

On Developing a Simple In-house Digital Library Archive

Eric L. Brown, Luis G. Velazco, G. Kirksey, S. Ramaswamy, M. Rogers
Software Automation and Intelligence Laboratory, Computer Science Department

Tennessee Tech University, Cookeville, TN 38505

elbrown@acm.org, lgvelazco@csc.tntech.edu, gkirksey@tntech.edu, srini@acm.org, mrogers@csc.tntech.edu

ABSTRACT

Digital libraries (DLs) have become a major focus of information technology resources for organizations from academia, to the US Department of Defense to the archeological efforts in Egypt. DLs have transitioned from theoretical exploration to support infrastructure development. The representation of multimedia streams that are both client-acceptable and compliant with digital rights models is currently in an “amoebic” stage; but promises to have a profound, if not extreme, influence on infrastructure development. This paper also describes our recent efforts towards digital archives of student portfolios, code-named Locutus.

Categories and Subject Descriptors

H.2.4 [Database Manager]: Management of heterogeneous formats – *multimedia streaming, document representation*. H.2.7 [Repository Manager]: Digital archive management – *archive content, archive document transport*. I.7.2 [XML]: Generic document markup – *multimedia, document representation*.

General Terms

Management, Documentation, Performance, Design, Reliability, Security, Human Factors, Standardization, Languages, Legal Aspects, Verification.

Keywords

Digital library, Digital archives, XML, Database management, Distributed file systems, Data compression, Multimedia, Multimedia streaming.

1. INTRODUCTION

The inherent need for documentation has existed from the beginning of civilization. In very early times, this task would have been performed in a verbal exchange between generations. This enabled future generations to not repeat the mistakes of its ancestors.

The physical representation of history largely determined accuracy and timeliness. Time also made archives obsolete. Ancient examples such as chiseled or painted stone using elaborate symbols, even if not a complete alphabet, portrayed a story or major event. Media has evolved from papyrus to bounded paper or paper-like volumes, to modern paper and digital media. Today we look to magnetic or optic media to store such history. Oddly enough, some recent digital formats antique more quickly in lieu of more “modern and sophisticated” formats. This process

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

43rd ACM Southeast Conference, March 18-20, 2005, Kennesaw, GA, USA. Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

functionally maroons old data to unreadable storage without the presence of a clear and deliberate migration plan.

Even today, in spite of current language and encoding techniques, we have allowed technological advancement to push valuable data into a virtual “black hole” through the antiquation of presentation and content marking techniques. This is sometimes done due to an encoding scheme’s inability to separate presentation from content. The other aspect to this type of content loss is data loss through lack of versioning controls. For example, imagine the editing techniques of writers. An ink pen and a few scribbles in the margins would not only make a correction to a document but also gave the reader a view into the thought process of the author – actually seeing the thought-refining process used to arrive at the final result. In spite of their incredible benefits, it can be argued that much data has been lost due to the editing and backspace function of word processors.

2. XML

Before we begin to look at the critical parts of a digital library (DL), we need to consider viable encoding and transport mechanisms for the Library of Digital Objects (LDOs) themselves. This is where markup schemes come to the rescue. We need a platform-neutral, vendor-neutral, language-neutral marking scheme capable of dealing with the problems described earlier when content and presentation cannot be separated. With content and presentation elements in an inseparable configuration, if either the content tagging or presentation tagging is antiquated, both elements are lost.

Consider the writing system employed by the ancient Egyptian civilization. It is one of the oldest writing systems in the world. Using hieroglyphic symbols, the Egyptians employed one of the very first markup schemes. Personal names were marked with an oval cartouche to distinguish them from ordinary words. They also used color to highlight important segments of a story. Consider such highlighting a type of pre-(word processing) boldface. The same symbols were used to instruct the reader to read either from left to right or right to left since early Egyptian text could be read from either direction [5].

