

2001

Arc - An OAI Service Provider for Digital Library Federation

Xiaoming Liu
Old Dominion University

Kurt Maly
Old Dominion University

Mohammad Zubair
Old Dominion University

Michael L. Nelson
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/computerscience_fac_pubs

 Part of the [Computer Sciences Commons](#), and the [Digital Communications and Networking Commons](#)

Repository Citation

Liu, Xiaoming; Maly, Kurt; Zubair, Mohammad; and Nelson, Michael L., "Arc - An OAI Service Provider for Digital Library Federation" (2001). *Computer Science Faculty Publications*. 1.
https://digitalcommons.odu.edu/computerscience_fac_pubs/1

Original Publication Citation

Liu, X., Maly, K., Zubair, M., & Nelson, M.L. (2001). Arc-an oai service provider for digital library federation. *D-Lib Magazine*, 7(4), 1-16. doi: 10.1045/april2001-liu

ARTICLES

D-Lib Magazine
April 2001

Volume 7 Number 4

ISSN 1082-9873

***Arc* - An OAI Service Provider for Digital Library Federation**[Xiaoming Liu](#)[Kurt Maly](#)[Mohammad Zubair](#)

Old Dominion University

Norfolk, Virginia USA

liu_x@cs.odu.edu, maly@cs.odu.edu, zubair@cs.odu.edu[Michael L. Nelson](#)

NASA Langley Research Center

Hampton, Virginia USA

m.l.nelson@larc.nasa.gov

Abstract

The usefulness of the many on-line journals and scientific digital libraries that exist today is limited by the inability to federate these resources through a unified interface. The Open Archive Initiative (OAI) is one major effort to address technical interoperability among distributed archives. The objective of OAI is to develop a framework to facilitate the discovery of content in distributed archives. In this paper, we describe our experience and lessons learned in building *Arc*, the first federated searching service based on the OAI protocol. *Arc* harvests metadata from several OAI compliant archives, normalizes them, and stores them in a search service based on a relational database (MySQL or Oracle). At present we have over 320,000 metadata records from 18 data providers from various subject domains. We have also implemented an OAI layer over *Arc*, thus making hierarchical harvesting possible. The experiences described within should be applicable to others who seek to build an OAI service provider.

1. Introduction

The lack of interoperability is one of the significant problems that digital libraries (DLs) currently face. The inability to federate, filter and provide value-added services on remote content limits DLs to covering only local holdings. The Open Archive Initiative (OAI) ([Lagoze & Van de Sompel, 2001](#)) is one major effort to address technical interoperability among distributed archives. The objective of OAI is to facilitate the discovery of content in distributed archives. OAI differs from other interoperability approaches, such as Z39.50 ([Lynch, 1997](#)), SDLIP ([Paepcke, et al., 2000](#)) or NCSTRL ([Leiner, 1998](#)), through its emphasis on a limited, simple, and easy to implement protocol that layers over an existing repository. The OAI framework defines two functional roles: data providers (archives) and service providers. Service providers extract metadata from data providers via the OAI metadata harvesting protocol. The service provider develops value-added services that are based on the metadata collected from data providers. These value-added services could take the form of cross-archive search engines, linking systems, and peer-review systems. OAI is becoming widely accepted, and there are many archives currently or soon-to-be OAI compliant.

Arc (<http://arc.cs.odu.edu>) is the first federated search service based on the OAI protocol. It originates from the Universal Preprint Service (UPS) prototype ([Van de Sompel, Krichel, Nelson, et al., 2000](#)), which was developed as a proof-of-concept and discussion piece for various DL technologies, including the feasibility of constructing a cross-archive searching service. UPS contained nearly 200,000 records harvested from six archives using NCSTRL+ ([Nelson, Maly, Shen & Zubair, 1998](#)), a modified version of the Dienst protocol ([Davis & Lagoze, 2000](#)). Constructing a DL the size of UPS uncovered a number of scalability problems with both file system storage and search engine. To address these issues, we re-implemented the core NCSTRL+ services using Java Servlets and an Oracle RDBMS ([Maly, Zubair, Anan, et al., 2000](#)). Once the OAI metadata harvesting protocol stabilized, it was possible to realize the vision of UPS in *Arc*, with a higher performance search capability and contents kept up to date through the OAI protocol.

In *Arc*, we also implement an experimental OAI layer over harvested data. Thus, one service provider can collect information from both data providers and service providers. By retrieving information from other service providers, service providers can also cascade indexed views from one another -- using the service provider's query interface to filter or refine the information from one service provider to the next.

We encountered a number of problems in developing *Arc*. Different archives have different format/naming conventions for specific metadata contents, thus necessitating data normalization. Arbitrary harvesting can overload the data provider making the data provider unusable for normal purposes. The data providers' security protection can block the crawler and make harvesting difficult to implement. Initial harvesting when a data provider joins a service provider requires a different technical approach than periodical harvesting that keeps the data current.

So far there has been a great variability in data providers' publicly disclosed implementations. We hope that service provider implementations will also enjoy this diversity. The discussion of the system architecture, harvesting and metadata processing experiences and the general lessons and observations reported in this paper are not designed to be the definitive guide on building OAI service providers. Rather, they are presented in the interest of illuminating some of the (now) known pitfalls that await future implementers.

