DRTC Workshop on
Digital Libraries: Theory and Practice
March 2003
DRTC, Bangalore

Paper: H

The Open Archives Initiative Protocol for Metadata Harvesting: An Introduction

Saiful Amin
Documentation Research and Training Centre
Indian Statistical Institute
Bangalore-560 059
email: *saifulamin@yahoo.com*

Abstract

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is a collaborative effort that provides an application-independent interoperability framework based on metadata harvesting. Though the OAI-PMH is a very recent development it is being regarded as an important step towards information discovery in the digital library arena. This paper looks into the issues leading to its development as well as gives an inside view of the proposed model.

1. INTRODUCTION

The OAI-PMH is a means of making machine-readable metadata widely available for use. The Open Archives Initiative was originally proposed to enhance access to e-print/pre-print archives. Gradually, however, the scope of the initiative has broadened to cover any kind of digital content including images and videos. It is available to all regardless of economic mechanism surrounding the content.

2. HISTORY OF OAI

The origin of OAI can be traced back to the efforts to increase interoperability among the e-print/pre-print servers that hosted scientific and technical papers (Breeding, 2002). A number of factors led to the development of the pre-print archives most important of which was the rising cost of journals. Scholars and researchers would deposit their articles and papers into these servers, which allow for the dissemination of information among the scholarly community much more rapidly than through traditional print journals.

The number of e-print/pre-print repositories was growing steadily in the nineties. This growth created an information overload and some other problems, which can be summarized as:

- The end-users/scholars may not be able to know the existence of a repository.
- Overlapping of coverage in terms of subjects.
- Multi-disciplinary nature of subjects needed the documents to be kept at a number of repositories.
- Discipline-specific and institution-specific archives created duplication efforts.
- The end-users/scholars had to search individual repositories to get documents of his interest.
- Also, it was undesirable to require scholars to deposit their work in multiple repositories.

Need was felt to build a framework to bring about a kind of integration of these e-print/pre-print archives to solve these problems. A meeting was convened in late 1999 at Santa Fe, New Mexico to address problems of the e-print world. The major work was to define an interface to permit e-print servers to expose their metadata for the papers it held, so that search services or other similar repositories could then harvest its metadata. These archives would then act as a federation of repositories by giving a single search platform for multiple collections.

After the meeting, the agreed principles were launched in January 2000 as the Open Archives Initiative specification by Herbert Van de Sompel, Rick Luce, and Paul Gisparg among others. The Digital Library Federation, the Coalition for Networked Information, and the National Science Foundation sponsored it.

The OAI Steering Committee was formed in August 2000 to give the strategic direction to the protocol. The protocol version 1.1 was launched in July 2001. The Open Archives Initiative Technical Committee (OAI-TC) was formed to develop and write version 2 of the Open Archives Initiative Protocol for Metadata Harvesting based on feedback from implementers. The OAI-PMH version 2.0 was eventually released in June 2002 (*http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm*).

3. OAI VS. Z39.50

There was a debate as to why not use the existing Z39.50 protocol, which is also used for the search and transfer of metadata. The OAI's metadata-harvesting approach might look operationally much different to the Z39.50, but both achieve what's often called "federated

searching." The federated searches allow users to gather information from multiple related resources through a single interface.

The basic difference between the two protocols is in the search approach. The Z39.50 allows clients to search multiple information servers in a single search interface in *real time,* whereas the OAI-PMH allows bulk transfer of metadata from the repositories to the Service Providers' database. Hence the clients do not need search multiple data providers in real time rather they search the metadata database of the Service Provider who collect and aggregate the metadata from different data providers.

There were many reasons to have a completely new protocol rather than implementing the Z39.50 as it stands. Some of the reasons are:

- Z39.50 is a mature, sophisticated, but unfortunately very complex protocol. It can be used as a tool to build federated search systems; in such a system, a client sends a search in parallel to a number of information servers that comprise the federation, and then gathers the results, eliminates or clusters duplicates, sorts the resulting records and presents them to the user.
- It has been proven that it is very difficult to create high-quality federated search services across large numbers of autonomous information servers through Z39.50 for several reasons.
- Retrieval accuracy is a problem: different servers interpret Z39.50 queries differently, in part due to lack of specificity in the standard, leading to semantic inconsistencies as a search is processed at different servers.
- There are scaling problems in the management of searches that are run at large numbers of servers; one has to worry about servers that are unavailable (and with enough servers, at least one always will be unavailable), and performance tends to be constrained by the performance of the slowest individual server participating in the federation of servers.
- Compromising speed of access since the user has to wait for a lot of record transfer and post-processing before seeing a result, making Z39.50-based federated search performance sensitive to participating server response time, result size, and network bandwidth.