Recent thoughts of textual markup came from the publishing industry through typesetters, printers, and editors. With the large amount of documentation being created as a result of office automation equipment and more specifically computers for technology-based industry, a need for standardized markup schemes emerged. Much of the early work in this area occurred at IBM. Ultimately the Generalized Markup Language (GML) was born. This would later be replaced by the Standard Generalized Markup Language (SGML) in 1980 [15]. XML is a subset of the SGML. The entire markup language is described and defined by SGML tags providing a verification and validation for the language. XML is actually two meta-languages in that it can be

used to describe other languages. The first is a set of rules that must be followed to produce well-formed XML documents. The second is used to develop a set of rules for producing a XML document type definition, or DTD, which allows the structure of the XML document to be validated. Further providing validation and definition for XML itself, the World Wide Web Consortium (W3C) has expressed the XML syntax in the Extended Backus-Naur Form (BNF) notation supplementing production rules with terse descriptions of the validity and well-formedness constraints [15].

```

<Employee>
  <FirstName>Eric</FirstName>
  <MiddleName>Lee</MiddleName>
  <LastName>Brown</LastName>
  <SSN>000-00-0000</SSN>
  <DateOfBirth>09/07/1960</DateOfBirth>
  <StartDate>09/01/1994</StartDate>
  <PayRate>20.00</PayRate>
  <HealthIns>true</HealthIns>
  <DisabilityIns>true</DisabilityIns>
  <RetirementPlan>>false</RetirementPlan>
  <Options>011101111100</Options>
</Employee>

```

Table 1. Sample XML Document for Employee

There is more to XML than just being content-centric. Through the use of Extensible Stylesheet Language (XSL) and Extensible Stylesheet Language Transformations (XSLT), one can mold our XML document in whatever format is necessary. The real power of XSLT is that different presentations of the same dataset can be developed with no alterations made to form or meaning of the original data [18]. An extension of this has immediately been applied to the Hyper Text Markup Language (HTML). A markup language defined by a XML DTD named XHTML provides all of the benefits for its HTML namesake with the validation and well-formedness of XML. XHTML, supported by most current web browsers, provides a web friendly interface to XML content.

XML allows its author to create a vocabulary to properly describe an entity. More importantly, the vocabulary can be specified through a DTD or schema definition (XSD) providing a way to assure the validation and well-formedness of a XML document based upon that definition. The vocabulary is human readable. Consider the example in Table 1. One can easily see that it is an employee record. It can be determined that this employee chose to participate in both insurance plans while opting out of the retirement plan. Even though one might not know what each Option element means exactly, one can guess that this is set of selected/not selected employment options. All of this can be done due to the human-readable nature of XML [18]. XML's strongest feature is its approach to data representation via a value-oriented approach. The core focus of this value-oriented approach, or programming if you will, is that data in and of itself is immutable. The manipulation of a certain value is not performed by the destruction of the previous value as done in many programming languages but by the creation of a new value. This method provides the best of both worlds by providing the new value necessary for processing while maintaining an audit trail of previous versions. Issues of

data manipulation take on an even greater significance when used in a distributed environment. Consider the overhead of a destructive value update where caching and data replication techniques are used to improve access times. This forces the system to employ as set of consistency protocols and semaphores to assure that the integrity of the manipulation is maintained for all users of the shared value. Such consistency tools are not required due to the atomic nature of the updates [17]

3. STRUCTURE

3.1 A Simple Example

First let's consider a simple model that most computer users would be comfortable with. Considering the Internet, in its simplest form, a user makes a request from a web browser to a server somewhere in the Internet. After some intermediate processing, host name resolution, and possibly a search, a server containing the desired content receives the request. The request is validated in some way. If all authentication constraints are satisfied, the content is packaged via the encoding scheme of a protocol and ultimately delivered to the requesting agent with some visualization modifications if needed. This example represents a successful transaction; of course, not all transactions are completed successful due to anomalies of some sort. While this example seems trivial, one can quickly recognize how complicated the transaction can become. The same can be said for a DL. Many "behind the scenes" actors in a document's transaction must be addressed.