2. Architecture

The *Arc* architecture is based on the Java servlets-based search service that was developed for the Joint Training, Analysis and Simulation Center (JTASC) ([Maly, Zubair, Anan, et al., 2000](#)). This architecture is platform independent and can work with any web server. Moreover, the changes required to work with different databases are minimal. Our current implementation supports two relational databases, one in the commercial domain (Oracle), and the other in the public domain (MySQL). The architecture improves performance by employing a three-level caching scheme. [Figure 1](#) outlines the major components: Search Engine, Harvester, and an OAI layer over *Arc* for hierarchical harvesting.

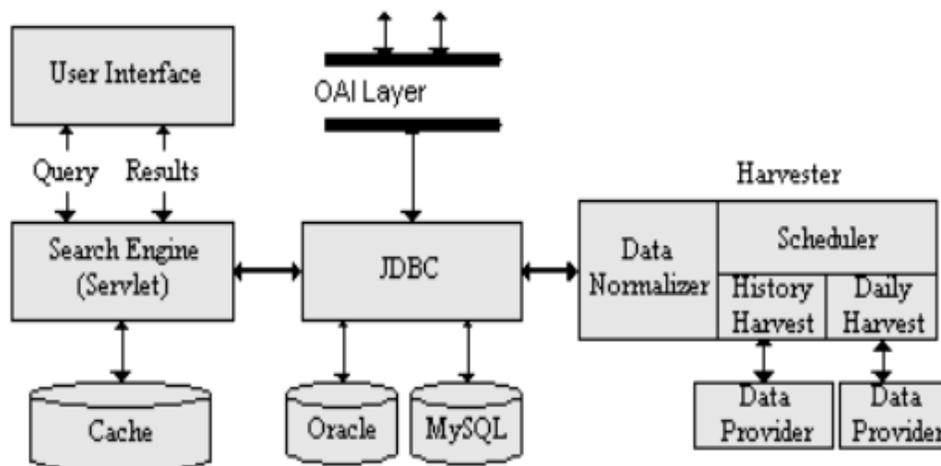


Figure 1. *Arc* Architecture

2.1 Harvester

Similar to a web crawler, the *Arc* harvester traverses the data providers automatically and extracts metadata. The significant differences include: normalizing the metadata, thus producing more complete and accurate results; and exploiting the incremental, selective harvesting defined by the OAI protocol.

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 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 protocol 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 defined four harvesting types for *Arc*:

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 a great deal of network traffic. However, bulk harvesting has two problems. First, the data provider may not implement the resumptionToken flow control mechanism of the OAI metadata harvesting protocol, and thus may not be able to correctly process large (but partial) data requests. Secondly, XML syntax errors and character-encoding problems -- these were surprisingly common -- can invalidate entire large data sets.

One-by-one harvesting is used when bulk harvesting is infeasible. However, this approach imposes significant network traffic overhead for both the 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 daemon written in Java and running on a Windows NT computer. 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 periodically checks and starts the appropriate task.

2.2 Hierarchical Harvesters

We have also implemented two experimental OAI layers on *Arc* that we feel demonstrate the flexibility of the OAI metadata harvesting protocol. The first experimental layer (<http://arc.cs.odu.edu:8080/oai/dp/index.jsp>) allows *Arc* to act as a data provider, disseminating metadata harvested from other data providers (Figure 2). This allows for the hierarchical harvesting of content, similar to the system of gathers and brokers defined in Harvest (Bowman, Danzig, Hardy, et al., 1994). This structure has a great deal of flexibility in how information is filtered and interconnected between data providers and service providers. For example, one service provider might index papers in computer science, while another could build a general scientific service by harvesting the existing computer science harvester. Hierarchical harvesting also could provide the mechanism for caching and replication services.

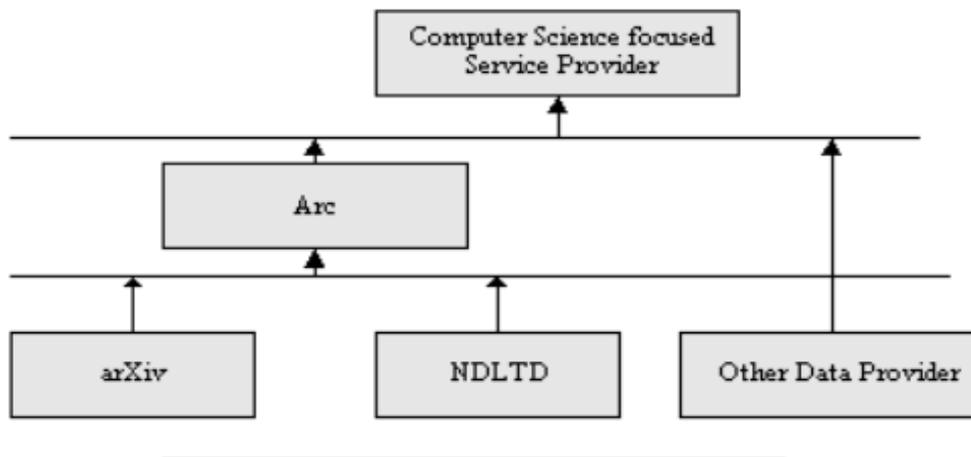


Figure 2. Hierarchical Harvesting

A service provider normalizes harvested data. Thus, the data harvested from *Arc* might not be the same data *Arc* harvested from the data providers, which can introduce both intellectual property and provenance issues. The document id is the one unique metadata item that should be kept in all locations to allow for tracking the source of the document. In *Arc*, we save all the original information sent through the OAI protocol.