The open archives committee adopted a model that rejected distributed search in favor of simply having servers provide metadata in bulk for harvesting services, subject only to some very simple scoping criteria, such as providing all metadata added or changed since a specified date, or all metadata pertaining to papers meeting matching gross subject partitions within an archive (Lynch, 2001).

Implementing PMH is very simple since one does not need a different port like Z39.50 (which uses port 210). It works over the HTTP, which any web server listens, and any web browser or web-downloader talks. It means one can use common Linux programs such as wget or curl to harvest the metadata from repositories. One does not need a special toolkit (like Yaz for Z39.50).

According to Lynch (2001) "*These two protocols are really meant for different purposes, with very different design parameters, although they can both be used as building blocks in the construction of similar services, such as federated searching. Neither is a substitute for the other […] and we should not think about the world becoming partitioned between Z39.50-based resources and MHP-speaking resources, but rather about bridges and gateways.*"

4. METADATA STANDARDS AND OAI-PMH

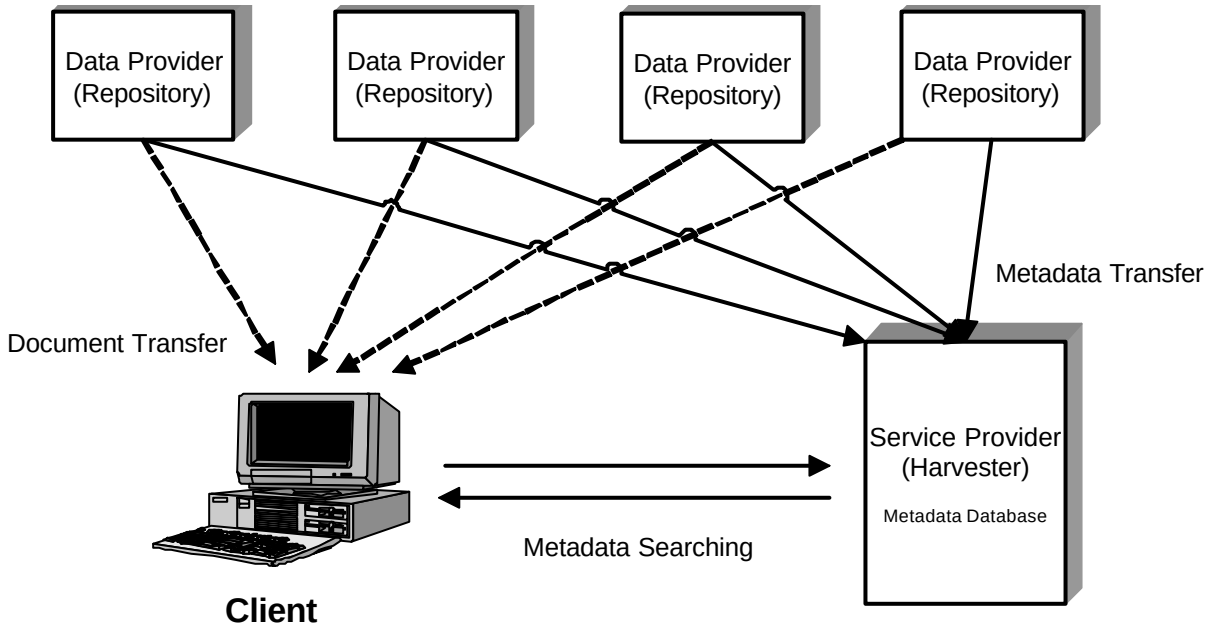
For the purpose of interoperability, the OAI Protocol for Metadata Harvesting specifies unqualified Dublin Core, encoded in XML, a mandatory metadata schema as the lowest common denominator. It is certainly clear that almost any metadata scheme can be "downgraded" into unqualified Dublin Core. However, each server is also free to offer metadata in one or more other schemas, and a harvester can request that metadata in any format in addition to the unqualified Dublin Core.

The *ListMetadataFormats* request will return the *metadataPrefix*, schema, and optionally a *metadataNamespace*, for either a particular record or for the whole repository (if no identifier is specified). In the case of the whole repository, all metadata formats supported by the repository are returned. It is not implied that all records are available in all formats.

5. THE OAI-PMH FRAMEWORK

There are two classes of participants in the OAI-PMH framework:

- **Data Providers:** Data Providers, or repositories, administer systems that support the OAI-PMH as a means for exposing their metadata. Here *data* means any kind of digital content, including text, images, sound, and multimedia.
- **Service Providers:** Service Providers, or harvesters, use metadata harvested via the OAI-PMH as a basis for building value-added services, such as building subject gateways, email alerts, etc.