3.2 Digital Library Process Examples

Figure 1 summarizes the major pieces of a DL as seen by the Corporation for National Research Initiatives' Digital Object Architecture Project. In an effort to further understand the pieces of the system and how they interoperate, let us analyze a typical transaction in light of the previous web browser to web server example.

For this example, let us assume that the user interface as shown in Figure 1 is a typical web browser on a computer with an Internet connection. We will also assume that there exists a set of global handle servers available to perform authentication and create handle generators as required. Our repository will be housed in a traditional relational database with the registry contained as a group of tables within the database for authentication and security role assignment. The search engine will be some type of indexing service connected with the web server using metadata provided by a backend connector to the database server [12].

First, we will look at a dissemination example. An originator makes a search request from a web portal providing a set of keywords for the search. The keywords are then packaged as a RAP request to the repository. Once received, the RAP request is disassembled and processed by the local search engine. The search engine pulls all digital objects with matching or pseudo-matching metadata. The list is compiled into a complete message, which is then returned to the originators web browser via a

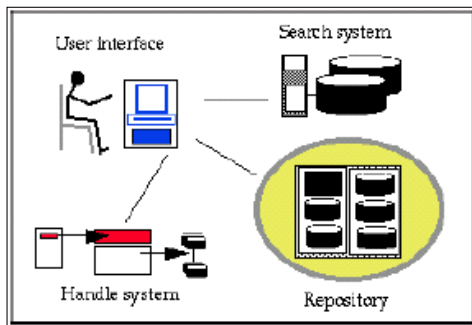


Figure 1 - Conceptual view of a digital library. (<http://www.cnri.reston.va.us/doa.html>)

RAP response over HTTP [13].

The originator may now select the digital object he/she would like to view much as if a traditional search engine request had been made. A summary of the information in each digital object listing is displayed. These summaries could be simple text, thumbnail images of a photographic collection, a short clip of a video stream, or a brief introduction of an audio file. Some of the information may have copyright information or other usage guidelines that may iconized in the summary.

Once an object is selected from the list, a retrieval request is made via a RAP request with the specific handle assigned to the registered digital object at the time of its registration. A lookup is performed using a handle server to determine which copy, if multiple copies exist, is logically closest to the originator. Once a repository has been identified by the handle server, retrieval is requested. The local repository via its registry must evaluate several criteria before releasing the digital object including but not limited to: (i) Is the originator an authenticated user of the repository? (ii) What is the security role or authorization level of the originator? (browser, contributor, editor, curator). (iii) What are the legal constraints binding access and usage of this digital object? Assuming that all constraints have been met, the digital object is returned to the originator's web browser in a complete or abbreviated form as requested. From this point, it is the responsibility of the local web browser's applications helper configuration to present the digital object in a usable format. At this point, dissemination has taken place.

While this may sound simple, consider the situation where the image is extremely large or so large that the local system does not have enough memory to process it. What if the data must be streamed at a rate faster than the supporting network can support in order to be usable? What if the underlying infrastructure cannot support the level of encryption required by the digital object's constraints? Again, as with the web browsing example given at the beginning of this section, we see that a simple transaction can easily become less than simple.

Now let us consider a deposit to the repository. With all of the assumptions as in the previous example standing, the originator prepares some digital material for submission. To be a true digital object, data or digital material must be combined with key metadata and possibly other metadata. As part of the process, the originator adds access information such as copyright notices or legal restriction upon the use of the proposed digital object. After the legal material, keywords are added to provide a descriptive profile for indexing by the repository. The originator would then markup the submission indicating if a handle should be requested or if the digital object should simply be stored without registration. Finally, the originator has the option of adding a signature to the proposed digital object.