A second interface (<http://arc.cs.odu.edu:8080/oai/sp/index.jsp>) causes *Arc* to use the OAI metadata harvesting protocol to describe the archives from which it harvests. That is, instead of the OAI records corresponding to records from the data providers, the records returned from this *Arc* interface describe the actual archives themselves. This interface was implemented to provide a dynamic and machine-readable mechanism for discovering the data providers from which a service provider harvests. It should be noted that both of these experimental OAI interfaces are not "official" uses of the OAI metadata harvesting protocol. Both are subject to change pending further experiments, and neither are required for others who seek to build OAI service providers.

2.3 End-User Search Facility

2.3.1 Database Schema

OAI uses unqualified Dublin Core (DC) ([Weibel, Kunze, Lagoze & Wulf, 1998](#)) as the default metadata set, and all *Arc* services are implemented on the data provided in the DC fields. All DC attributes are saved in the database as separate fields. The archive name and sets 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 ([Oracle, 2001](#)) and MySQL full-text search ([MySQL, 2001](#)). The search engine communicates with the database using JDBC ([Reese, 2000](#)) and Connection Pool ([Moss, 1999](#)).

2.3.2 Search Server Implementation

The search server is implemented in Java using Servlets. The components of the search server are shown in [Figure 3](#).

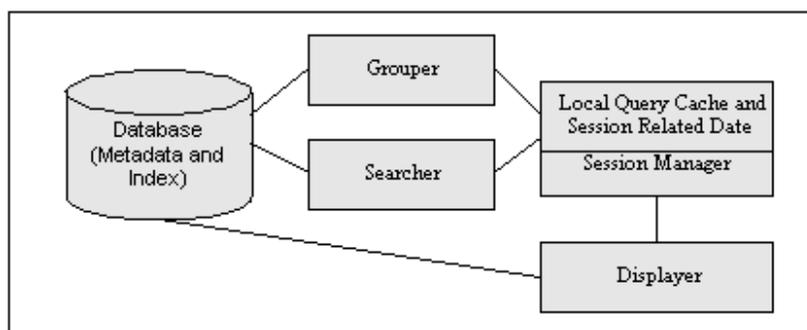


Figure 3. Information Retrieval Process.

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 a large number of results (hits) that cannot be displayed on one page. Thus sessions are used to cache results in order to make 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).

2.3.3 Search Interface Specification

The search interface supports both simple and advanced searching as well as results sorting by date stamp, relevance ranking and archive. Simple searching allow users to search free text across archives. Advanced searching ([Figure 4](#)) allows users to search in specific metadata fields. Users can also search/browse specific archives and/or archive partitions if they are familiar with specific data providers. Author, title, and abstract searches are based on user input, and the input can include Boolean operators (AND, OR, NOT). Archive, set, type, language and subject fields use controlled vocabularies, which are accumulated from the participating archives' source data.

For search results sorting, there is a pull down menu for either type of searching that allows specifying the sorting of search results. Search results can be sorted by rank, datestamp, or archive. For the search result group, there is a pull down menu for choosing the grouping of results. Search results may be grouped according to archive, year of datestamp and subject.

In the search result page ([Figure 5](#)), the left panel shows all groups and hits numbers, and the right panel shows summary information about each document in the selected group. The user can also traverse different pages if multiple search pages exist. When users are interested in a document, they can view the detail page ([Figure 6](#)), and follow the link to the full text document that resides in the data provider's repository. A demonstration of *Arc* is available in [Appendix 1](#).

ARC Cross Archive Search Service - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://arc.cs.odu.edu:8080/oai/servlet/search?formname=searchform> Go

Y! Messenger Y! Bookmarks My Yahoo! Yahoo! Finance Y! Mail News

Search specific bibliographic fields

Creator

Title

Description

Combine fields with AND OR **search**

Filter options

Archive

Archive's Set

Subject

Type

Language

DateStamp (yyyy/mm/dd)

Discovery Date (yyyy/mm/dd)

Display options

Sort results by

Group results by

search

Done Internet

Figure 4. Advanced Search Interface.

ARC - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print

Address <http://arc.cs.odu.edu> Go

Y! Messenger Y! Bookmarks My Yahoo! Yahoo! Finance Y! Mail News Shopping

arc Cross Archive Searching Service 

Simple search Advanced Search Help

Matches were found in these archives

archive	Hits
LTRS	10
NDLTD	1
arXiv	273
etdcat	46
idli	20
mit.etheses	5

This is page 1, hits (1--10) of total 273 hits.

Results Pages: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [Next >>]

SEARCH RESULTS

Title	Jacobi polynomials of type BC, Jack polynomials, limit transitions and $O(\infty)$
Creators	Koorwinder, Tom H.
Description	This is an extended abstract of a lecture held at the Conference "Fourier and Radon transformations on symmetric spaces in honor of Professor S. Helgasons 65th birthday, Roskilde, Denmark, Sept. 10--12, 1992.
Archive	arXiv
Date Stamp	1993-07-09
Document ID	oai:arXiv:math.CA/9307216

This prototype is based on the JISC project and the NCSTDL based digital library developed by Old Dominion University.

Internet

Figure 5. Search Results, Grouped by Archive.