The OAI-PMH Architecture

The metadata stored in the data providers' database is transferred in bulk to the metadata database of the service providers. The transfer of metadata is done in a series of requests and responses between the data provider and the service provider/harvester. The OAI-PMH Protocol depends upon the HTTP-transaction framework for communication between a harvester and a repository. Requests may be made using either the HTTP GET or POST methods. All successful replies are encoded in XML, and all exception and flow-control replies are indicated by HTTP status codes.

5.1. Request Verbs

When a service provider makes a request to the data provider they must use one of the six requests, also known as "verbs", defined by the protocol:

- ***Identify:*** is used to retrieve information about a repository. It gives submission policies, copyright notices, administrator email, etc.
- ***ListMetadataFormats:*** is used to retrieve the metadata formats available from a repository.
- ***ListSets:*** is used to retrieve the set structure in a repository. It is particularly useful for multi-disciplinary and inter-disciplinary repositories to use sets to allow set-based selective harvesting.
- ***ListIdentifiers:*** is used to retrieve the identifiers of records that can be harvested from a repository. It can be called the smaller version of the *ListRecords* request in the sense that it retrieves only the header of the records instead of the entire record.
- ***ListRecords:*** is used to harvest records from a repository.
- ***GetRecord*:** is used to retrieve an individual record from an item in a repository.

5.2. HTTP Request Format

OAI-PMH requests must be submitted using either the HTTP GET or POST methods. POST has the advantage of imposing no limitations on the length of arguments. Repositories must support both the GET and POST methods. There is a single *baseURL* for all requests. The base URL specifies the Internet host and port, and optionally a path, of an HTTP server acting as a repository. Repositories expose their base URL as the value of the *baseURL* element in the *Identify* response.

In addition to the base URL, all requests consist of a list of keyword arguments, which take the form of *key=value* pairs. Arguments may appear in any order and multiple arguments must be separated by ampersands [&]. Each OAI-PMH request must have at least one *key=value* pair that specifies the OAI-PMH request issued by the harvester. The first *key* is invariably the string 'verb' and the *value* is one of the six defined OAI-PMH requests.

5.2.1. The Keys and their Values

The number and nature of additional *key=value* pairs depends upon the arguments for the individual request.

- *identifier* – The key *identifier* identifies a particular record in the repository. Each *identifier* is unique to the repository in the sense that it can represent only one record. The verb *ListIdentifiers* gives the entire list of identifiers available for harvesting in the repository. The *identifier* key is a compulsory argument for the *GetRecord* request verb. An *identifier* has three sections separated by an indicator, which normally is a colon (:). The three sections are respectively the protocol name (e.g., *oai*), the *repositoryIdentifier* (e.g., arxiv), and a unique identifier for a document within the repository whose format is decided by the individual repository or data provider.
  e.g., &identifier=oai:arxiv:hep-th/9901001
  Here "oai" is the protocol name, "arxiv" is the *repositoryIdentifier*, and "hep-th/9901001" is the unique identifier for the particular document in the repository.
- *metadataPrefix* – The key *metadataPrefix* indicates the metadata format (like MARC, Dublin Core, etc.) in which the record is requested. The verb *ListMetadataFormats* gives the list of metadata formats supported by a repository or data provider.
  e.g., &metadataPrefix=oai_dc
  It means the request is limited to the Dublin Core metadata format.

- *resumptionToken* – The use of *resumptionToken* is discussed in the section on flow control.
- *from* and *until* – These two argument keys are used in combination for date-based harvesting. It will be discussed in the section of selective harvesting.
- *set* – The *set* argument is used for set-based harvesting and will also be discussed in the section on selective harvesting. However, set-based harvesting is not supported by all the repositories.

For example,
http://arxiv.org/oai2?verb=GetRecord&identifier=oai:arXiv.org:cs/0112017&metadataPrefix=oai_dc

5.3. Response Format

Once the harvester has sent a request, the server returns a series of sets of XML-encoded metadata elements (i.e., title, authors, etc) as well as identifiers for objects that the metadata describes in the form of a record. A record is an XML-encoded byte stream that is returned by a repository in response to an OAI protocol request for metadata from an item in that repository. The URL of a metadata schema identifies each metadata format that is included in a record disseminated by the OAI protocol within the repository by a metadata prefix. The metadata schema is an XML schema that may be used as a test of conformance of the metadata included in the record (Shearer, 2002).
Responses to requests are formatted as HTTP responses, with appropriate HTTP header fields. The Content-Type returned for all OAI-PMH requests must be text/xml.

