Proc. Int. Conf. on Dublin Core and Metadata for e-Communities 2002: 27-34

Archon - A Digital Library that Federates Physics Collections

K. Maly, M. Zubair, M. Nelson, X. Liu, H. Anan, J. Gao, J. Tang, Y. Zhao
Computer Science Department
Old Dominion University
Norfolk, Virginia, USA
{maly,zubair,mln,liu_x,anan,gao_j,tang_j,yzhao}@cs.odu.edu

Abstract

Archon is a federation of physics collections with varying degrees of metadata richness. Archon uses the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) to harvest metadata from distributed archives. The architecture of Archon is largely based on another OAI-PMH digital library: Arc, a cross archive search service. However, Archon provides some new services that are specifically tailored for the physics community. Of these services we will discuss approaches we used to search and browse equations and formulae and a citation linking service for arXiv and American Physical Society (APS) archives.

1. Introduction

Archon is a federation of physics digital libraries. Archon is a direct extension of the Arc digital library [13]. Its architecture provides the following basic services: a storage service for the metadata of collected archives; a harvester service to collect data from other digital libraries using the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [10]; a search and discovery service; and a data provider service to expose the collected metadata to other OAI harvesters. However, for Archon we have developed services especially for physics collections based on metadata available from the participating archives that go beyond the required (by the OAI-PMH) unqualified Dublin Core (DC) [22]. For example, we provide a service to allow searching on equations embedded in the metadata. Currently this service is based on LaTeX [11] representation of the equations (due to the nature of archives used), but we plan to include MathML [8] representations in the near future. We also use context-based data to search for equations related to specific keywords or subjects. By intelligent template matching, a cross-archive citation service has been developed to integrate heterogeneous collections into one unified linking environment.

2. Overview of Archon Services

The Archon architecture is based on the Java Servlets-based search service that was developed for Arc and earlier for the Joint Training, Analysis and Simulation Center (JTASC) [16]. This architecture is platform independent and can work with any web server (Figure 1). Moreover, the changes required to work with different databases are minimal.

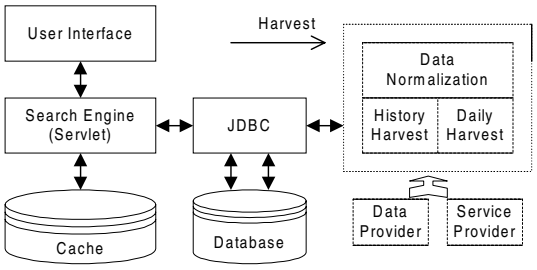

Figure 1. Overall Architecture

2.1 Search Service

The search server is implemented using Java Servlets (Figure 2). The session manager maintains one session per user per query. It is responsible for creating new sessions for new queries (or for queries for which a session has expired). Sessions are used because queries can return more results (hits) than can be displayed on one page, Caching results makes browsing through the hits faster. The session manager receives two types of requests from the client: either a request to process a new query (search); or a request to retrieve another page of results for a previously submitted query (browsing). For a search request, the session manager

calls the index searcher that formulates a query (based on the search parameters) and submits it to the database server (using JDBC) then retrieves the search results. The session manager then calls the result displayer to display the first page. For a browsing request, the session manager checks the existence of a previous session (sessions expire after a specific time of inactivity). If an expired session is referenced, a new session is created, the search re-executed, and the required page displayed. In the case where the previous session still exists, the required page is displayed based on the cached data (which may require additional access to the database).

2.2 Storage Service

The OAI-PMH uses unqualified Dublin Core as the default metadata set. Currently, Archon services are implemented based on the data provided in the DC fields, but in the prototype implementation we are already using richer metadata sets. All DC attributes are saved in the database as separate fields. The archive name and set information are also treated as separate fields in the database for supporting search and browse functionality. In order to improve system efficiency, most fields are indexed using full-text properties of the database, such as the Oracle InterMedia Server [18] and MySQL full-text search [9]. The search engine communicates with the database using JDBC [20] and Connection Pool [17].