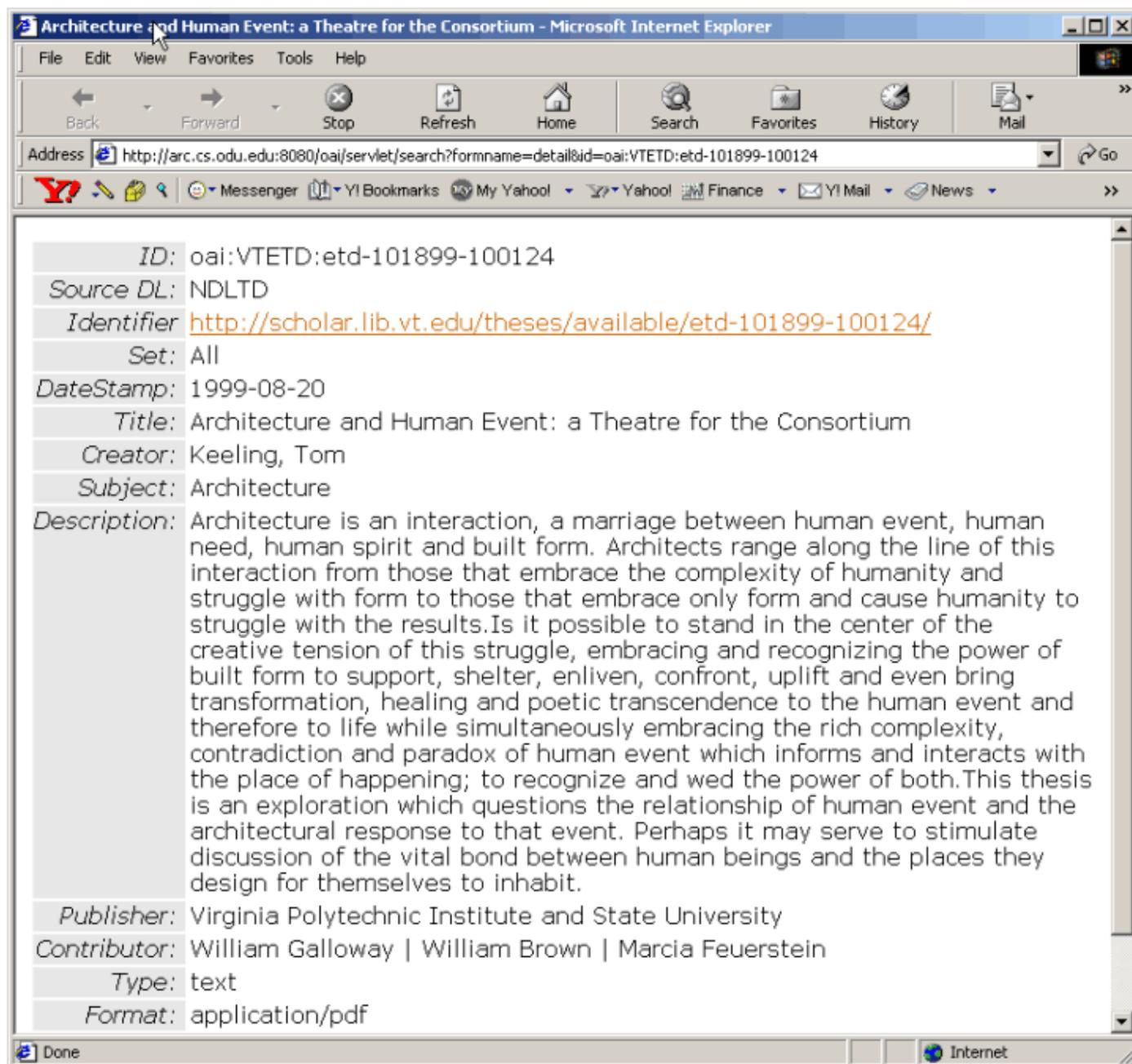


Figure 6. An Individual Record from the Result Set.

3. Results

3.1 The Harvested Records in Arc

We collected approximately 30 data providers from the OAI homepage and other resources. These sources cover several different communities, ranging from museum to e-prints collections. We reviewed them individually and selected eighteen data providers simply to get a representative sample.

Table 1. Collections Harvested by Arc (by March 22, 2001).

Archive Name	URL	Records

arXiv.org e-Print Archive	http://arxiv.org/	155231
California International and Area Studies Digital Repository	http://eprints.cdlib.org	12
Cognitive Science Preprints	http://cogprints.soton.ac.uk/	1024
Humboldt University of Berlin Document Server	http://dochost.rz.hu-berlin.de/	347
Library of Congress: American Memory	http://memory.loc.gov/	3784
M.I.T. Theses	http://theses.mit.edu/	5037
NASA Langley Technical Report Server	http://techreports.larc.nasa.gov/ltrs/	2323
National Advisory Committee for Aeronautics	http://naca.larc.nasa.gov/	6352
NCSTRL (Cornell portion only)*	http://www.ncstrl.org	2080
NSDL Open Archives Server at Cornell University	http://siteforscience.nsdlib.org	2536
OCLC Online Computer Library Center Theses and Dissertations Repository	http://www.oclc.org/home/	95434
Open Video	http://www.open-video.org/	183
Perseus Digital Library	http://www.perseus.tufts.edu/	1009
PhysNet, Oldenburg, Germany, Document Server	http://physnet.uni-oldenburg.de/	9467
Resource Discovery Network	http://www.rdn.ac.uk/	387
University of Illinois at Urbana-Champaign, Digital Library Initiative	http://images.library.uiuc.edu/projects/DCHC/	38065
Virginia Tech ETD Initiative	http://etd.vt.edu/	2412
Web Characterization Repository*	http://researchsmp2.cc.vt.edu/cgi-bin/reposit/index.pl?	131
<i>Total Records</i>		325814

