A Scalable Architecture for Harvest-Based Digital Libraries - The ODU/Southampton Experiments

Xiaoming Liu Tim Brody Stevan Harnad Les Carr
Kurt Maly Mohammad Zubair Michael L. Nelson

May 28, 2002

Abstract

This paper discusses the requirements of current and emerging applications based on the Open Archives Initiative (OAI) and emphasizes the need for a common infrastructure to support them. Inspired by HTTP proxy, cache, gateway and web service concepts, a design for a scalable and reliable infrastructure that aims at satisfying these requirements is presented. Moreover it is shown how various applications can exploit the services included in the proposed infrastructure. The paper concludes by discussing the current status of several prototype implementations.

1 Introduction

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is an important new infrastructure for supporting distributed networked information services. OAI promotes interoperability through the concept of metadata harvesting. The OAI framework supports Data Providers (DPs) and Service Providers (SPs). SPs develop value-added services based on the information collected from DPs. DPs are simply collections of harvestable metadata that may or may not contain additional services and content. This paper introduces the concepts of OAI-PMH proxies, OAI-PMH caches, OAI-PMH gateways as tools for the optimization of the functioning of the DP/SP model underlying the OAI-PMH.

[1] X.Liu, K.Maly, M.Zubair are with Computer Science Department,Old Dominion University, USA

[2] T.Brody, S.Harnad, L.Carr are with Intelligence, Agents, Multimedia Group, Electronics and Computer Science, University of Southampton, UK

[3] M.L.Nelson is with NASA Langley Research Center, USA

The OAI-PMH uses HTTP-based request-response communication between a DP and an SP. The XML-formatted metadata is encoded in the HTTP response, which makes on-demand services possible. Using OAI-PMH, one DP may be harvested by any number of SPs, each possibly implementing different services. These service providers can interoperate using the multiple-resolution capability (one identifier is resolved to multiple instances) based on unique identifiers. In OAI-PMH, the metadata is distributed and replicated in many different places and potentially provides a highly redundant and fault-tolerant system.

The Old Dominion University (ODU) Digital Library Group and the University of Southampton have been engaged in research focused on various OAI-PMH services. The goal of this research is to achieve interoperability, scalability and reliability of OAI-PMH services.

Over the past two years, ODU has developed the Arc [22], Archon [1], Kepler [24], and DP9 [23] services. Arc and Archon are federated searching services based on the OAI-PMH; they focus on the process of building a unified search interface over heterogeneous collections. Kepler is an end-user software that allows the individual to easily create and maintain a small, OAI-compliant archive. DP9 is a gateway service that allows general search engines (e.g. Google, Inktomi, etc.) to index OAI-PMH-compliant archives.

As part of the JISC/NSF Open Citation Project [27][19] (and previously the Open Journals Project), the University of Southampton Intelligence, Agents, Multimedia Group (IAM) has researched and developed methods for citation linking and analysis of the refereed scientific literature. Recent efforts have focused on CiteBase Search [14]. CiteBase is an impact-ranking OAI-PMH SP, providing a federated searching service, citation linking (resolving author-provided references to their OAI-PMH-available targets), and citation analysis. CogPrints [2] is an archive, hosted at Southampton, that allows authors to archive their articles (e-prints) for free in a freely accessible web service [18][2]. The software behind CogPrints, ePrints.org [16][4], has been developed by Southampton and released under Open Source. Now anyone, from individuals to institutions, may create their own self-archiving repositories. ePrints.org is fully OAI-PMH-compliant.

In these efforts we notice challenges faced by OAI-PMH based applications, namely:

Data Provider and Metadata Quality During the testing of DPs, numerous problems were found. Not all archives strictly follow the OAI-PMH; many have XML syntax and encoding problems. The UPS

Prototype [33] also found significant problems with metadata, however, with OAI-PMH, syntax for metadata is strictly defined (XML schema validation), problems still appear. This suggests that some DPs do not strictly check their OAI-PMH implementations.