The digital material and all of the profile information is then sent to the repository. Just as in the access example, authentication and

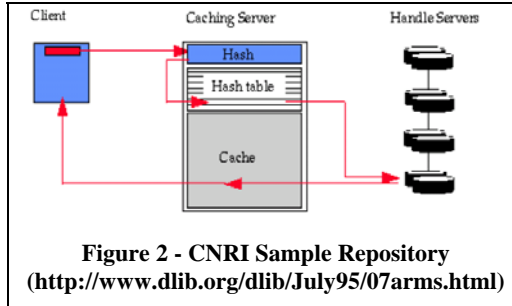


Figure 2 - CNRI Sample Repository
(<http://www.dlib.org/dlib/July95/07arms.html>)

server will complete the registration process and make the digital object available for future "global" searches.

3.3 Repository

While some might make the assumption that the repository is a simple relational database, that is not necessarily the case. The Kahn-Wilensky framework does not establish an implementation as to how the repository should be developed. The framework defines simply what the repository should do – that is to accept a LDO and maintain its state for future retrieval via a RAP request. The underlying structure could be any data storage and retrieval system – relational, object-oriented, or even XML/XML-hybrid. The actual implementation is hidden from the users by a repository access protocol or RAP which handles all aspects of authentication and security privilege audits in addition to providing a reliable "transport" of the digital object to the requesting application. The repository is responsible for the long-term storage of digital objects assigned to it. It may also provide services such as: (i) authentication and security role assignment via the registry subsystem, (ii) interpretation and view management in addition to

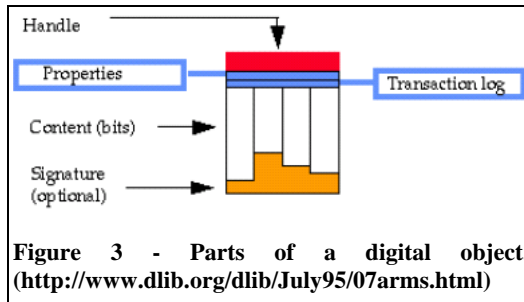


Figure 3 - Parts of a digital object.
(<http://www.dlib.org/dlib/July95/07arms.html>)

whatever viewing technologies are available via the client's interface. (iii) inter-repository replication management of specific digital objects as part of redundancy and scalability oversight, (iv) implementing digital objects that function as cataloging containers for the repository much the way an index services the contents of a printed book.

Authentication includes much more than a simple "Who are you?" question. In addition to knowing the individual making the request, considerations must be made for the level of trust the user has within the repository. These security levels would range from guest (limited view only), user (full submission and retrieval participation), to curator (complete administrative control). The logical source of the request must also be a calculated factor in access granting. The oversight role played by the curator of the collection also helps to ensure the content integrity of the collection by eliminating potential liabilities created by problem originators. These evaluations are performed by the registry subsection of the repository.

The repository might also deal with transformation management at a much lower level via XML. Consider an XML data store [17] employing a hierarchical structure that separates document in a large, tree structure providing built-in versioning control as well as retrieval benefits by loading only the portion of the document

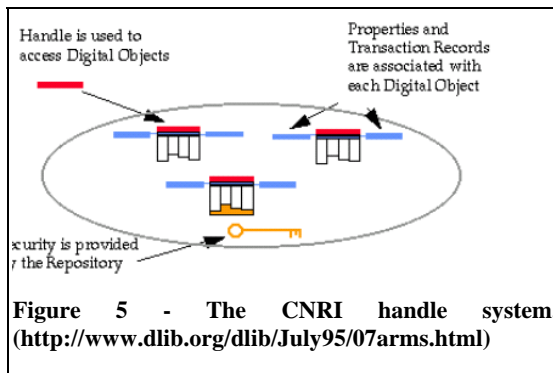
necessary for the transaction request. This format would allow the document to be traversed on disk avoiding a costly load of the entire document into memory. Improvements in performance and management of document replication could be realized from such a format. The XML data store system proposed by Thorn, Baumann, and Fennestad [17] provided such a unique perspective on an immutable value store providing several desirable elements for such a repository using the MIT Chord protocol such as: no single point of failure, scale gracefully, high performance / load balancing, fault tolerant and data availability assurance / self-organization.