* = supports only an older version of the OAI protocol

3.2 Update Frequency of Data Providers

The synchronization problem -- how to keep the metadata records of data providers and those in *Arc* consistent -- is another problem that can distort the results a user obtains from a search. The user must trust that the service provider has an accurate assessment of the contents of the data providers that it harvests. The OAI protocol supports selective, incremental and scheduled harvests. Service providers are expected to exploit these properties in order to limit the load imposed on the data providers while

still maintaining fresh data for their services. The frequency of new or modified records available through the data provider plays a major role in determining the balance between harvesting too often and not enough. The nature of the data provider can influence how often records are modified or updated. E-print type data providers are likely to have a small but steady stream of ongoing daily or weekly updates. Museum or historically oriented archives will have an initial bursty period of accession (perhaps all at once), but then are likely to trickle down to just infrequent error corrections or edits. Although not currently implemented by any data providers, if a data provider allowed the metadata to change based on usage, annotations or reviews, the required harvesting would likely become significant. [Figure 7](#) lists the daily number of modified or added records for all of *Arc*'s holdings. Some data providers do not provide correct datestamp information and are not considered in this graph. [Figure 8](#) lists monthly accession of records in the e-print archives.

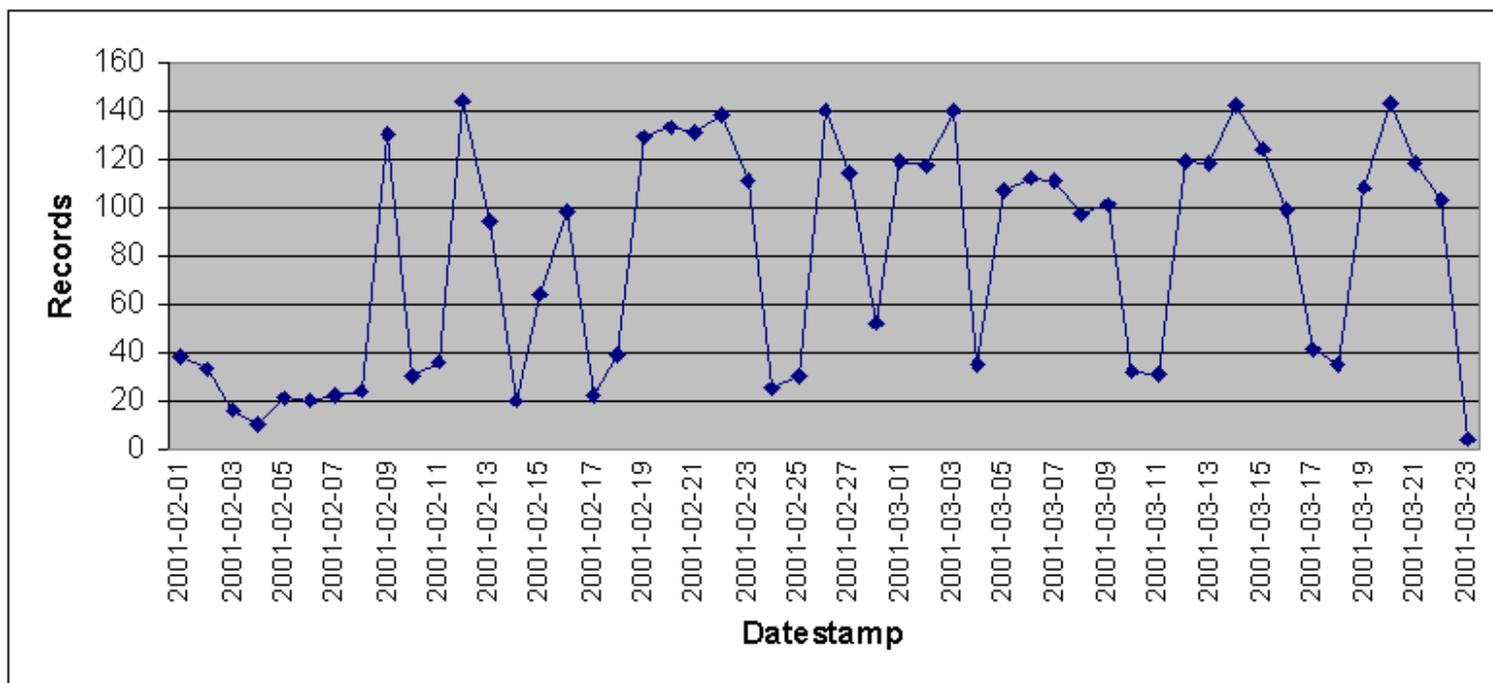


Figure 7. New or Modified Records Added to Arc.