5.3.1. XML Response Format

All responses to OAI-PMH requests must be well-formed XML instance documents. Encoding of the XML must use the UTF-8 representation of Unicode. Character references, rather than entity references, must be used. Character references allow XML responses to be treated as stand-alone documents that can be manipulated without dependency on entity declarations external to the document.
The XML data for all responses to OAI-PMH requests must validate against the XML Schema given at http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd. All responses to OAI-PMH requests should have the following common markup:
1. The first tag output is an XML declaration where the version is always 1.0 and the encoding is always UTF-8.
eg: <?xml version="1.0" encoding="UTF-8" ?>
2. The remaining content is enclosed in a root element with the name OAI-PMH. This element must have three attributes that define the XML namespaces used in the remainder of the response and the location of the validating schema:

- *xmlns* -- the value of which must be the namespace URI of the OAI-PMH (http://www.openarchives.org/OAI/2.0/).
- *xmlns:xsi* -- the value of which must be the namespace URI for XML schema (http://www.w3.org/2001/XMLSchema-instance).
- *xsi:schemaLocation* -- is a pair, the first part of which is the namespace URI (as defined by the XML namespace specification) of the OAI-PMH (http://www.openarchives.org/OAI/2.0/), and the second part is the URL of the XML schema for validation of the response (http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd).

3. For all responses, the first two children of the root element are:

- ***responseDate*** -- a UTCdatetime indicating the time and date that the response was sent. This must be expressed in UTC.

- ***request*** -- indicating the protocol request that generated this response. The rules for generating the request element are as follows:
    - The content of the request element must always be the *baseURL* of the protocol request;
    - The only valid attributes for the request element are the *keys* of the *key=value* pairs of protocol request. The attribute values must be the corresponding values of those *key=value* pairs;
    - In cases where the request that generated this response did not result in an error or exception condition, the attributes and attribute values of the request element must match the *key=value* pairs of the protocol request;
    - In cases where the request that generated this response resulted in a *badVerb* or *badArgument* error condition, the repository must return the *baseURL* of the protocol request only. Attributes must not be provided in these cases.

4. The third child of the root element is either:
    - an error element that must be used in case of an error or exception condition;
    - an element with the same name as the verb of the respective OAI-PMH request.

An example of a successful reply to the *GetRecord* request is as shown below:

1.	<?xml version="1.0" encoding="UTF-8" ?>
2.	<OAI-PMH xmlns=http://www.openarchives.org/OAI/2.0/ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
3.	<responseDate>2003-02-24T18:03:00Z</responseDate> <request verb="GetRecord" metadataPrefix="oai_dc" identifier="oai:arXiv.org:cs/0112017">http://arXiv.org/oai2</request>
4.	<GetRecord> <record> […] </record> </GetRecord>
2.	</OAI-PMH>

5.3.2. Metadata Record (XML Format)

A record is returned in an XML-encoded byte stream in response to an OAI-PMH request for metadata from an item. A record is identified unambiguously by the combination of the unique identifier of the item from which the record is available, the *metadataPrefix* identifying the metadata format of the record, and the datestamp of the record. The XML-encoding of records is organized into the following parts:

5.3.2.1.Header

The header section of the record contains the unique identifier of the item and properties necessary for selective harvesting. The header consists of the following parts:

- the unique identifier -- the unique identifier of an item in a repository;
- the datestamp -- the date of creation, modification or deletion of the record for the purpose of selective harvesting.
- zero or more *setSpec* elements -- the set membership of the item for the purpose of selective harvesting.

5.3.2.2.Metadata

The metadata section is a single manifestation of the metadata from an item. The OAI-PMH supports items with multiple manifestations (formats) of metadata. At a minimum, repositories

must be able to return records with metadata expressed in the Dublin Core format, without any qualification. Optionally, a repository may also disseminate other formats of metadata. The specific metadata format of the record to be disseminated is specified by means of an argument -- the *metadataPrefix* -- in the *GetRecord* or *ListRecords* request that produces the record. The *ListMetadataFormats* request returns the list of all metadata formats available from a repository, or for a specific item (which can be specified as an argument to the *ListMetadataFormats* request).

