Digital Library Service Models and NODAL

Lee Iverson
Dept. of Electrical and Computer Engineering
University of British Columbia
Vancouver BC, Canada
leei@ece.ubc.ca

ABSTRACT

New classes of services for digital libraries have been explored in the research community with many now being tested and deployed in real-world settings. Unfortunately, there have been problems, some predicted and some unforeseen, in the development of these services. Moreover, a number of problems have been identified as becoming critical in the future, especially the difficulties associated with preservation of archives of digital content when data formats and media both have limited lifespans.

In this paper, we consider a number of these services (personal collection management, annotation, institutional repositories, and learning object repositories) in terms of their user and task requirements and develop a unified conceptual model of how these services should work together. This analysis reveals a set of technical requirements that highlight a number of critical gaps in the conceptual integration of the services and in the infrastructure underlying their current implementations.

Finally, we consider current potential unifying technologies and suggest that the best hope for a solution that meets both user and technical needs lies in a combination of integrated services built on a document modelling and collaboration infrastructure that we call NODAL, the Network-Oriented Document Abstraction Language. We will attempt to illustrate how this model may enable even the most ambitious visions of the potential of the digital library.

Categories and Subject Descriptors

D.4.3 [**File Systems Management**]: Distributed File Systems; H.2.4 [**Systems**]: Distributed Databases; H.3.7 [**Digital Libraries**]; H.5.4 [**Hypertext/Hypermedia**]: Architectures

Keywords

Digital Libraries (primary keyword); Computer Supported Cooperative Work (CSCW); Database access / Information Retrieval ; World Wide Web and Hypermedia ;

1. INTRODUCTION

Digital libraries are moving into a new phase of development. Many of the inherent problems of collection digitization, distribution, rights management and the changing of mindsets to recognize the inherent advantages of digital versus paper collections have been achieved. In that context, there is now an increased interest in exploring new service models and means of enhancing digital libraries for their patrons that would be impossible to achieve with physical collections.

Unfortunately, many of these efforts are being pursued quite separately from each other and with more of a focus on the new technology than on the ways these systems interact with their users' needs and tasks. And even if each of these efforts were pursued with a full human-centered, participatory design process their common assumptions and infrastructure needs could still be hidden from their builders. In order to provide some alternate direction and hopefully expose these commonalities, we have chosen to examine usage scenarios for a number of these services, namely personal collection management, annotation, learning object repositories and institutional repositories. Using these scenarios, we then developed a high-level conceptual model that identifies common objects and relations and defined a set of operational requirements for applying these objects to enable the scenarios described.

Finally, we suggest that a new application model and storage infrastructure will provide exactly the substrate on which to integrate the delivery of these services. Moreover, this integration is exactly what is necessary to enable the easy, seamless access to library services that is assumed in the scenarios described. Without this seamlessness and integration with actual useful tasks even great effort and impeccable design may doom these services to the kind of failures of other groupware applications described by Grudin[16] and Ackerman[3].

2. SERVICE SCENARIOS

We explored a number of proposed or deployed categories of digital library services based on a consideration of user tasks that would require those services. To illustrate the work, we have selected a number of scenarios for each area, focusing on certain tasks for three classes of users in an academic library setting: researchers, teaching staff, and stu-

dents. In all cases, we developed scenarios without explicit reference to the capabilities and limitations of existing tools and services, concentrating instead on capturing "natural" task-specific behaviors and expectations. The scenarios we have chosen to present here were selected to provide sufficient coverage to ground all of the elements of the conceptual model developed.

2.1 Personal Collection Management

Personalized collection management has been proposed [36, 4] and widely deployed in some circumstances (e.g. MyLibrary[28]). It has proven to be a usable and reliable service and the foundation of a number of attempts to explore more general models of personalized services in digital libraries. Some of the scenarios developed in this context are:

1. A *researcher* is searching the Web (e.g. using Google or CiteSeer) and her institution's online collections and wants to preserve a bibliographic link to some resource in his personal collection. Much of the metadata for the entry is automatically extracted from the data source. The resource is categorized based on both reference metadata and augmented with contextual criteria provided by the search parameters and the researcher himself. The bibliographic record becomes immediately available to her preferred bibliography management tools. The researcher is able to provide various views of the collection to selected collaborators.

2. A *teacher* is preparing a lecture and uses library resources to find a number of readings for his students which are preserved in his personal collection. In order to provide these to his students he creates a view of the resources for that particular lecture that is accessible to his students and creates a link to this view in his online lecture notes. Students following this link have no access to other parts of the teacher's personal collection, and only students registered in the course have access to the contents of the link.