2.3 Harvester

Similar to a web crawler, the Archon harvester (same as the Arc harvester) traverses the list of data providers and harvests metadata from them. Unlike a web crawler, the Archon harvester performs metadata normalization, and exploits the incremental, selective harvesting defined by the OAI-PMH. Data providers are different in data volume, partition definition, service implementation quality, and network connection quality: all these factors influence the harvesting procedure. Historical and newly published data

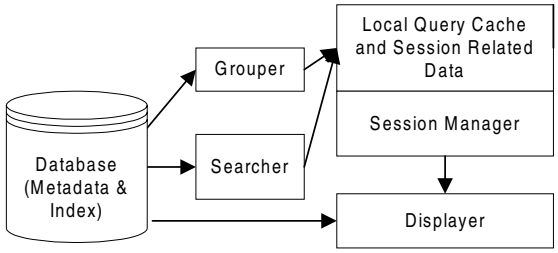

Figure 2. Search Server Implementation

harvesting have different requirements. When a service provider harvests a data provider for the first time, all past data (historical data) needs to be harvested, followed by periodic harvesting to keep the data current. Historical data harvests are high-volume and more stable. The harvesting process can run once, or, as is usually preferred by large archives, as a sequence of chunk-based harvests to reduce data provider overhead. To harvest newly published data, data size is not the major problem but the scheduler must be able to harvest new data as soon as possible and guarantee completeness – even if data providers provide incomplete data for the current date. The OAI-PMH provides flexibility in choosing the harvesting strategy; theoretically, one data provider can be harvested in one simple transaction, or one is harvested as many times as the number of records in its collection. But in reality only a subset of this range is possible; choosing an appropriate harvesting method has not yet been made into a formal process. We define four harvesting types:

1. bulk-harvest of historical data
2. bulk-harvest of new data
3. one-by-one-harvest of historical data
4. one-by-one-harvest of new data

Bulk harvesting is ideal because of its simplicity for both the service provider and data provider. It collects the entire data set through a single http connection, thus avoiding the overhead of multiple network connections. However, bulk harvesting has two problems. First, the data provider may not implement the optional resumptionToken flow control mechanism of the OAI-PMH, and thus may not be able to correctly process large (but partial) data requests. Secondly, XML syntax errors and character-encoding problems are surprisingly common and can invalidate entire large data sets. A discussion of general issues regarding metadata variability in OAI-PMH harvesting can be found in Liu, et al. [14].

One-by-one harvesting is used when bulk harvesting is infeasible. However, this approach imposes significant network traffic overhead for both service and data providers since every document requires a separate http connection. The default harvesting method for every data provider begins as bulk harvest. We keep track of all harvesting transactions and if errors are reported, we determine the cause and manually tune the best harvesting approach for that data provider.

The Arc harvester is implemented as a Java application. At the initialization stage, it reads the system configuration file, which includes properties such as user-agent name, interval between harvests, data provider URL, and harvesting method. The harvester then starts a scheduler, which