Header	<record> <header> <identifier>oai:arXiv.org:cs/0112017</identifier> <datestamp>2003-02-05</datestamp> <setSpec>cs</setSpec> </header>
Metadata	<metadata> <oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/ http://www.openarchives.org/OAI/2.0/oai_dc.xsd"> <dc:title>Using Structural Metadata to Localize Experience of Digital Content</dc:title> <dc:creator>Dushay, Naomi</dc:creator> <dc:subject>Digital Libraries</dc:subject> <dc:subject>H.3.7</dc:subject> <dc:description> With the increasing technical sophistication of both information consumers and providers, […] </dc:description> <dc:description>Comment: 23 pages including 2 appendices, 8 figures</dc:description> <dc:date>2001-12-14</dc:date> <dc:type>text</dc:type> <dc:identifier>http://arXiv.org/abs/cs/0112017</dc:identifier> </oai_dc:dc> </metadata> </record>

The example shown above is an XML-encoding of a record and its components:
1. The *header* part with:

- a unique identifier of the item from which the record was disseminated, equal to oai:arXiv:cs/0112017;
- the datestamp of the record equal to 2001-12-14;
- the *setSpec* value **cs** indicates that the item from which the record was disseminated belongs to only one set of the repository;

2. The *metadata* part. This consists of a single root tag - in the example the tag oai_dc:dc - with the nested tags belonging to the corresponding metadata format -- in the example, Dublin Core elements such as dc:title. Note that the root tag within the metadata part includes a number of attributes that are common to all XML documents that use namespaces and schema validity:

- *namespace declarations* -- the declarations of the namespaces used within the metadata part, each of which is prefixed with xmlns . Namespace declarations within the metadata part fall into two categories:

- *metadata format specific namespace(s)* - every metadata part must include one or more xmlns prefixed attributes that define the correspondence between a metadata format prefix -- e.g. dc -- and the namespace URI (as defined by the XML namespace specification) of the respective metadata format. Some metadata formats employ tags from multiple namespaces, requiring multiple xmlns prefixed attributes -- in the example, there are declarations for both oai_dc and dc.
- *xml schema namespace* - every metadata part must include the attribute xmlns:xsi, the value of which must always be the URI shown in the example, which is the namespace URI for XML schema.
- *xsi:schemaLocation* -- the value of which is a URI, URL pair; the first is the namespace URI (as defined by the XML namespace specification) of the metadata that follows in this part, and the second is the URL of the XML schema for validation of the metadata that follows.

6. SELECTIVE HARVESTIING

Harvesters can also limit the metadata to be returned by applying restrictions based on two relatively simple criteria:

Date-based: Harvesters may use datestamps to harvest only those records that were created, deleted, or modified within a specified date range. To specify datestamp-based selective harvesting, datestamps are included as values of the optional arguments, *from* and *until*, in the *ListRecords* and *ListIdentifiers* requests.

Example:

http://arxiv.org/oai2?verb=ListRecords&from=20021112&untill=20030212&metadataPrefix=oai_dc

Set-based: Harvesters may specify set membership as a criterion for selective harvesting. To specify set-based selective harvesting, a *setSpec* is included as the value of the optional set argument to the *ListRecords* and *ListIdentifiers* requests, thereby specifying selective harvesting of records from items within the respective set.

Example:

http://rocky.dlib.vt.edu/~jcdlpix/cgi-bin/OAI/jcdlpix.pl?verb=ListRecords&set=200105dle&metadataPrefix=oai_dc

7. FLOW CONTROL AND THE *RESUMPTIONTOKEN*

One of the concerns with the PMH model involves how a service provider can obtain large numbers of metadata records from a data provider without overburdening the system. The way that metadata records are transferred remains under the control of the data provider.

Flow control is supported with the HTTP retry-after status code 503. This allows a server (data-provider) to tell the harvesting agent (service-provider) to try the request again after some interval. It is left entirely up to the server implementer to determine the conditions under which such a response will be given. The server could base the response on current machine load or limit the frequency at which requests will be serviced from any given IP address. The retry-after response may also be used to handle temporary outages without simply taking the server off-line. In an environment where one of a set of servers may handle a request, the server may dynamically redirect a request using the HTTP 302 response.

The PMH takes into consideration that the data provider will have preferences regarding when it will want to respond to harvester and how many records it will deliver in a given time. PMH includes a control mechanism called a *Resumption Token.* At any time, a data provider's server

can return an incomplete set or records in response to a request, issuing a *resumptionToken***.** To retrieve the next portion of the complete list the next request must use the value of that *resumptionToken* element as the value of the *resumptionToken* argument of the request. Optionally, this token may be valid for a certain period of time only mentioned as *expirationDate*.

7.1. Exception Condition and Error Handling

The OAIMH protocol has very simple exception handling: syntax errors result in HTTP status code 400 replies, and parameters that are invalid or have values that do not match records in the repository result in empty replies. For example, a *ListRecords* request for a date range when there were no changes, or for a metadata format not supported, will result in a reply with header information but no <*record*> elements (Shearer, 2002).