Server Availability The stability and service from DPs are difficult to predict since many factors may influence DP availability and efficiency [26]. If a large DP is periodically unavailable, this can be a serious problem for harvesting. A recent research points out that a significant number of DPs could not be harvested in the testing period [20].

Scalability OAI-PMH harvesting is resource-expensive to DPs, partially because the HTTP responses are dynamically generated, and DPs may need to keep current harvest sessions (harvesting may take several days for a large data set). Besides steps taken by individual DPs to improve services, a general infrastructure is required.

Linking Across Service Providers In OAI-PMH, several DPs may be harvested by many SPs, each providing different services for the same records. Cross-service linking and data sharing can be achieved by using the unique OAI identifiers. Unique identifiers also allow the detection of record duplication.

In this paper we discuss a joint effort between ODU and Southampton that addresses these problems using a variety of techniques. We present an architecture to achieve interoperability, scalability, and reliability by optimizing dataflow in DPs/SPs. This architecture introduces an OAI-PMH Proxy concept that could improve DP quality by fixing implementation problems just in time. An OAI-PMH Cache service improves data availability and avoids bottlenecks through Hierarchical Harvesting. An OAI-PMH Gateway translates operations from other resource discovery systems into operations in OAI-PMH and vice versa. We also discuss how to build a series of services such as cross-archive linking based on the suggested architecture.

The remainder of this paper is organized as follows. In Section 2 we discuss related work. In Section 3 we present an overview of the optimized model. We discuss each of the subsystems in Sections 4-8. In Section 9 we discuss several working applications, and we summarize in Section 10.

2 Related work

2.1 Global Research Archive

The driving force behind the development of OAI-PMH was the need for a common method of federating heterogeneous e-print archives into cross-archive search engines and other end-user services, e.g., the UPS prototype [33].

A global research archive can start with e-print archives, perhaps subject-, institution-, or publisher- based. The software e-Prints.org has been developed to support author self-archiving [18]. This and other e-print archives can be harvested to form federated services, potentially a *Global Research Archive*, e.g., Scirus [6], Arc [22], CiteBase [14], and My.OAI [5].

This structured model for federated archives contrasts with CiteSeer [17], which is a web crawler that retrieves research articles from personal and institutional web sites, and automatically builds the bibliographic and reference data from the articles. There is no interoperability model underlying CiteSeer.

2.2 Caching and Replication

HTTP proxy and cache distribute load, reduce network traffic and access latency, and protect the network from erroneous clients [12]. There are two basic approaches for web cache implementation: the passive cache only loads a data object as a result of a client's request to access that object, and an active cache employs some mechanism to prefetch data [21] in advance of a request by a client.

Mirror software is designed to duplicate a directory hierarchy between two machines. It avoids copying files unnecessarily by comparing the file timestamps and file sizes before transferring [25].

2.3 Hierarchical Harvesting

The earlier Harvest project explored the concept of the hierarchical arrangement of object caches and focused on the content extraction for general web documents [10]. After OAI-PMH was released, both Arc and CiteBase explored the issues of hierarchical harvesting in OAI-PMH SPs. The Open Digital Library project [29] and the OAI "result set filtering" whitepaper [34] explored using the OAI-PMH sets concept for OAI-PMH metadata filtering.

2.4 Unique Identifiers

Identifiers are a powerful tool for communication within and between communities. For example, the Handle system [30] and DOI (Digital Object Identifier) [28] provide a mechanism for implementing naming systems for arbitrary digital objects. The multiple-resolution capability becomes important in OAI-PMH community, as metadata may be widely replicated and modified, and many different services will be implemented on the same metadata records. An "intelligent" resolution service should be able to deliver different outcomes to a resolution request dependent on user-specified requirements [32][31].

In OAI-PMH, unique identifier unambiguously identifies an item within a repository, the format of the unique identifier must correspond to that of the URI syntax, Individual communities may develop community-specific URI schemes for coordinated use across repositories.

However, the unique identifiers may conform to a recognized URI scheme with greater scope. Especially the *oai-identifier* schema describes a specific, recommended implementation of unique identifiers which repositories may adhere to; *oai-identifiers* should have global scope and guaranteed global uniqueness. The *oai-identifier* has been widely accepted in implementation of OAI-PMH 1.1 and is further refined in the version 2.0 of the protocol by introducing a globally unique OAI URN. Both the ODU and Southampton implementations use the *oai-identifier* schema and rely on its uniqueness.