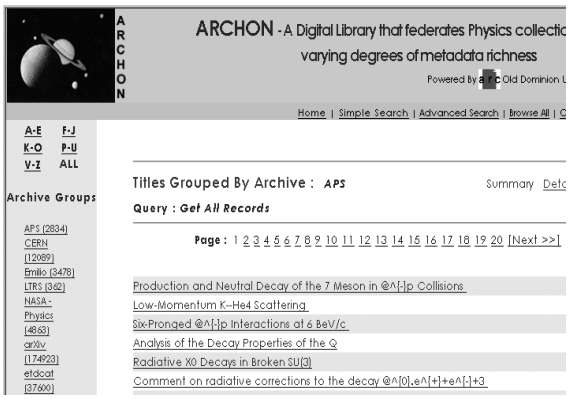

Figure 3. Archon Interface for Searching

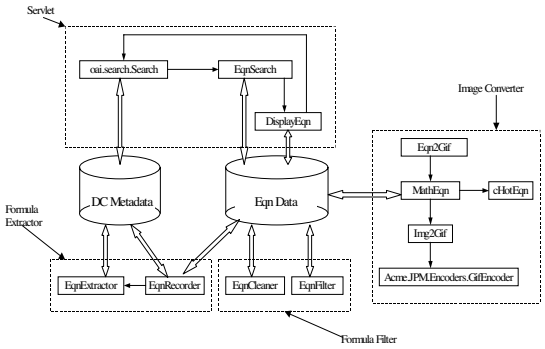

Figure 4. Equation Search and Display Service Architecture

periodically checks and starts the appropriate task. Some archives such as Emilio [5] were not OAI-PMH compliant. To overcome this problem, we created a gateway that crawls the Emilio web site and acts as a data provider to provide metadata that is harvested into Archon (Figure 3).

2.4 Data Provider Service

The data provider service manages OAI-PMH requests to Archon and allows Archon to act as an aggregator for the metadata contents it harvested from other digital libraries.

3. Equations-Based Search

In Archon, many metadata records contain equations in LaTeX and other formats. These equations are harvested as text format and not easy for users to browse and view. It is a value-added service to search equations by traditional text query but present it in a user-friendly way (e.g GIF file). By this method we build virtual metadata (images) over the original flat text metadata. Issues that where addressed to enable Archon to search and browse equations include:

1. Rendering of equations and embedding them into the HTML display.
2. Identifying equations inside the metadata.
3. Filtering common meaningless equations (such as a single n) and incomplete equations.
4. Equation storage.

3.1 Rendering of Equations

Most of the equations available on Archon are written in LaTeX. However, viewing enocoded LaTeX equation is not as intuitive as viewing the equations themselves, so it is useful to provide a visual tool to view the equations. There are several alternatives to display equations in a HTML page. One alternative is to represent equations using HTML tags. This is an appropriate choice only for simple expressions; using this method severely limits what can be displayed with the usual notation. A browser may not be able to properly display some special symbols, such as integral or summation symbols or Greek characters. The alternative we chose is to write a program to convert the LaTeX equations into an image and embed it inside the HTML page. We implemented this tool as a Java applet.

3.2 Identifying Equations

LaTeX equations have special characters (such as $) that mark the start and end of LaTeX strings. However, the presence of these symbols does not automatically indicate the presence of equation. Moreover, an equation can be written as a sequence of LaTeX strings instead of as a whole LaTeX string. This is why we implemented a simple state machine based program to identify equations. Some of the rules used in this state machine are:

1. Isolate the unpaired '$' symbol;
2. Glue the small pieces together into the whole formula;
3. Check the close neighbors (both ends) of a LaTeX string to obtain a complete equation.

3.3 Filtering Equations

Despite our progress to date, there are many situations which cannot be solved by the methods described above,

because it is impossible to distinguish if a string is a part of formula when it is not quoted with '$' symbols. We have some "broken" formulas due to this reason. We worked around these limitations by filtering those formulae out. We established a "rule book" where every rule is a pattern of the regular expression which describes what kind of LaTeX string is going to be dropped. Every collected LaTeX string is checked against the rules and ay matching LaTeX strings are removed.

Furthermore, there are also some formulae with 'illegal' LaTeX symbols. Some of these 'illegal' symbols are misspellings, such as a missing space or mistaken use of the backslash symbol. Some of these symbols are user defined. A general-purpose LaTeX string parser cannot properly handle them. All of these will cause a blank image or a formula with missing parts, because the image converter cannot pick up the corresponding display element for it. To solve this problem, each extracted LaTeX string is screened and strings having 'illegal' symbols are dropped.

3.4 Equation Storage

For fast browsing, we stored the extracted equation in a relational database. Figure 4 shows the schematic class diagram that shows the relationships between the classes and the relationships between the classes and the database.

Overall, we provide a novel search function, search with equation, to our digital library. To realize this function, LaTeX strings that are used to express equations are extracted from the metadata records. The extracted LaTeX strings are filtered and cleaned to eliminate errors and illegal symbols. Then the clean LaTeX strings are converted into GIF images. We have provided three search alternatives for the user in the search interface Figure 5.