8. SOME EXISTING DATA PROVIDERS

As discussed earlier the Data Providers are repositories or archive of a digital content with some kind of metadata describing the content. The Data Providers expose their metadata, by installing a piece of software, in such a manner that harvesters can harvest their metadata to build value added services.

8.1. ArXiv E-Print Archive

Description: ArXiv is an e-print service in the fields of physics, mathematics, non-linear science and computer science. The contents of arXiv conform to Cornell University academic standards. arXiv is owned, operated and funded by Cornell University, a private not-for-profit educational institution. ArXiv is also partially funded by the National Science Foundation.
Homepage: http://arxiv.org/
Base URL: http://arXiv.org/oai2

8.2. E-Prints in Library and Information Science (E-LIS)

Description: E-LIS is an electronic open access archive for scientific or technical documents, published or unpublished, in Librarianship, Information Science and Technology, and related application activities. E-LIS is an archive to deposit preprints, postprints and other LIS publications, finding and downloading documents in electronic format, offered as a free service to the international LIS community. The goal of the E-LIS Archive is to promote communication in the field by the rapid dissemination of papers.
Homepage: http://eprints.rclis.org/
Base URL: http://eprints.rclis.org/perl/oai2

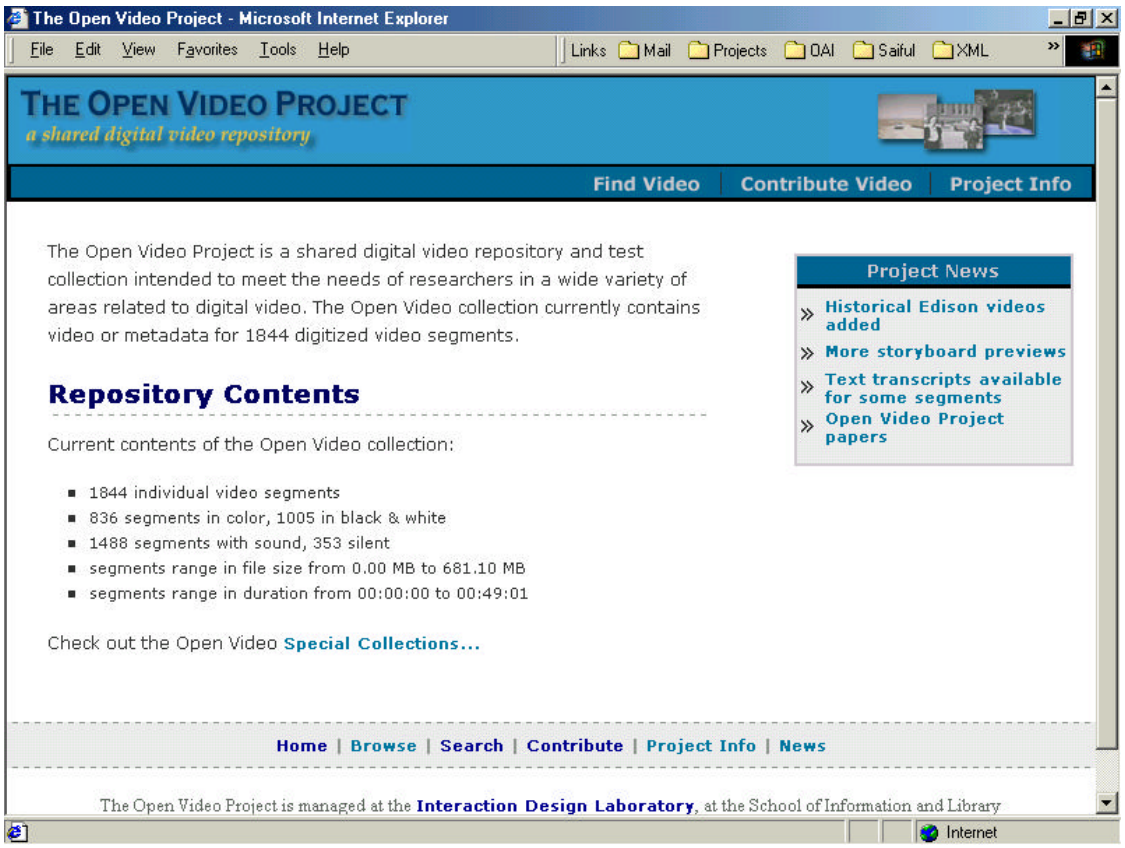
8.3. CogPrints

Description: Cognitive Sciences E-print Archive. An electronic archive for self-archive papers in any area of Psychology, neuroscience, and Linguistics, and many areas of Computer Science (e.g., artificial intelligence, robotics, vision, learning, speech, neural networks), Philosophy (e.g., mind, language, knowledge, science, logic), Biology (e.g., ethology, behavioral ecology, sociobiology, behaviour genetics, evolutionary theory), Medicine (e.g., Psychiatry, Neurology, human genetics, Imaging), Anthropology (e.g., primatology, cognitive ethnology, archeology, paleontology), as well as any other portions of the physical, social and mathematical sciences that are pertinent to the study of cognition.
Homepage: http://cogprints.ecs.soton.ac.uk/
Base URL: http://cogprints.ecs.soton.ac.uk/perl/oai2