3. A *student* seeks information using the Web, instructor-provided links and both paper and digital library resources to write a term paper. He enters the references into his personal collection and uses URL references to these collection items to provide hyperlinks in his paper. A reference list is automatically produced by the collection management system that the student can copy and paste in HTML into his bibliography.

The conceptual model from these examples (and others) highlights the need for objects such as Documents, Collections, Personal Collections and Sub-Collections, Search Contexts, Metadata, Bibliographies, Data Formats and Access Control. Aside from the Access Control, each of these objects must provide URL-based access and seemingly in a variety of formats, including at least HTML, BibTeX and EndNote. Of course any of these Collections must provide efficient search facilities on both content and structure. Metadata is produced with a combination of direct reference, context-dependent automatic extraction and user-provided context and verification. Categories are either contextual, user-defined, search-based, subject-dependent or some combination of these.

2.2 Annotation

Annotation of documents has been identified as one of the most significant ways in which we record the interpretation of documents (both for personal use and communication of ideas) and as one of the most important challenges in taking greater advantage of the digital document paradigm [26, 27]. It was one of the hallmarks of early hypertext visions and systems (e.g. Bush's Memex[5] and Engelbart's NLS[12]) and a foundation of pre-Web hypertext[18, 37, 15, 17]. Certain attempts have been made to provide annotation facilities for remote Web-accessible resources, the most interesting ones allowing annotations of a variety of document formats with category tags [30, 6, 35]. Of particular note is the Anchored Conversations project[7], in which interactive, conversational annotations are embedded in digital documents. From these foundations we consider a number of scenarios, including:

1. A *researcher* reviews an article for publication. She makes both personal and general critical observations on particular sections of the text and writes a review based on these, making reference to certain entries in his personal collection. The text is then revised and reworked by the authors and sent back to the reviewer who considers the changes from the old text to the new, his previous annotations and review and decides that the paper is now ready for publication. When she is electronically notified of the published work, she enters it into her personal collection, moves the personal annotations from the draft versions and notifies her graduate students of the new paper along with her personal notes.

2. A *teacher* receives an email question from a student and makes an annotation on a document in his personal collection that answers the student's question. He emails the annotation reference to the student who reads it. An electronic chat conversation between teacher and student ensues which is recorded and added to the annotation. The teacher categorizes the annotation with reference to the student's question and saves it for future reference, at the same time making it available to the rest of the class.

3. A *student* sends a question to a teacher and receives an annotated document reference from the teacher in answer. He reads the annotation and its source, adds them both to his personal course notebook and saves the annotated document reference in his personal collection.

These scenarios add to our model the Annotation and Review objects, along with similar Access Control and URL-based access requirements as for personal collection management. Annotations are interesting objects however with Categories and Conversational components and the ability to migrate through versions of a Document. One clear requirement, implicit in these scenarios, is the need to be able to annotate any kind of document in any format and to manage the annotations separately from the documents.

2.3 Learning Object Repositories

Learning Object Repositories (LORs) have become a focus of much excitement and research in pedagogical communities, since "object repositories are seen as key enablers for bringing increased value to learning resources by providing opportunities for reuse, repurposing, or reengineering to suit a variety of purposes and end-user needs. Creating learning resources in object formats is seen as way to bring about increased flexibility, customization, ease of update, searchability, and manageability to rich stores of content and learning resources that are available from publishers or that have been created by faculty members or teachers."[31] In essence, since teachers in many contexts are migrating towards the use of electronic media and teaching tools (learning objects), it makes perfect sense to archive and share these resources both with other instructors and their students. Early experiences with these repositories has, however, been mixed with successes associated with good central planning, incentive programs and substantial expenditures of resources [31, 2, 24] and failures attributed to problems with one of these three requirements or failures to achieve good work/benefit balance or critical mass[16]. Some of the working scenarios that illustrate potential interactions with these repositories are:

1. A *researcher* needs to learn a new technique for working in her lab. She searches a federation of LORs to find a good tutorial on the use of that technique. After working through the exercises, she makes notes about the applicability of this technique for use in her lab and passes the notes and the learning object on to her lab assistants. Together they adapt the technique and begin using it. As a professional courtesy, she forwards these notes back to the author of the learning object.

2. A *teacher* prepares lecture notes and learning objects for a course with both local and remote students. He delivers the lecture with a webcam link and fields questions from both students in class and via a textual chat system. She responds to some of these questions by making reference to the lecture notes, a video and demonstrates a principle with one of the active learning objects he has prepared. At the end of the class, the entire recorded session is encapsulated in a new learning object and uploaded to the university's repository with metadata derived automatically from the course description, syllabus and lecture notes.