1. Search for the LaTeX string directly.
2. Display a list of all equations and the user can select an equation visually.
3. Search for equations by subject or abstract keywords.

For example, when a user types in a word such as 'Newton' into the 'abstract 'field in Figure 5, we will present to the user all images of formulae that occur in the abstract of papers that contain the keyword 'Newton'. Once a user has selected a subject entry in the box shown in Figure 5, we again display all formulae that occur in papers categorized as having that subject. Finally, by clicking on the formula such as shown in Figure 6, users will receive all the records related to this formula.

At this point we have completed this service for arXiv and are in the process to include the other archives shown in Figure 3. Our approach is to convert all local representation to LaTeX and then use the currently implemented scheme.

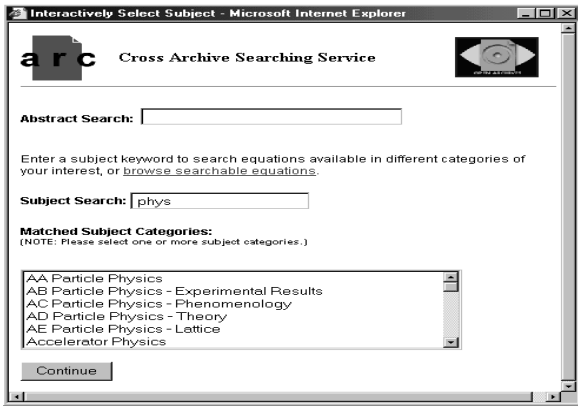

Figure 5. Formula Search Interface

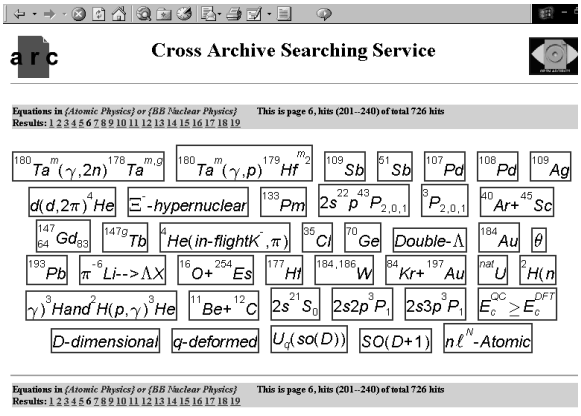

Figure 6. Formula Search Result Page

For instance, APS will export its equation in the metadata records in MathML through OAI-PMH's parallel metadata harvesting scheme and we will translate them to LaTeX and store them in our database. CERN already uses LaTeX so it is only a matter of time to use their metadata records.

4. Reference Linking Service

The reference linking service provides a convenient method to access references in a document. It provides the references information for a document as well as links from the references to their corresponding documents. There are several kinds of reference linking services. One method is to provide reference links within a controlled collection, such as the Open Journal Project [7]. The feature of our reference linking service is to provide reference-linking service among several collections, the membership of which is subject to change.. The service architecture is shown in Figure 7. In addition to providing reference service for Archon users, we will consider extending our approach for:

1. OAI Citation Provider: Implementing an OAI layer to let other service providers to harvest the citation information from our collections.
2. Public Cross-linking Service: Users can get the reference information by issuing an OpenURL request [21].

The following sub-sections describe our approach in implementing reference linking along with a number of issues that we addressed.

4.1 Obtaining Reference information

In order to acquire reference information, we divided the sources into three categories:

OAI-Compliant Data Provider Some data providers, such as APS and CiteBase (http://citebase.eprints.org), provide reference information in their specific metadata formats. CiteBase extracts citation information from LaTeX source files in arXiv. In this case, we harvest reference information directly.

Online Citation Service Some data providers, such as arXiv, provide online Citation Service. When a user searches an article in arXiv, he/she can press "reference" link to get the reference information about this article. In this case, we have two choices: use a gateway to make it OAI-compliant, or issue HTTP requests to get HTML files and process HTML files directly.

Articles It is possible that there is neither harvestable reference information nor an online citation service available. In this case, one may directly extract the reference information from the article's text. CiteSeer [12] and CiteBase have used this approach.

