNOWLEDGMENTS

This research was sponsored in part by Portuguese Science Foundation grant DecorAR-POSC/EIA/59938/2004.

5. REFERENCES

[1] M. Cooper, J. Foote, A. Girgensohn, and L. Wilcox. Temporal event clustering for digital photo collections. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(3):269–288, 2005.

[2] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genoma Research*, 9(11):1106–1115, Nov. 1999.

[3] D. Kirk, A. Sellen, C. Rother, and K. Wood. Understanding photowork. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 761–770, New York, NY, USA, 2006. ACM.