2.5 Citation Linking

Citation linking is the general term for hypertext linking the reference lists (the bibliography) in research articles to the cited articles. In recent years, citation linking has been extensively developed [19] [11] [9] [32]. With the wide acceptance of OAI-PMH, new challenges are raised about cross-archive (i.e. cross collection) linking and cross-service linking. With various DPs providing metadata of different qualities and formats, cross-archive linking is necessary to integrate them into one unique linking environment. Similarly, the distributed and highly redundant OAI-PMH architecture allows different services to be built which, with context sensitive and dynamic cross-service linking [32] [31], could potentially be integrated.

Such integrated services might provide citation analysis for forward links (to articles that have referenced the current article), impact factors, co-citation analysis, and novel navigation methods [13].

2.6 Distributed Search vs. Harvest

There are two ways to implement DL interoperability, through distributed search or harvest. The distributed searching approach is studied in NCSTRL, Stanford Infobus [8][15][10]. Currently, the harvesting approach has better scalability and is a realistic approach for a large number of existing DLs. It is studied in Harvest [6] and is the basis of OAI-PMH.

3 System Overview

The need for an optimized model is motivated by the major challenges faced in a basic OAI-PMH model, which can be seen in Figure 1. The basic structure of OAI-PMH supports two roles: the SP and the DP. Multiple SPs may harvest multiple DPs at the same time. If one DP has implementation problems (e.g., XML encoding), all SPs have to address these problems. If one DP is unavailable, all SPs have to wait until the DP comes up again, even if some SPs have already cached the data from the DP.

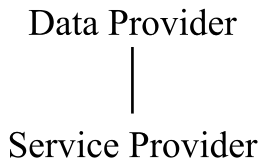

Figure 1: Basic OAI-PMH Harvesting Model

Figure 2 illustrates the optimized model based on the hierarchical harvesting. An OAI-PMH proxy dynamically forwards requests DPs with value-added services. For example, it can dynamically fix common XML encoding errors and translate between different OAI-PMH versions. An OAI-PMH cache caches metadata and can filter and refine them before exposing them to SPs. It also serves as a simple cache that reduces the load on source DPs and improves DP availability. An OAI-PMH gateway can convert the OAI-PMH to other protocols and applications. For example, the gateway could provide value-added services like automatic citation extraction, or conversion between different protocols (SOAP, Z39.50) and OAI-PMH. An end-user service will present various services such as search and citation linking. Figure 2 illustrates how each layer may fetch data from any of its lower layers depending on availability and service type.