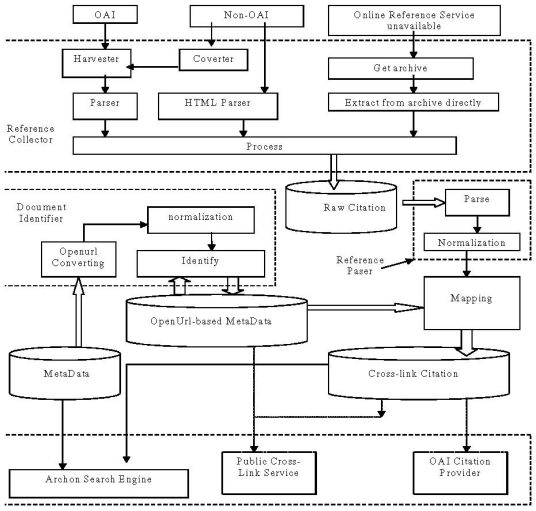

Figure 7. Service Architecture for Reference Linking in Archon

4.2 Internal Citation Format and Citation Parsing

Since documents and citations are collected from heterogeneous sources and formats, they must be integrated into one unified format for processing. We also consider re-exposing citation information and supporting third-party linking in a standard way during design. Several formats have been studied such as Academic Metadata Format [3] and DC-Citation standard [1]. However, The DC-Citation standard is still under development, and it only addresses journal citation information so far.. We chose the OpenURL [21] metadata sets for four reasons:

1. OpenURL is expressive and accurate in identifying a document.
2. OpenURL has been widely supported and evolves in the context of its NISO standardization.
3. We plan to extend our service to support OpenURL in the future.
4. The possible converge of DC and OpenURL as discussed in [19].

The heterogeneous citation data are converted to the OpenURL metadata set. For example, APS citations are expressed in a XML format that makes it easy for parsing. An example of an APS citation is:
$< citationcid = "C" >$

$< ref >$
$< inbook >$
$< refauth > Langley < /refauth >$
$< booktitle > ReportonMt.WhitneyExpedition$
$< /booktitle >, Profess.Papers,$
$< publisher > U.S.SignalService < /publisher >$
$< volume > XV < /volume >$
$< /inbook >$
$< /ref >$
$< /citation >$

For CiteBase references, the citation information is stored in a semi-structured string that requires heuristics to parse. An example of a CiteBase citation is:
$< relation.References >$
$H.F.Fongetal.Phys.Rev.Lett., 75 : 316, 1995$
$< /relation.References >$

In this case, we implemented a simple state machine based program to parse the citation information. The state machine tries to match the citation against several pre-defined patterns. Some patterns are:

1. Creators, article title, journal short title, volume, issue, start page, year
2. Creators, article title, journal short title, volume, issue, start page-end page, year
3. Creators, journal short title, volume, issue (year), start page
4. Creators year journal short title, volume, issue, start page

Since no uniform format is defined, normalization and other heuristic processing are necessary. For example, our heuristic algorithm will identify "Phys. Lett.", "Phys. Lett. B 15", "Phys. Lett 15B", "Phys. Letter B 15" as the same journal.

4.3 Match between citations and documents

The OpenURL metadata sets almost cover every field that is necessary to identify a document. But document and citation only use a subset of these fields. It is possible that some documents use only the first author and article title while others use journal title, volume number and start page. In our approach, we use multiple rules to match citations and documents based on what kind of the information is present.