3.4 Handle Services

The handle sub-system is critical to the lifetime of the document. Even if we properly establish a way to digitally store the material in a searchable form, it is the persistence of the naming mechanism that will determine the “shelf life” of the digital material submitted to these repositories. The handle system is responsible for providing a two-part, unique identification string that does not lose relevance as a result of storage location or the passage of time. The id consists of a local handle generator identifier concatenated with a locally unique string. Again, alluding to our web page request earlier in the paper, notice the similarity between the two parts of the handle and the network/hosts separation of the IP address. The two part scheme is a tremendous asset to the goal of global uniqueness. This namespace allows the digital object to be portable yet traceable. Even if the digital object is replicated to other repositories in the collection’s network, the handle and metadata concerning the ROR will provide a way to determine the origin of the object. The handle can be issued by a handle server for the collection domain. However, it is most likely the case that a handle server has spawned/authorized one or more handle generators to provide such services. The framework offers some guidance concerning naming schemes but little else in the way of implementation of this system.

3.5 Search Engine

The search engine simply indexes the metadata provided by the repository for each of the contained digital objects. Search engines must pay special attention to digital meta-objects that serve the role of a catalog of a specialize subset of the collection. These digital objects are simply registered digital objects that contain handle information of related digital objects represented by the metadata in the header of the digital metaobject [16]. With proper manipulation of the metadata provided by the



repository, existing search engines could assist in the role of handle servers for public collections on the Internet. Proper configuration of existing DNS servers might allow them to play the role of global handle servers.

3.6 User Interface

The user interface has no definition in the Khan-Wilensky model other than to

say that one will be provided and that the data contained within the digital object should be altered to assure that some acceptable level of presentation can be provided in the event that the raw data cannot be processed. All of the examples in this paper have assumed that the user interface would be a typical web browser with content transformed to be presented within the XML/XHTML markup language with the appropriate helper functions.

4. FUTURE NEEDS

4.1 Legal Issues

Many of the factors that will shape DLs and related archival technologies lay in the hands of the lawyers and the courts more so than in the hands of the computer science community. While laws concerning the rights and proper use of tangible materials such as books and machines have been on the books for over 100 years, these “real world” applications have not found their way to digital materials [6]. In fact, some today believe that the policy and procedures for the content accountability, distribution, and licensing of digital materials exist in an amoebic state which is determined by the level of activism of a particular court of jurisdiction.

Another major issue is content accountability. Who is to be held responsible for the content of a body of work, the real author of the work or the content provider, maybe called a transport agent? Should the messenger be shot for the content of a message produced by someone else? These are questions that have been brought to light with cases such as Napster and the RIAA. More

importantly, should content hosts be granted the same rights and protections provided newspapers and other traditional content outlets? Unfortunately, outside of organizations that can acquire solid distribution agreements from specialized groups, these questions will likely remain. Even for those that provide solid agreements, we must consider how long those agreements are binding and if there exists some extra-agreement force that could render the agreement invalid or outdated. Copyright laws surrounding digital properties continue to be ambiguous [6].

4.2 Integration Issues

A lack of integration of digital sources,

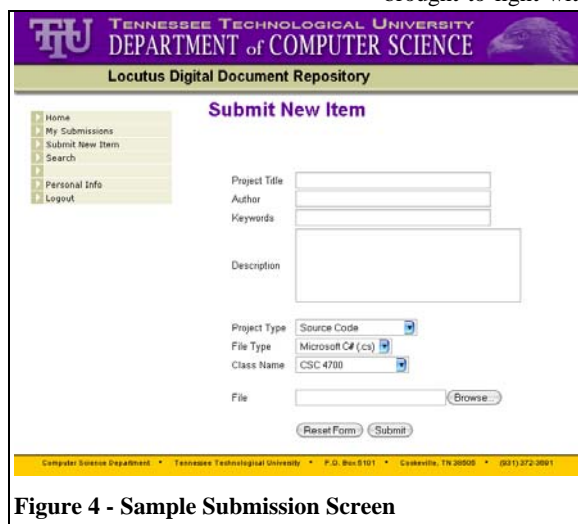


Figure 4 - Sample Submission Screen

especially within the academic community is the greatest barrier to growth of digital archives. [8] While some committee work and framework layout has taken place in recent years, a large body of work remains unaddressed. It is critical to harness the material provided by traditional and non-traditional sources in a consistent and methodical ways that honor the integrity of the content's source, intent, and dissemination guidelines.