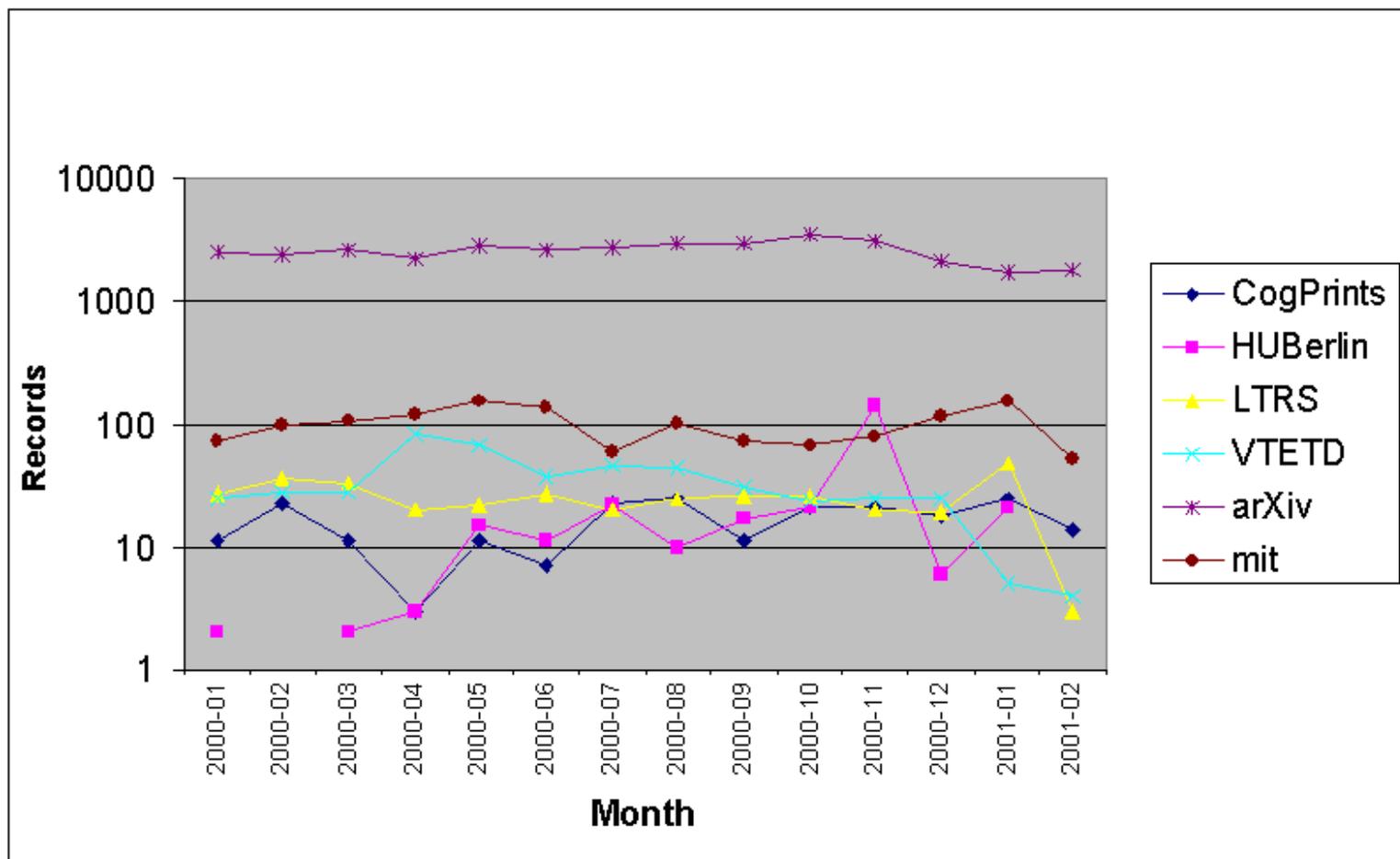


Figure 8. Monthly Accessions for E-Print Servers.

3.3 Controlled Vocabulary and Authority Files

A critical factor in implementing generic search parameters is the availability of controlled vocabularies. Making controlled vocabularies available in a search interface to constrain a search will greatly help users construct more precise and reliable queries.

However, we found that it is difficult to use controlled vocabularies in a service provider. The quality of the metadata we harvested is extremely variable. Attempting to perform more than a simple Boolean search across multiple OAI archives can yield inconsistent results.

In an effort to create a unified search interface for cross archive searching, we analyze the subject, type, publication date and language fields. These fields are defined in DC and we had assumed that they would be easier to normalize as the basis of a cross archive search service. In [Table 2](#), we list the results. The subject level represents the structure of the subject system; it could be two levels, one level or no subject information available. We also list the number of distinct language, subject and type fields used in each archive, and in most circumstances, each archive certainly has its own understanding and/or value definition for these fields. [Appendix 1](#) illustrates how the controlled vocabularies based on the values extracted from the metadata fields are incorporated into the advanced search interface.

Table 2. Metadata variability in Dublin Core fields (March 22, 2001)

Archive	"Subject" Field	Number of Unique Values in the	Number of Unique Values	Consistent Format in the

	Level	Number of Unique Values	"Language" Field	in the "Type" Field	"Date" Field?
ArXiv	2	123	10	1	Yes
Cogprints	2	64	N/A	10	Yes
Huberlin	1	252	3	1	Yes
LOC	1	1012	17	7	No
NACA	N/A	N/A	N/A	1	No
MIT	1	1171	N/A	1	Yes
NCSTRL	N/A	N/A	N/A	1	Yes
LTRS	N/A	N/A	N/A	1	No
NDLTD	1	223	1	1	Yes
NSDL-CU	1	1694	9	9	No
Open Video	N/A	N/A	1	1	Yes
OCLC	**	**	269	35	Yes
Perseus	N/A	N/A	5	N/A	N/A
PhysNet	N/A	N/A	13	1	Yes
RDN	1	601	24	N/A	N/A
Idli	N/A	N/A	4	3	N/A
WCR	N/A	N/A	N/A	N/A	N/A

***Data not yet collected*

Analysis of the data harvested from the data providers showed that the variability in the data is mainly a product of:

- misspelling and/or the absence of authority control in the local systems, and
- the use by the data providers of different authority files, such as subject classification methods.