8.4. Open Video Project

Description: The Open Video Project is a shared digital video repository and test collection intended to meet the needs of researchers in a wide variety of areas related to digital video. The

Open Video collection currently contains video or metadata for 1844 digitized video segments. (Accessed on February 20, 2003).
Homepage: http://www.open-video.org/
Base URL: http://www.open-video.org/oai2.0/



9. SOME EXISTING SERVICE PROVIDERS

As mentioned earlier, the Service Providers harvest the metadata exposed by the Data Providers. Their job is similar to the web-crawlers of the Internet search engines. They go to the individual repositories to harvest their entire metadata, collects in its database in the XML format. The collected metadata is then parsed to provide an integrated search interface and browsing indices to the collections of all the participating data providers/repositories.

9.1. OAIster

Description: OAIster is a project of the University of Michigan Digital Library Production Services, originally funded through a Mellon grant. Our goal is to create a collection of freely available, difficult-to-access, academically-oriented digital resources that are easily searchable by anyone.
Homepage: http://oaister.umdl.umich.edu/o/oaister/

9.2. Networked Computer Science Technical Reference Library

Description: The Networked Computer Science Technical Reference Library (NCSTRL - pronounced as "ancestral") is an international collection of computer science research reports made available for non-commercial use from over 100 participating organizations worldwide. The organizations that participate in NCSTRL include Ph.D. granting computer science

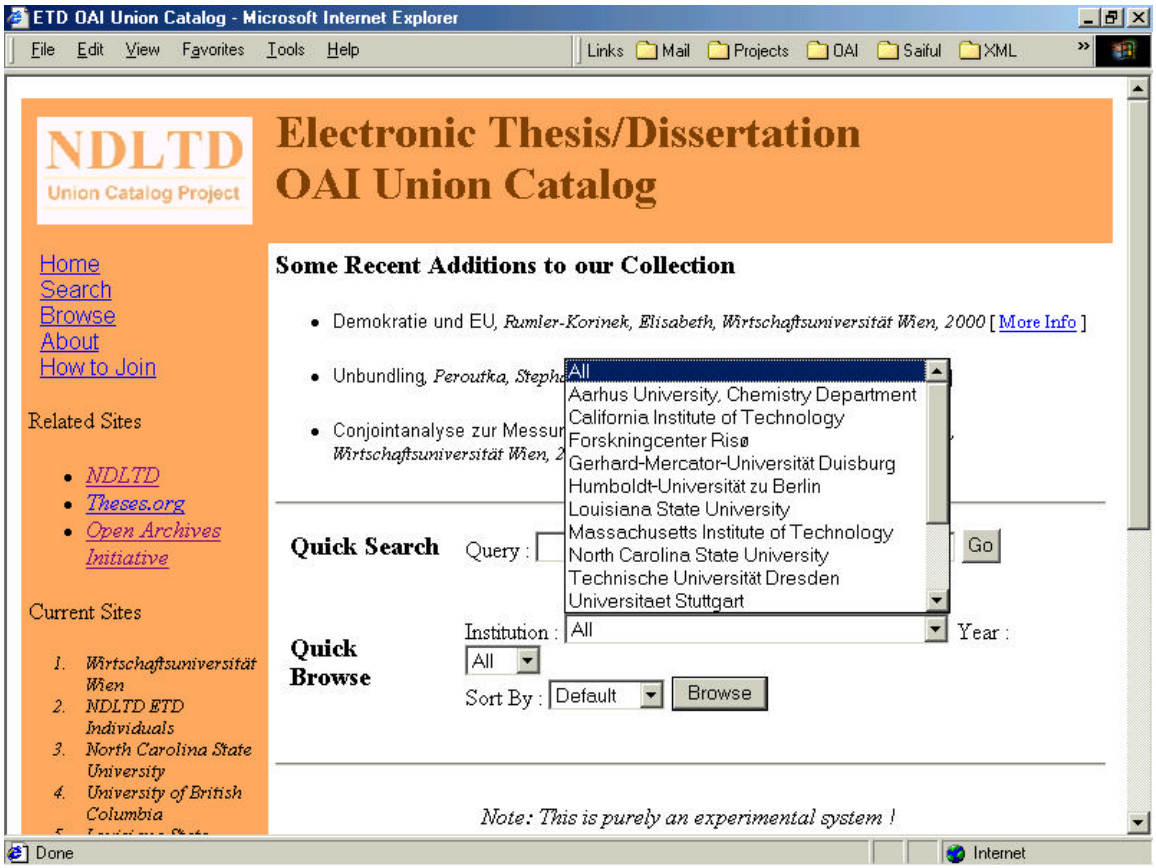
departments, research laboratories, ePrint repositories, and electronic journals. The documents in NCSTRL are almost all textual, ranging in size from 100-plus page doctoral dissertations to short technical reports.
Homepage: http://www.ncstrl.org