3. A *student* explores a learning object provided by his instructor. He asks a number of particular questions about particular aspects of the object by creating annotated references and sending these to the learning object's author. These are answered by the author by augmenting the annotations with text and references to other learning objects. The author, with the student's permission, adds these annotations to the published version of the learning object.

From these scenarios we need to expand our conceptual model only slightly, considering that all of the objects and services necessary to accomplish these tasks have already been discussed except for the learning objects themselves. An LOR can simply be characterized as a specialized Collection with multimedia object that may or may not be active. It is important to note however, that in these LOR scenarios, the repository is largely managed as a resource. The social and institutional problems of populating an LOR is highlighted by the imbalance in these scenarios between use and augmentation of the repository. In the *teacher* scenario, the work/benefit balance is achieved by *automatically* creating and indexing a learning object from the context and interactive record of the classroom session. Without these kinds of tools, it is an enormous burden for teachers to actually create effective learning objects themselves.

2.4 Institutional Repositories

An Institutional Repository (IR) is a digital library collection that records all of the scholarly output of an academic institution[1]. It is assumed that as scholars prepare their work, they will first place publishable content into an IR and then submit it for further review and the imprimature of "publication" by a journal or conference[10]. In concept, they are natural follow-ons to the success of preprint archives in transforming the nature of academic publishing in Physics and Mathematics. Some simple IR scenarios:

1. A *researcher* completes a paper. She uploads it to her University's IR and provides the necessary metadata, including categories, authorship, publication status etc. The paper is automatically incorporated into her online CV and personal web page. She then submits the IR reference to a conference and is accepted with revisions. She revises the paper and submits camera ready copy to the IR as a new version, updates the paper's status and forwards the IR reference to the conference organizers. After the conference, she receives weblog trackback pings from a number of attendees who have comments on the paper on their personal logs. She responds to these by creating annotations in the paper with reference to and responses to some of the comments in the weblog entries. These annotations are made accessible to the weblog authors who continue some of the conversations in the context of the annotations. Some of these annotations are made available to the public accessing the paper via the IR.

2. A *teacher* introduces his students to a number of papers from IR sources that illustrate certain principles in class. The teacher prepares a summary of the class discussion on the course email list and augments it with annotations of the documents pertinent to the topics being discussed. He forwards the email to the authors of the documents and engages in discussion with them on the points made by his students. One or some of the authors make the annotations available on the IR.

3. A *student* composes a paper using documents from a number of IRs (found using Google) as primary sources. The paper contains quotes that are described using hypertextual references to individual phrases in the documents. The student's instructor uploads the paper to the course website (and thus automatically makes it available to a spider that populates the University's

LOR). Trackback-like notices are then sent to the authors of the original papers and the student paper is automatically added to the citation lists for those papers.

Again, we have few new concepts introduced other than a great expansion of our understanding of the potential of these annotations and hypertextual cross-references. We can also clearly see that early assessments of the potential of IRs for revolutionizing academic communication may, in fact, be understated. One point that is becoming very clear though is that institutional boundaries are almost certainly insufficient in managing selective Access Control to these resources. There are very good reasons to allow external users other than read-only access to the archives (e.g. in the conversational annotations). We almost certainly require a flexible, distributed authentication scheme that communicates identity accurately and in a trustable manner, such as Shibboleth[14].

There are a few caveats though. Academics are assumed to be the source for all of the material in these archives and often the source for the metadata as well. In many cases, plans for institutional repositories assume that if the technical problems, usually in terms of ease-of-use, are solved then academics will willingly enter content into the IR. Unfortunately, no matter how easy to use these systems are, they are more work for already overworked academics, and provide no immediate benefit to the contributor, thus running afoul of Grudin's [16] work/benefit imbalance and predicting failure to reach a critical mass of user-contributed content.

One solution to this problem is to offload the work from the academics to support staff, a strategy that has been undertaken at some of the pioneering institutions such as MIT and University of Toronto. Unfortunately, this strategy is very expensive and error-prone with data entry and often metadata construction being performed by people who don't have the subject-expertise necessary to select categories or choose collections. Automatization of metadata construction is, of course, still an open research topic and one which may depend on the solution of "grand challenge" problems in natural language processing and artificial intelligence.