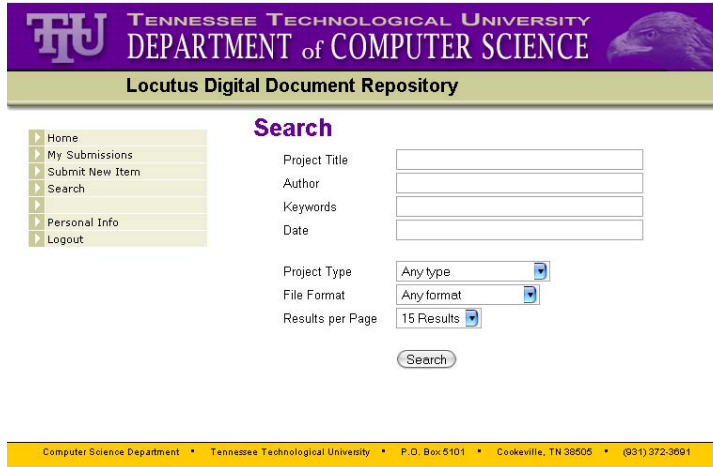


Figure 6 - Sample Search Screen

In our department, academic content comes from a variety of sources including WebCT courses, online content from publishers, open source and commercial web sites, and from personal web sites of faculty and staff members. Secure and legal acquisition protocols must be established to incorporate this information into a central archive. Another source to consider is video material from distance learning courses in-house and from external university sources. This leads back to legal issues of ownership and material dissemination guidelines.

5. LOCAL EFFORTS

We are in the early stages of developing a digital library. For many reasons, a decision has been made to create a in-house library archive. These reasons included (i) recovery of what would otherwise be lost works, (ii) better service to the research component of our undergraduate and graduate programs, (iii) better service to the reviewers and evaluators. The ultimate goal of this project is a fully-functional digital archive to address the objectives outlined above. The project is phased into four distinct phases.

Phase 1 will concentrate on collecting data and source code from existing efforts made as part of course projects. While each of these projects demonstrated some merits, all of them fell short in terms of one or more critical areas: deployment, maintainability, scalability, and usability. Based upon the analysis of these projects and current technologies, we will build a basic framework that will allow for simple submissions with limited reporting. For reasons of simplicity and uniformity, the decision was made to use a web browser for the client interface. Our goal is to develop a prototype that has its

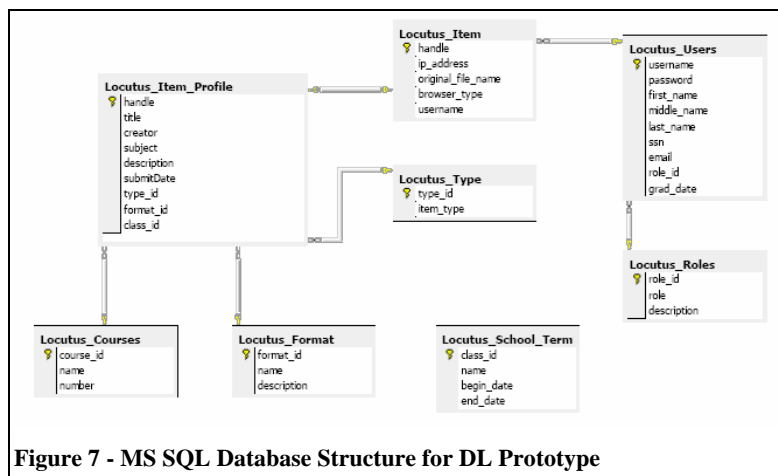


Figure 7 - MS SQL Database Structure for DL Prototype

basis in the Kahn-Wilensky model with guidance from the Dublin Core Metadata Initiative.