Despite our effort, there are many cases that a reference fails to match any document in our collections. There are two possibilities: the referred document does not exist in our collection or the referred document exists in our collection but the matching algorithm failed to find the document

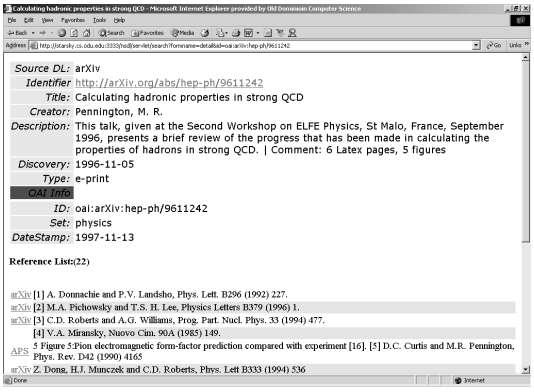

Figure 8. Reference Display in Archon

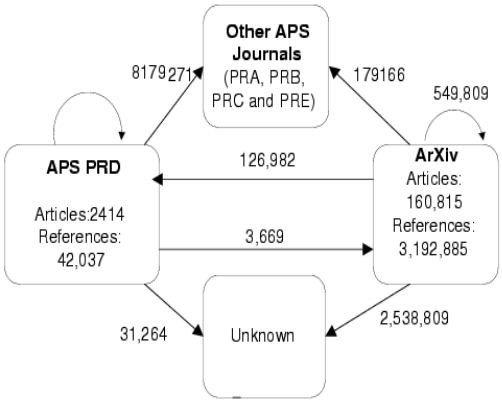

Figure 9. Initial Result of References Processing in Archon

Our approach is to compute the similarity between the citation and documents to find possible links. The possible links are presented to the user if the similarity is larger than a pre-defined threshold. The advantage of this approach is that it gives users some possible links in which users may be interested.

Since Archon harvests from various data sources, the same document may exist in more than one source. For example, there is significant overlap between arXiv and APS. To address this problem, we developed algorithms to detect duplicate documents. The duplication is presented the user and we leave the user to select the appropriate copy.

Figure 8 shows the user interface of reference information. Figure 9 shows number of references identified between APS (a subset of the journal Physical Review D) and arXiv. The number of unidentified or unknown references

is currently large, but we are addressing this number by applying more intelligent normalization techniques to identify references.

5. Discussion and Future Work

Future work will include updating the Archon harvester and data provider to be compliant with OAI-PMH 2.0, which was released in June 2002 and will run concurrently with OAI-PMH 1.1 until the end of 2002. At that time, OAI-PMH 2.0 will be the only version officially sanctioned by the OAI. Fortunately, OAI-PMH 2.0 represents an incremental evolution of version 1.1, so conversion will not be difficult. Usage of unqualified DC as a common metadata format in OAI-PMH proves to be very helpful for building a quick prototype. However, richer metadata formats are essential for building a richer service. All of the data providers harvested by Archon support metadata formats richer than unqualified DC. Specific parser and citation extraction algorithm have been developed for each of these metadata formats. We consider a standard and rich metadata format for scholarly communication is essential for building richer service over a large number of heterogeneous data providers.

We also plan to continue to refine the equation and citation services. For the equations, we plan to define categories of equations and allow "fielded" searching within those categories of equations. We believe this will increase the precision of equation-based searching. We also created some interfaces for equation search and we are planning to adapt these interfaces to be easier to use from the user point of view. For the citation linking service, we intend to increase the accuracy of our citation parsing and more fully support the OpenURL reference linking framework.

In summary, we created Archon, a digital library for physics. We added services for easier search and browsing of archives as well as their related documents. Our collection includes several OAI-PMH compliant repositories such as arXiv and non OAI-PMH compliant repositories such as Emilio. Other projects like CiteBase, Cyclades [4] and Torii [2] also provide value-added service for physical collections and we plan to compare these services and explore the possibility of cross service linking. At this point it is only our contention that adding equation based search and full cross-linking across all participating archives is a valuable service. In the months to come we will perform user testing to see if these service are welcomed by the physics community. Our prototype implementation has implemented standard ways to ingest metadata of different degree of sophistication and representation and make use of them in a meaningful way.