9.3. iCite: CITATION INDEXING

Description: iCite is a citation indexing service based on OAI-PMH by Scuola Internazionale Superiore di Studi Avanzati (SISSA, International School for Advanced Studies), Italy. It allows searching 3613394 citations in 150984 documents (as on February 20, 2003).
Homepage: http://icite.sissa.it:8888/icite/

9.4. Electronic Thesis/Dissertation OAI Union Catalog

Description: This is a service built by harvesting metadata from Open Archives of electronic theses and dissertations. The underlying technology is based on layered Open Archives with data being harvested from source archives and then stored in a Union Catalog. This Union Catalog is then front-ended with a search engine for demonstration purposes, but the data is just as easily accessible to other service providers, both local and remote.
Homepage: http://rocky.dlib.vt.edu/~etdunion/cgi-bin/index.pl



10. CONCLUSION

The growth and proliferation of digital media has been growing faster than ever. No digital library can be self-sufficient, even if it is involved in a narrow field of study. Thus the digital libraries need to share their resources. Authorities have already started to see the benefits of the

networked digital libraries. Interoperability has been the main hurdle in effective sharing of resources between digital libraries over a network. The OAI Protocol for Metadata Harvesting achieves interoperability by very simple means.

The aim of the Open Archives Initiative had been to promote the accessibility of scholarly material through the development of universal interoperability standards. The scope of the protocol has gradually broadened to the domain of digital libraries. With the release of the version 2.0 of the protocol it has started showing the signs of maturing. It not only covers the various text document formats but image, video, audio, and multimedia as well.

There are still a number of large-scale archives, such as PubMedCentral, that are not exposing their metadata using the OAI Protocol for Metadata Harvesting. However the number of OAI compliant repositories has been rising steadily. The simplicity and the ease of implementation has been the main strength of this protocol. It promises be a major force in effective utilization of digital archives and popularization of digital libraries.

11. REFERENCES

1. Breeding, M. (2002, April). The Emergence of the Open Archives Initiative: This Protocol could become a key part of the digital library infrastructure. *Information Today*. from http://www.findarticles.com/cf_0/m3336/4_19/85251474/p1/article.jhtml
2. Breeding, M. (2002). Understanding the Protocol for Metadata Harvesting of the Open Archives Initiative. *Computers in Libraries*, 22(8).
3. Lagoze, C., & Sompel, H. V. d. (2001, January). *The Open Archives Initiative Protocol for Metadata Harvesting*, from http://www.openarchives.org/OAI/openarchivesprotocol.htm
4. Lynch, C. A. (2001, August). Metadata Harvesting and the Open Archives Initiative. *ARL Bimonthly Report 217*. from http://www.arl.org/newsltr/217/mhp.html
5. Shearer, K. (2002, March). The Open Archives Initiative: Developing an Interoperability Framework for Scholarly Publishing. *CARL/ABRC Background Series*, No. 5. from http://www.carl-abrc.ca/projects/scholarly/open_archives.PDF
6. Suleman, H., & Fox, E. A. (2001, December). A Framework for Building Open Digital Libraries. *D-Lib Magazine*, 7(12). from http://www.dlib.org/dlib/december01/suleman/12suleman.html
7. Sompel, H. V. d., & Lagoze, C. (2000, February). The Santa Fe Convention of the Open Archives Initiative. *D-Lib Magazine*, 6(2). from http://www.dlib.org/dlib/february00/vandesompel-oai/02vandesompel-oai.html
8. Warner, S. (2001, June). Exposing and Harvesting Metadata Using the OAI Metadata Harvesting Protocol: A Tutorial. *HEP Libraries Webzine* Issue 4. from http://library.cern.ch/HEPLW/4/papers/3/