Phase 2 will concentrate on the establishment of a set of XML templates that will service as the main transport mechanism for the archived documents and the reports based upon them. As referenced earlier in the XML section, there exists an active research line in the area of distributed value-oriented XML stores. We see this area of research as an intriguing proposal that would allow for increase flexibility and reliability in document storage.

XML will not only take its obvious roll as the catalysis for inter-archive document transfer, it will also play a significant role in (i) database management and transactional processing, (ii) document grooming and presentation, and (iii) serve as a foundation for rules-based document routing and processing.

Phase 3 will concentrate on improvements and extensions of the reporting sub-system leveraging techniques available via the XML framework established in Phase 2. Development work will be in areas of current XML parsing and searching algorithms. Proposed XML extensions to the current SQL database servers will make the reporting sub-system much more flexible and efficient.

Phase 4 will concentrate on the integration and assimilation of digital resources across the campus. For total benefit to be achieved from the local DL project, logical protocols and administrative procedures must be put in place to assure that this information makes it way ultimately to a cataloged DL rather than a lost zip file. To address these concerns, it should be possible to develop an inter-repository repository access protocol (IRAP) and an extra-repository repository access protocol (ERAP). Such protocols would provide safe and secure mechanisms to not only distribute registered or stored digital objects across the network infrastructure, ERAP would provide a standard protocol to datamine resources external to the collection. While these protocols

are not outlined in the 1995 framework, they would seem to be a logical extension of the framework to allow for system and collection interoperability.

While some solutions are available on the web using open source technologies, most notably the Fedora project, the decision was made to begin development of a new system customized to our needs. The initial focus was on open source initiatives, with assumption that some combination of

Linux, Apache, MySQL, PHP, and Python would work. However, due to the increasing availability of .NET platform experience and access through the MSDN Academic Alliance, the decision was made to move forward with a .NET implementation. Our model will use IE for a client interface. The search engine will make use of indexing services provided by IIS and metadata cataloging features within SQL Server 2000. The database will essentially store user profile and document profile information. The database will also provide lookup verification for submission and search screens presented via the web. The actual document data will be stored in an IIS-accessible folder using zip compression. The document data will use the assigned handle as a filename. The original file name is maintained in the profile information.

The Locutus_Users and Locutus_Roles tables provide the registry function described in the reference model. For our prototype, we have decided to provide 4 authorization levels. The view-only role will be referred to as a guest. This role will be used by reviewers for evaluation. The student role will allow the originator to make deposits into a specific virtual container. This role will be allowed to make withdrawals of objects approved to public release. The staff role will have the same rights as the student role with limited administration rights allowing tasks such as class container maintenance as well as creating local specialized collections. Finally, the curator role will have complete administrative rights to all parts of the system including starting and stopping sub-systems. The handle system will be an independent sub-system that can be started and stopped via the master web service. This same web service will serve the role of a meta-RAP managing all calls made the repository for deposits, disseminations, and system maintenance requests. The web service will be based upon C# code using ASP.NET 1.1. This project also coincides with a major upgrade to our high performance computing infrastructure which will be based upon the Apple G5 platform. This infrastructure will become a valuable part of the several proofing exercises including: (i) handle server distribution and management, (ii) distributed file system protocol and implementations, (iii) use of traditional relational database systems (iv) use of new hierarchical database system and XML value stores, (v) scalability and redundancy tests

6. CONCLUSIONS

In this paper, we looked at DLs/archives from various perspectives including (i) the background and basic need for such entities; (ii) the structure of such entities including data storage, data representation, data migration, and security; and (iii) future issues that must be addressed by academia, industry, political and legal processes for DLs to become widely accepted and useful. Lawyers, their backing institutions, and congress will profoundly hinder several outstanding challenges in developing an accountable, maintainable, flexible legal framework that translates to current and future technology standards. Our efforts in their DL/archive implementation are in its infancy. Issues that need to be addressed include: testing the framework's durability, flexibility, scalability, GUI validation and the total end user experience evaluation.