References

[1] Apps, A. (2002) A Journal Article Bibliographic Citation Dublin Core Structured Value. Available at http://epub.mimas.ac.uk/DC/citdcsv.html

[2] Bertocco, S. (2001). Torii, an Open Portal over Open Archives, HEP Libraries Webzine, 1(4). Available at http://library.cern.ch/HEPLW/4/papers/4/

[3] Tim D. Brody, Zhuoan Jiao, Thomas Krichel & Simeon M. Warner (2001) Syntax and Vocabulary of the Academic Metadata Format. Available at http://amf.openlib.org/doc/ebisu.html

[4] Cyclades project. http://www.ercim.org/cyclades/

[5] Emilio. AIP Emilio Segr Visual Archives, American Institute of Physics. Available at ¡http://www.aip.org/history/esva/use.htm¿

[6] Harnad, S. & Carr, L. (2000). Integrating, navigating and analyzing open eprint archives through open citation linking (the OpCit project). Current Science Online, 79(5). Available at http://www.ecs.soton.ac.uk/ harnad/Papers/Harnad/harnad00.citation.htm.

[7] Hitchcock, S., Carr, L., Hall, W,, Harris, S., Probets, S., Evans, D. & Brailsford, D. (1998). Linking electronic journals: Lessons from the Open Journal project. D-Lib Magazine, 4(12). Available at http://www.dlib.org/dlib/december98/12hitchcock.html.

[8] Ion, P. & Miner, R. (eds) (1999). Mathematical Markup Language (MathML) 1.01 Specification, W3C Recommendation. Available at http://www.w3.org/TR/REC-MathML/

[9] Kofler, M. (2001). MySQL. New York, NY: Springer.

[10] Lagoze, C. & Van de Sompel, H. (2001). The Open Archives Initiative: Building a low-barrier interoperability framework. Proceedings of the First ACM/IEEE Joint Conference on Digital Libraries, Roanoke, VA. Available at http://www.openarchives.org/documents/oai.pdf.

[11] Lamport, L. & Bibby, D. (1994). LaTeX: A Document Preparation System, 2nd edition. MA: Addison Wesley.

[12] Lawrence, S., Giles, C. L. & Bollacker, K. (1999). Digital Libraries and Autonomous Citation Indexing. IEEE Computer, 32(6), 67-71.

[13] Liu, X., Maly, K., Zubair, M. & Nelson, M. L. (2001). Arc - An OAI service provider for digital library federation. D-Lib Magazine, 7(4). Available at http://www.dlib.org/dlib/april01/liu/04liu.html.

[14] Liu, L., Maly, K., Zubair, M., Hong, Q., Nelson, M., Knudson, F. & Holtkamp, I. (2002). Federated Searching Interface Techniques for Heterogeneous OAI Repositories, Journal of Digital Information, 2(4). Available at http://jodi.ecs.soton.ac.uk/Articles/v02/i04/Liu/.

[15] Maly, K., Zubair, M. & Liu, X. (2001). Kepler - An OAI Data/Service Provider for the Individual. D-Lib Magazine, 7(4). Available at http://www.dlib.org/dlib/april01/maly/04maly.html.

[16] Maly K., Zubair M., Anan H., Tan D., & Zhang Y. (2000) "Scalable Digital Libraries based on NCSTRL/DIENST", Proceedings of ECDL 2000, Lisbon Portugal, pp. 168-180.

[17] Moss, K. (1999). Java Servlets (Second Edition). Boston, MA: McGraw-Hill Companies, Inc.

[18] Oracle. (2001). Oracle InterMedia Server.

[19] Powell, A. & Apps, A. (2001). Encoding OpenURLs in Dublin Core metadata, Ariadne Magazine, Issue 27, Available at http://www.ariadne.ac.uk/issue27/metadata/.

[20] Reese, G. (2000). Database programming with JDBC and Java. Sebastopol, CA: O'Reilly & Associates.

[21] Van de Sompel, H. & Beit-Arie, O. (2001). Open Linking in the Scholarly Information Environment Using the OpenURL Framework. D-Lib Magazine, 7(3). Available at http://www.dlib.org/dlib/march01/vandesompel/03vandesompel.html.

[22] Weibel, S., Kunze, J., Lagoze, C. & Wolfe, M. (1998). Dublin Core metadata for resource discovery. Internet RFC-2413. Available at ftp://ftp.isi.edu/in-notes/rfc2413.txt.