So, at this point in time, in assessing the potential and impact of these institutional repositories, Richard Johnson [23] has pointed out the need to clearly understand the social context surrounding academic publication that may determine the success or failure of these projects. Since academics, the sources of content for the IRs, have high stakes (e.g. career advancement) dependent on the recognition of the impact of their published work, it is fundamental to the success of the IRs that they are rewarded for contributing meaningful content.

3. OTHER ISSUES: PRESERVATION

An Associated Press story from January 2003 summarizes one of the greatest problems with the preservation of digital archives: the outdating of data formats and digital media. He quotes a Joe Miller, a USC Neurobiologist who was unable to extract information from the tapes recording data sent back from the 1976 Viking lander on Mars. "All the programmers had died or left NASA," Miller said. "It was hopeless to try to go back to the original tapes." One of the projects attempting to resolve this issue is the InterPARES Project [19]. They have highlighted the need to preserve both data and software and hardware to recover it. One suggestion they make is to provide a permanent archive of software for reading data formats and associating this with the preserved data records themselves.

4. CONCEPTUAL MODEL

It should be fairly obvious that there is a single, fairly well-defined conceptual model that allows us to present any of these objects and services with a common structure and set of facilities. Each of the service classes described above can be presented and manipulated with the same set of basic primitives.

4.1 Object classes

A set of kinds of objects that must be represented in these digital repositories.

Document Some structured collection of data with defined boundaries. Named and associated with a particular data format and metadata records.

Data Format Some translation from the structured data that is a Document into a data stream that can be stored or transmitted.

Collection (including Personal Collections, Sub-Collections and Bibliographies) Some defined collection of Documents. May be defined by user, category, search context, metadata tag or context of use.

Search Context A description of search parameters that can be used to define a sub-collection or reference to an external search facility such as Google.

Metadata (including Categories and Usage Contexts) Data about Documents and Collections. Contains both intrinsic metadata, inherent to objects such as type information, ownership, and object history as well as extrinsic metadata such as categories and contexts of use. With efficient search facilities, it may be possible to store such extrinsic metadata in a way that simply references the content that it applies to (such as in RDF metadata models).

Annotation Snippets of structured data (or references to services such as with anchored conversations) that reference other content. It should be possible to annotate any document, collection or parts thereof with almost any kind of content.

User Identity It is clear that subscribers to library services need to be individuated (how else could we personalize?) but it may not be so obvious that we need to respect identities of external users that may be simply accessing the services as well. It may be simpler initially to ignore this, but with it may eventually be possible eventually to allow collaborative access to external users identified by some trusted identity broker (e.g. by Shibboleth[14].

Access Control Many of the scenarios described above implied a sophisticated user-managed access control to content, with the ability to delegate limited access to other users, identified groups, or based on category membership. In addition, there are likely to be usage policies and restrictions applied by the libraries themselves that limit this flexibility in certain ways.

4.2 Other Requirements

Aside from the above object classes, there are some general requirements that apply across classes.

URI-based reference A number of the scenarios make it clear that most if not all objects in the system should be uniquely identifiable by URL or URN.

Granularity for annotation and reuse If annotation is to be built on top of a URI-based reference system, then URIs must be able to select document fragments in a meaningful way. If we can only address fragments for certain classes of document, then we will be very restricted in how useful those annotations can be. For example, if we are unable to annotate a scene in a movie, then much scholarly discussion of motion pictures would be difficult to support.

Synchronous and asynchronous interaction The scenarios above highlighted both synchronous and asynchronous interaction between users and repositories and between users and other users. While we may not want to directly support synchronous, user-user communication in a digital repository infrastructure (relying instead on simply providing introductions to external communication services), we would be missing a great opportunity to facilitate agreement. There is much evidence that the formation of mutual knowledge[8] seems to depend on shared context and experiences and thus on synchronous communication modes[34].

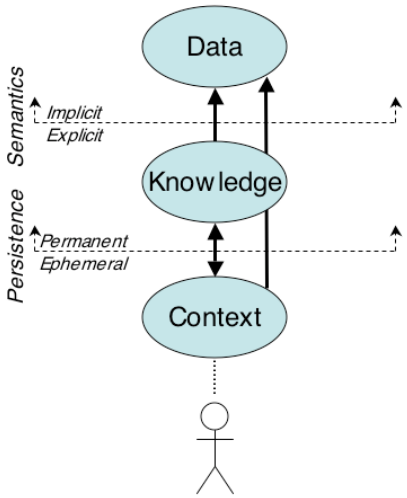
Integration with existing tools The scenarios seem to assume a seamlessness in which the users can move back and forth between tasks of immediate concern and interacting with library services. In essence, we are arguing for a kind of digital library that can interact seamlessly with a user's existing tools and desktop. To do this, we will need to respect the interaction and storage assumptions of those tools and make all efforts to integrate them with the library infrastructure.