7. REFERENCES

[1] Arms, W. Implementing Policies for Access Management. D-Lib Magazine (Feb. 1998).
<http://www.dlib.org/dlib/february98/arms/02arms.html>

- (current as of November 2, 2004).
- [2] Arms, W. Key Concepts in the Architecture of the Digital Library. D-Lib Magazine (Nov. 1995)
<http://www.dlib.org/dlib/July95/07arms.html> (current as of November 2, 2004).
- [3] Bishop, A.P., Van House, N.A., Battenfield, B.P. Digital Library Use: Social Practice in Design and Evaluation, MIT Press, 2003.
- [4] Corporation for National Research Initiatives. Digital Object Architecture Project. <http://www.cnri.reston.va.us/doa.html> (current as of November 2, 2004).
- [5] Cover Pages. Encoding and Markup for Texts of the Ancient Near East. (Feb 12, 2002).
<http://xml.coverpages.org/xmlMarkupANE.html> (current as of November 2, 2004).
- [6] Coyle, K. The "Rights" in Digital Rights Management. D-Lib Magazine Vol. 10. #9. (Sept. 2004).
<http://www.dlib.org/dlib/september04/coyle/09coyle.html> (current as of November 2, 2004).
- [7] Digital Libraries Initiative Phase 2. <http://www.dli2.nsf.gov> (current as of November 2, 2004).
- [8] Flecker, D., McLean, N. Digital Library Content and Course Management Systems: Issues of Interoperation. Digital Library Foundation. July 2004 Report.
- [9] George, B., Haritsas, J. R. Secure buffering in firm real-time database systems?, The VLDB Journal (Nov. 2000). 178-198.
- [10] Griffin, S. 4th DELOS Workshop. Eval. of Digital Libraries: Testbeds, Measurements, and Metrics. (6-7 June 2002).
http://www.dli2.nsf.gov/internationalprojects/working_group_reports/evaluation.html. (current as of November 2, 2004)
- [11] <http://www.cs.cornell.edu/payette/prism/security/policy.htm> (current as of November 2, 2004)
- [12] Johnston, W., Guojun, J., Larsen, C., Lee, J., Hoo, G., Thompson, M., Tierney, B. Real-Time Digital Libraries based on Widely Distributed, High Performance Management of Large-Data-Objects.
<http://dsd.lbl.gov/WALDO/DigLib/LargeDataObj-Arch.fm.html> (current as of November 2, 2004).
- [13] Kahn, R., Wilensky, R. A Framework for Distributed Digital Object Services. (May 13, 1995)
<http://www.cnri.reston.va.us/cstr/arch/k-w.html> (current as of November 2, 2004).
- [14] Lagoze, C. The Cornell Digital Library Research Group: Architectures and Policies for Distributed Digital Libraries.
<http://www.cs.cornell.edu/payette/Prism/Security/Architectures and Policies for Distributed Digital Libraries.htm> (current as of November 2, 2004).
- [15] Phillips, L. A. Special Edition Using XML. QUE. ISBN: 0-7897-1996-7. (Copyright @ Aug. 2000).
- [16] Summann, F., Lossau, N., Search Engine Technology and Digital Libraries - Moving from Theory to Practice. D-Lib Magazine Vol. 10. #9. (Sept. 2004).
<http://www.dlib.org/dlib/september04/lossau/09lossau.html> (current as of November 2, 2004).
- [17] Thorn, T., Baumann, A., Fennestad, M., A distributed, value-oriented XML Store. MS Thesis - IT Univ. of Copenhagen. (Aug. 2002).
- [18] Wyke, R.A., Rehman S., Leupen B. XML Programming. Microsoft Press. ISBN: 0-7356-1185-8. (Copyright @ 2002)