From this table, one might conclude that it would be difficult or impossible to create a browsing interface for *Arc* that would allow users to browse the combined collection according to a particular metadata-filed value. We are exploring the use of approximate word matching and other algorithms ([French, Powell, Schulman, & Pfaltz, 1997](#)) to improve the relation between what the user expects and what *Arc* actually delivers.

4. Lessons Learned and Proposed Solutions

Little is known about the long-term implications of a harvest-based DL. Construction of this prototype demonstrated several issues that are likely to recur in any attempt to build an OAI service provider. The effort of maintaining a quality federation service is highly dependent on the quality of the data

providers. Some are meticulous in maintaining exacting metadata records that need no corrective actions. Other data providers have problems maintaining even a minimum set of metadata and the records harvested are useless.

4.1 Data Provider and Metadata Quality

During the testing of data harvesting from OAI data providers, numerous problems were found. We discovered that not all archives strictly follow the OAI protocol; many have XML syntax and encoding problems; and some data providers are periodically unavailable. Many OAI responses were not well-formatted XML files. Sometimes foreign language and other special characters were not correctly encoded. XML syntax errors and character-encoding problems were surprisingly common and could invalidate entire large data sets. Incremental harvesting proved beneficial as a work-around.

The OAI website validates registered data providers for protocol compliance. It uses XML schemas to verify the standard conformance. However, this verification is not complete; it does not cover the entire harvesting scenario and does not verify the entire data set. Additionally, such verification cannot detect semantic errors in the protocol implementation, such as misunderstanding of DC fields. For certain XML encoding errors, an XML parser can help avoid common syntax and encoding errors. If the data provider builds quality control and data cleaning into its local accession policy ([Suleman, Fox & Abrams, 2000](#)), the service provider will have significantly less work to do and will have to discard fewer dirty data records.

4.2 Update Frequency, Push Model and Security

Due to the variability of size and frequency of updates in DLs, we also faced a trade-off in the frequency of harvests: too many harvests could over-burden both the service and data providers, and too few harvests allow the data in the service provider to potentially become stale. We believe the synchronization problem is a major problem of the harvesting model -- maintaining data coherency between the data providers and service providers. In many ways, the complexity introduced by the synchronization problem is the price that has to be paid by avoiding the common problems of distributed searching.

The OAI harvesting model is built on service providers "pulling" metadata from a set of data providers. However, it is possible to extend the harvesting model to include "push" or even hybrid "push/pull" models for data harvesting. For example, if a harvester had a large number of data providers that only occasionally updated their holdings, the harvester might wait for an external event (automated email, or even a phone call to a human) before re-scheduling a harvest of that data provider. Another scenario in which a push-based model could be appropriate is to overcome institutional firewall or other security restrictions. Similar to the method use in Gnutella ([Oram, 2000](#)), a data provider behind a firewall could establish a connection with the service provider outside firewall and send updates directly to the service provider, negating the need for a service provider to establish an in-bound connection with a possibly restricted machine.

Setting up a harvesting schedule is not an entirely automated process. Security is deliberately not directly addressed in the OAI protocol, but security can be attached to data providers via standard http mechanisms of usernames/passwords and host based access. This also requires out of band communication between the service provider and the data provider to either set the correct authentication need for a particular data provider, or to inform the data provider of the hostname that the harvests will originate from. The OAI protocol is extremely flexible in that although it does not require this level of sophistication between service and data providers, it does not prevent complex arrangements from being constructed where needed.

Although most data providers will not require such measures to be taken, it is important to note that some do. While this creates some additional work for the service providers, it does provide a measure of confidence to the data providers that establishing an OAI interface to their repository does not equate to

loss of control of how and who harvests their metadata.

4.3 The Availability of Data Providers

The stability and service from data providers are difficult to predict since many factors may influence data provider availability and efficiency. Previous studies have shown that total availability of distributed repositories is difficult to maintain (Powell & French, 2000). Because it is expected that service providers in OAI shall support high quality service, such instability must be considered and solved by both service providers and data providers. So far, data provider unavailability has not been a serious issue. However, if the number of data providers grows to hundreds or even thousands, approaches will have to be designed for maintaining high availability of metadata (during harvesting) and data (during user sessions).

4.4 Controlled Vocabulary

Some normalization was necessary to achieve a minimum presentation of query results. However, we did so on an ad hoc basis with no formal definition of the relationship mappings. A controlled vocabulary will be of great help for a cross-archive search service to define such metadata fields as "subject". The variation of fields in different DLs is caused by both spelling variants and archive submission policies. Spelling variations can be addressed through the construction of authority files. However, there is a limit to the quality of services that can be offered on metadata from archives that allow free text entries from contributors for fields such as "subject", "type" and "language".