4.3 Application Models and Technologies

We suggest that one way of understanding how we can apply this conceptual model to the development of real digital library services is to rely on a layered application development model we refer to as the DKC model[22], for Data, Knowledge and Context1. The Data layer is an operating system or middleware layer that handles data storage, communication and security for persistent data, such as a distributed operating system or database. The Knowledge layer is a semantic layer in which explicit semantics are used to organize information in the data layer and provide services to the users, for example object models and Semantic Web services would both be embedded at thus layer both in generic and task-specific forms. The Context layer is equivalent to the client application, managing user and task-specific activities and views on the reliable data storage and semantic infrastructure provided below.

This model is most closely related to the three-tier architecture from enterprise-level information systems with Data, Business Logic and Presentation closely related to the DKC layers. There are a number of conceptual distinctions though, with the Knowledge layer in DKC assumed to be managing both declarative and procedural semantics as well as meaningful information filters from the Data layer. In addition, the DKC model specifically allows direct Context - Data interactions instead of assuming that the Knowledge layer is always mediating. Finally, the DKC model is intended as a universal application model with very specific requirements for the Data layer, which we outline below. It does however, retain the extreme flexibility and adaptability of front-end applications all built on a common data architecture that has come to be recognized as one of the main advantages of the three-tier model.

Figure 1: The DKC Model, a three-layer model of Data, Knowledge and Context layers. The Data and Knowledge layers store persistent, structured information and provide services. The Knowledge and Context layers are semantically rich with actions and stored knowledge defined in terms of meaningful operations to support a variety of tasks. The Context layer stores ephemeral data and is responsible for user interaction and knowledge mediation and discovery.



Given this application model, it is clear that many of the elements of the conceptual model above lead primarily to constraints on the data layer. If we have a sufficiently rich and flexible data layer, then the difficulty of implementing the knowledge and context layers for a particular "application" or "service" may be significantly reduced. In a certain respect, it is thus our goal to define just what such a rich and flexible data layer must include:

Data Model The observations from experience with enterprise-level architectures, shared virtual worlds and even mod-

ern dynamic web sites suggests that the best foundation for sharing and flexible reuse of data is a general and flexible data model.

Database View Given the emphasis on a structured, constrained data model, it should be clear that most of the basic interactions with the model must be managed exactly as they are in a modern, multi-user transactional database. In fact, we suggest that implementing a data layer on top of either an object database or relational database is the best option for bootstrapping such a system.

Filesystem View We have pointed out clearly above that it is not a luxury or afterthought to support existing applications, work patterns and data formats. As such, it will be essential to maintain a connection between the database view and a distributed filesystem view. For this reason, we suggest that a Data layer must have some extensible architecture for relating data models and formats and that the availability of these crosswalks be made explicit and public. Moreover, the Data layer should be able to incorporate and provide facilities for accessing remote filesystem-like resources (e.g. NFS, HTTP, IMAP, etc.) with the same APIs and modeling tools available on the database side.

Asynchronous and Synchronous Interaction Since effective work patterns and communication for collaborative activities involves both synchronous and asynchronous communication, and end-user devices may move on and off the network, it will be necessary to provide means of interacting with both the database and filesystem layers in both synchronous and asynchronous modes. This would potentially mean that when connected, a user could be interacting with an interactive, shared data model in an effectively shared environment with any other connected users of a data repository. When offline, the user should still be able to work, but with an interaction mode that is queued and then synchronized when eventually reconnected.

Granular Reference We argued that granular reuse of data was important. Moreover, the key enabling technologies for Semantic Web services rely on the ability to reference content within files. If we hope to bring knowledge management and hypertext capabilities to all data formats (as we stated above) then it is clear that we need to provide some sort of universal, granular referenceability of content *inside* arbitrary files.

Change Auditing and Synchronization In order to enable the orderly transition from disconnected to connected use and resynchronization of resources, it is necessary to have some kind of change tracking and then synchronization methods. We suggest that the flexible data model and granularity of reference can provide a mechanism that can allow for universal change tracking and synchronization for any kind of document format via identification of data models and independent history auditing for all data model objects within documents.

Flexible Security and Privacy Clearly providing a means for users to flexibly manage the trust environment for allowing others to reuse and even manipulate their data resources is essential for effective sharing and collaboration. Moreover, we suggest that the appropriate granularity for this security management is at the level of the sub-document objects within the data model.