5. Conclusions

The contribution of *Arc* is to prove not only that an OAI-compliant service provider can be built, but also that one can be built at a scale previously unrealized within the e-print community. The Open Archives Initiative has been successful in getting data providers to adopt the protocol and provide an OAI layer to their repositories. In addition to the data providers registered on their website, there are many more being used in localized and non-public applications. However, to date most of the services that interact with these data providers have focused on tools to help with the creation of data providers (such as the Repository Explorer (Suleman, 2001)). *Arc* is the first service provider to focus on providing DL-type services to the user, and is based on the original design goals of the Universal Preprint Server. In addition to providing a vehicle to learn the long-term implications of running an OAI service provider, *Arc* will also provide a large collection of metadata for additional experimentation and services. The latest version of *Arc* is accessible at <<http://arc.cs.odu.edu>>. Future focus will not only include increasing the breadth of *Arc* coverage as new data providers become available, but also on increasing the depth and richness of the services and user experience. In the meantime it is becoming a very useful tool to study the quality and usefulness of metadata in a variety of digital libraries.

[Appendix 1: A Demo of Arc](#)

References

- Bowman, C. M., Danzig, P. B., Hardy, D. R., Manber, U., Schwartz, M. F. & Wessels, D. P. (1994). *Harvest: A scalable, customizable discovery and access system*. Technical Report CU-CS-732-94, Department of Computer Science, University of Colorado, Boulder. Available at <<ftp://ftp.cs.colorado.edu/pub/cs/techreports/schwartz/Harvest.Jour.ps.Z>>.
- Davis, J. R. & Lagoze, C. (2000). NCSTRL: design and deployment of a globally distributed digital library. *Journal of the American Society for Information Science*, 51(3), 273-280.
- French, J. C., Powell, A. L., Schulman, E. & Pfaltz, J. L. (1997). Automating the construction of authority files in digital libraries: a case study. In C. Peters & C. Thanos (eds.), *Research and advanced*

technology for digital libraries, first European conference, ECDL '97 (pp. 55-71), Berlin: Springer.

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.

Leiner, B. (1998). The NCSTRL Approach to Open Architecture for the Confederated Digital Library. *D-Lib Magazine*, 4(12). Available at <<http://www.dlib.org/dlib/december98/leiner/12leiner.html>>.

Lynch C. (1997). The Z39.50 Information Retrieval Standard: Part I: A Strategic View of Its Past, Present and Future. *D-Lib Magazine*, 3(4). Available at <<http://www.dlib.org/dlib/april97/04lynch.html>>.

Maly, K., Zubair, M., Anan, H., Tan, D. & Zhang, Y. (2000). Scalable digital libraries based on NCSTRL/Dienst. *Proceedings of the Fourth European Conference on Digital Libraries - ECDL 2000* (pp. 169-179), Lisbon, Portugal.

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

MySQL (2001). *MySQL reference manual for version 3.23.36*. Available at <<http://www.mysql.com/doc/home.html>>

Nelson, M. L., Maly, K., Shen, S. N. T., & Zubair, M. (1998). NCSTRL+: adding multi-discipline and multi-genre support to the Dienst protocol using clusters and buckets. *Proceedings of the IEEE forum on research and technology advances in digital libraries* (pp. 128-136), Santa Barbara, CA. Available at <<http://techreports.larc.nasa.gov/ltrs/PDF/1998/mtg/NASA-98-ieeeedl-mln.pdf>>.

Oracle. (2001). Oracle InterMedia Server. Available at <http://otn.oracle.com/docs/products/oracle8i/doc_index.htm>.

Oram, A. (2000). Gnutella and Freenet represent true technological innovation, *O'Reilly Network*. Available at <<http://www.oreillynet.com/pub/a/network/2000/05/12/magazine/gnutella.html>>.

Paepcke, A., Brandriff, R., Janee, G., Larson, R., Ludaescher, B., Melink, S. & Raghavan, S. (2000). Search Middleware and the Simple Digital Library Interoperability Protocol. *D-Lib Magazine*, 6(3). Available at <<http://www.dlib.org/dlib/march00/paepcke/03paepcke.html>>.

Powell, A. L. & French, J. C. (2000). Growth and server availability of the NCSTRL digital library. *Proceedings of the Fifth ACM Conference on Digital Libraries* (pp. 264-265), San Antonio, TX. Available at <<http://www.cs.virginia.edu/~cyberia/papers/DL00.pdf>>.

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

Suleman, H., Fox, E. A. & Abrams, M. (2000). Building quality into a digital library. *Proceedings of the Fifth ACM Conference on Digital Libraries* (pp. 228-229), San Antonio, TX.

Suleman, H. (2001). *Open Archives Initiative Repository Explorer v1.0*. Available at <http://purl.org/net/oai_explorer>.

Van de Sompel, H., Krichel, T., Nelson, M. L., Hochstenbach, P., Lyapunov, V. M., Maly, K., Zubair, M., Kholief, M., Liu, X. & O'Connell, H. (2000). The UPS Prototype: An Experimental End-user Service across E-print Archives. *D-Lib Magazine*, 6(2). Available at <<http://www.dlib.org/dlib/february00/vandesompel-ups/02vandesompel-ups.html>>.

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>>.

Copyright 2001 Kurt Maly, Mohammad Zubair, and Xiaoming Liu. (Although he is a co-author of this article, Michael Nelson is not listed as a copyright holder because his work on this project was done as an employee of the U.S. Federal Government.)

[Top](#) | [Contents](#)
[Search](#) | [Author Index](#) | [Title Index](#) | [Back Issues](#)
[Letters to the Editor](#) | [Next Article](#)
[Home](#) | [E-mail the Editor](#)

[D-Lib Magazine Access Terms and Conditions](#)

[DOI: 10.1045/april2001-liu](#)