Search for Content and Structure Clearly for purposes of data mining and information management, providing a universal search infrastructure is essential. Moreover, we suggest that this searchability be presented at both content and structural levels.

One possible candidate for this data layer is an XML databases[32]. All we would need is to require that all data to manipulated by our library be expressed in XML and then use a federation of XML databases as our digital library infrastructure. In order to support non-XML data formats, we simply convert them to XML and be done with it. Unfortunately, this is probably not a feasible solution. It's success would depend critically on the universal adoption of XML by vendors and developers. Moreover the need to satisfy many different parties in the standardization processes around XML have resulted in XML-based standards that are extremely complicated and generally unmanageable by any but the largest corporations or communities of developers.

A case in point is the widely-perceived complexity and unusability of the current XML Schema: Part 1 (Complex Datatypes) standard[33]. This has been characterized as a conflict between the data-oriented (i.e. database management) and text-oriented (i.e. SGML and HTML) communities and the difficulty of resolving their often competing requirements[9]. We would like to suggest that in addition to these difficulties is the fundamental confusion embedded in the XML standards between data models and syntax. XML, as flexible and adaptable as it may be, is still a markup language, a syntax designed primarily for "marking-up" primarily textual documents. It has been adapted with mixed success to the goal of representing and exchanging fundamentally non-textual data resources (e.g. XML-RPC and SOAP), but it retains its roots as a language for expressing text. We suggest that a step beyond XML would clearly separate data model from the (possibly numerous) languages for expressing that data model and build a collaborative foundation around that. This is exactly what we have done with the NODAL architecture described below.

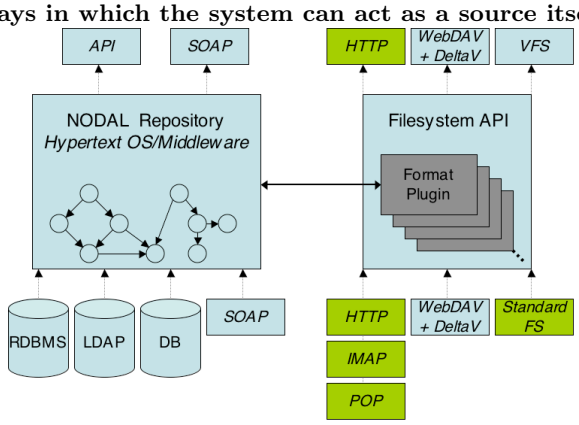
5. THE NODAL ARCHITECTURE

Inspired significantly by the historical visions and successes of Vanevar Bush [5] and Douglas Engelbart [12, 13] we have developed an architecture and system that fulfills all of the requirements outlined above. Inspired significantly by much more recent work, especially the Multivalent Browser [29, 30] and the Xerox Presto architecture [11] we have developed a system that supports both database-like access and a flexible, overlayable filesystem model thus supporting the development of new, richly collaborative applications without leaving old tools or old filesystem models behind.

NODAL[20, 21], the Network-Oriented Document Abstraction Language, is designed as an extensible data storage and communication middleware layer. Presenting itself (see Fig. 2) as both a database (via SOAP or the intrinsic API)

and as a filesystem (via HTTP+WebDAV or as a virtual filesystem on certain systems) and able abstract the data stored in a variety of databases and filesystem types (including network filesystems such as HTTP+WebDAV and IMAP). The key to this dual nature is a schema language for the data model and a related system of data format plugins that encode and decode from the data streams handled by the filesystems and the objects represented with the data model. In this way, we allow seamless integration of new systems built on the database APIs and old applications that interact only with filesystems.

Figure 2: NODAL System Architecture, centered around a document-oriented data model and API, it is accessible either in filesystem mode or in a database mode. The modules on the bottom are data sources for the system and those on top are ways in which the system can act as a source itself.



5.1 The NODAL Data Model

The foundation of NODAL is a data modelling language that is designed to provide a sufficiently rich environment to adequately describe the internal structure of all document types. Moreover, the API is designed so that applications (context layers) may be built directly on top of the API, using the database as a seamless backing store for application objects.

The data model begins with a standard set of atomic types, similar, but smaller, than the set described in XML Schema: Part 2. It includes such value types as character, integer, float, timestamp and name.

5.1.1 Node Types

Each document is modelled as a graph of nodes, each of which has a particular type. These node types can be seen as a minimal set of collection types. Their role in the NODAL model is fundamental, as they not only form the building blocks for the document graphs but are also the fundamental units of addressability, auditing and security. In essence, each node is the minimal unit of modifiable data in the model. Significantly, each node is directly accessible by a URL. There are three types of nodes: the record, the sequence and the map.

5.1.1.1 Record

A record is a node that represents a collection of named values. Variously called a "class," a "struct," or a "table" in relational databases, it is a familiar building block for structured data. In abstract terms, a record type is a set of mappings from names of fields to data types.

5.1.1.2 Sequence

A sequence is an ordered set of values of like type. Also known as a "list", a sequence is the basic building block for ordered data. A sequence type consists of an itemtype, which constrains the contents of the sequence. As we noted previously, in NODAL, a string is a represented as sequence of characters.

5.1.1.3 Map

A map is a collection which maps values of one data type to values of another. Also known as a "dictionary," it is an extensible association between objects. Unlike a record, a map type has a single key type and a single value type. For example, an XML attribute list is a map from names to strings.

5.1.2 Documents

Every NODAL document has a well-defined type, identified by a MIME type. Associated with that MIME type is an encoder/decoder pair (available through a plugin) and possibly a document schema if existing types cannot be reused. This schema consists of a set of type declarations that are used to build the document. Type declarations can be expressed in XML and provide a simple means of defining document types (see Fig. 3 in which we model a text file as a sequence of strings). More complicated document types have much more complicated schemas and data format plugins, but they still only produce a graph of nodes as the document. Directories are in fact simply modelled as a kind of Document that maps names to Documents.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE schema SYSTEM "types.dtd">

<!-- A NODAL schema for a plain text file. -->
<schema xmlns:ndl="baseTypes.nls">
  <sequence name="lines" itemType="ndl:String"/>
</schema>
```

Figure 3: The complete schema for plain text files

To achieve the granular addressability required above it is necessary to appreciate that each of the three types of node can be abstracted as a set of property/value pairs (sufficiently constrained). Fragment URLs are derived by appending the document name with a path from the root node of the document down to the fragment in question. For example, a plain text file identified by the URL `http://nodal.sf.net/readm` we can access the fifth character on the second line with the URL `http://nodal.sf.net/readme.txt#/1/4` (sequences have origin 0). In fact, there is an extensive path language with various operators on the path available (e.g. we can reference the first 20 characters of the second line using a `range(0,19)` operator).

It may not be perfectly clear from this short summary, but the NODAL model has been designed to fulfill all of the requirements described above.

6. CURRENT STATUS AND TESTING

The first prototype implementation of the NODAL middleware layer is being implemented in Java and is available for download from Sourceforge. The NODAL data modelling environment and API is nearly complete, with only the security and privacy protection completely unimplemented. It currently supports data input from a local filesystem or HTTP and has integrated support for plain text, HTML and XML files. With this infrastructure, we have been able to test the applicability of the data modelling framework to a variety of other formats and demonstrate its generality. In addition, we are nearly completed a persistence layer that will allow us to use a relational database as a read-write data repository.

Using this infrastructure, we have begun to experiment with a generic editor for any document type and a note-taking application that builds RDF models. The first major project we plan to undertake is a reconsideration of the email client as a relationship management tool. The NODAL middleware model will be essential in allowing us to interconnect mail messages with RDF-based category and ontology development tools that will allow for much more effective contextual search. We are also exploring the use of NODAL as an implementation layer for some semantically rich digital collection management tools.

7. IMPLICATIONS

To reiterate, we have developed a simple conceptual model that integrates at least the four different classes of advanced services described above. We have done this by exploring a large set of user- and task-oriented scenarios and By ignoring the assumed models and limitations of existing implementations in developing our scenarios, we have clearly outlined a set of integrated facilities that could be the foundation for the kinds of advanced interactive library-hosted collaboration environments described in Marchionini's Sharium[25] and maybe even a step closer to Bush's Memex[5].

8. REFERENCES

[1] The DSpace federation. `http://dspace.org`.

[2] S. Acker, D. Pearl, and S. Rissing. Is the academy ready for learning objects. *Syllabus*, 2003.

[3] M. Ackerman. The intellectual challenge of cscw: The gap between social requirements and technical feasibility. *Human-Computer Interaction*, 15:179–203, 2000.

[4] R. D. Burke. Salticus: guided crawling for personal digital libraries. In *ACM/IEEE Joint Conference on Digital Libraries*, pages 88–89, 2001.

[5] V. Bush. As we may think. *Atlantic Monthly*, 176(1):101–108, 1945.

[6] J. J. Cadiz, A. Gupta, and J. Grudin. Using web annotations for asynchronous collaboration around documents. In *Computer Supported Cooperative Work*, pages 309–318, 2000.

[7] E. F. Churchill, J. Trevor, S. Bly, L. Nelson, and D. Cubranic. Anchored Conversations: chatting in the context of a document. In *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems*, pages 454–461, The Hague, The Netherlands, 2000. ACM Press.

[8] H. H. Clark and C. R. Marshall. *Definite reference and mutual knowledge*. Cambridge University Press, 1981.

[9] M. Classen. Schema wars: Xml schema vs. relax ng. `http://www.webreference.com/xml/column59/`, 2002.

[10] R. Crow. The case for institutional repositories: A SPARC position paper. `http://www.arl.org/sparc/IR/ir.html`, 2002.

[11] P. Dourish, K. Edwards, A. LaMarca, and M. Salisbury. Presto: An experimental architecture for fluid interactive document spaces. *ACM Transactions on Computer-Human Interaction*, 6(2):133–161, 1999.

[12] D. C. Engelbart. The mother of all demos. `http://sloan.stanford.edu/mousesite/1968Demo.html`, 1968.

[13] D. C. Engelbart. The unfinished revolution: Strategy and means for coping with complex problems, 2000.

[14] M. Erdos and S. Cantor. The Shibboleth architecture. `http://shibboleth.internet2.edu/`.

[15] A. M. Fountain, W. Hall, I. Heath, and H. Davis. MICROCOSM: An open model for hypermedia with dynamic linking. In *European Conference on Hypertext*, pages 298–311, 1990.

[16] J. Grudin. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, 37(1):92–105, 1994.

[17] F. Halasz and M. Schwartz. The Dexter hypertext reference model. *Communications of the ACM*, 37(2):30–39, 1994.

[18] F. G. Halasz. Reflections on notecards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31(7):836–852, July 1988.

[19] InterPARES. *The Long-term Preservation of Authentic Electronic Records: Findings of the InterPARES Project*. 2001.

[20] L. Iverson. Nodal: A filesystem for ubiquitous collaboration. `http://nodal.sf.net/NODAL-WhitePaper.html`, 2001.

[21] L. Iverson. NODAL: A network-oriented data abstraction language. 2004 (in preparation).

[22] L. Iverson. The DKC model: an application model for collaborative work. 2004 (submitted).

[23] R. K. Johnson. Partnering with faculty to enhance scholarly communication. *D-Lib Magazine*, November 2002.

[24] B. Lamb. How can we avoid the pitfalls of learning objects and use them effectively instead? `http://www.e-strategy.ubc.ca/news/update0311/031126-elo.htm` November 2003.

[25] G. Marchionini. Augmenting library services: towards the sharium. In *ISDL 99*, 1999.

[26] C. C. Marshall. Annotation: From paper books to digital library. In *ACM DL*, pages 131–140, 1997.

[27] C. C. Marshall. The future of annotation in a digital (paper) world. In *35th Annual GSLIS Clinic: Successes and Failures of Digital Libraries*, Univ. of Illinois at Urbana-Champaign, 1998.

[28] E. L. Morgan. Mylibrary@ncstate: The implementation of a user-centered, customizable interface to a library's collection of information resources. In *Proceedings of SIGIR '99*, 1999.

[29] T. A. Phelps and R. Wilensky. Multivalent documents: Inducing structure and behavior in online digital documents. In *29th Hawaii International Conference on System Science*, 1996.

[30] T. A. Phelps and R. Wilensky. Multivalent annotations. In *European Conference on Digital Libraries*, pages 287–303, 1997.

[31] D. Porter, J. Curry, B. Muirhead, and N. Galan. A report on learning object repositories: Review and recommendations for a pan-canadian approach to repository implementation in canada. Technical report, CANARIE Inc., March 2002.

[32] A. Schmidt, M. Kersten, M. Windhouwer, and F. Waas. Efficient relational storage and retrieval of XML documents. *Lecture Notes in Computer Science*, 1997:137+, 2001.

[33] The World-Wide Web Consortium. Xml schema: Part 1 (structures). `http://www.w3.org/TR/xmlschema-1/`, 2001.

[34] S. Whittaker. *Theories and Methods in Mediated Communication*. MIT Press, 2003.

[35] R. Wilensky. Digital libraries resources as basis for collaborative work. *Journal of the American Society of Information*, 5(3), February 2000.

[36] R. Wilensky. Personal libraries: Collection management as a tool for lightweight personal and group document management. 2001.

[37] N. Yankelovich, J. B. Haan, N. Meyrowitz, and S. Drucker. Intermedia: The concept and the construction of a seamless information environment. *IEEE Computer*, 21(1):81–96, 1988.