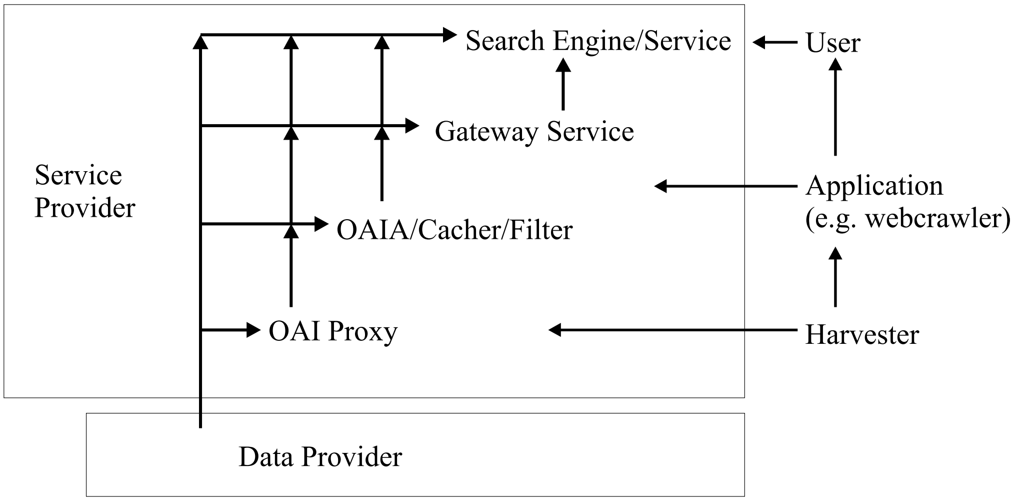

Figure 2: Hierarchical Harvesting Model

4 Data Provider

The OAI-PMH is designed as a low-barrier and structured method for DPs to expose hidden content to add-on services. (We define “hidden” as both data that cannot be accessed via the Web, and data that can only be accessed through proprietary web interfaces)

A DP, using OAI-PMH, exposes metadata for the resources for which it is responsible. Any (or many) metadata formats may be exposed for any given resource, although to be OAI-PMH compliant the DP must support Dublin Core.

A DP does not necessarily provide any added value from the end-user’s perspective; it simply gives OAI-PMH compliant harvesters access to the raw metadata in the repository. However, a richer metadata resource allows more interesting services to be built on top of it. For example, DPs that are collections of research literature (articles with bibliographies), may export the resource’s reference list as well as the current authors, title, and so on.

The first step to building value-added services on repositories is harvesting their metadata. Where there are OAI-PMH implementation faults, SPs must try to work around these problems, or, as we suggest, use a separate layer to fix errors with the DP’s implementation: an OAI-PMH Proxy.

5 OAI-PMH Proxy

From a harvester's point of view, the most convenient solution to incorrectly implemented DPs is to place a layer (i.e. a proxy) over source repositories that can be trusted to provide correct responses to the harvester's requests. The proxy can protect the network from erroneous and malicious clients, for example, a proxy can serve as the single access point for outside world to DPs inside a firewall.

An OAI-PMH proxy can either act as an HTTP proxy or be OAI-PMH-specific. As an HTTP proxy, it effectively becomes a transparent layer accepting HTTP requests and responding with HTTP responses. As an OAI-PMH-specific proxy, it must re-write request URLs; for an example of mapping a given subdirectory to a source baseURL, see Figure 3.

Request URL	Wrapped URL
oai-proxy/cgi/proxy/cogprints	cogprints.soton.ac.uk/perl/oai
oai-proxy/cgi/proxy/bmc	www.biomedcentral.com/oai/1.1/bmcoai.asp

Figure 3: OAI-PMH-Specific Style Proxy Requests

An OAI-PMH proxy will fix the following errors:

Character Encoding OAI-PMH uses the Unicode's UTF-8 character encoding to support international character sets by using multiple bytes for non-English characters. As an OAI-PMH response is received from a repository, the proxy can replace any faulty character encoding that would normally cause an XML parser to fail.

XML Encoding The mark-up characters used in XML must be encoded when used in string data. Similar to recent web browsers, the proxy can use heuristics to determine whether a mark-up character is actually part of mark-up, or should be encoded.

XML Mark-Up XML requires that all arguments to XML elements are quoted, and heuristics can be used to fix some quoting errors. An advanced proxy may be able to fix missing, or out-of-order, XML elements; however, this can easily lead to untraceable errors.

Protocol Errors A proxy can check the validity of the response it receives. It can check that the schemas used in the response are the ones used by the OAI-PMH, and that the XML can be validated against those schemas.

Whereas character and XML encoding problems can be fixed in a stream, more complex errors require caching the entire OAI-PMH response. This requirement leads naturally to an OAI-PMH Cache that is capable of storing and fixing OAI-PMH responses.

6 OAI-PMH Aggregation and Caching

OAI-PMH caching/aggregation (an OAIA) expands the DP/SP model with a middle layer that aggregates/caches responses from source DPs. This feature provides benefits to both aggregated DPs, and SPs that choose to harvest from the OAIA.

6.1 Caching Data Providers

A DP can be harvested by any number of services. Frequent harvesting by many services can result in a significant overhead for providers, who can range from small collections running on simple web-scripts to large multi-million record collections running on powerful databases. A small provider may have inefficient systems or few resources, and a large provider may not appreciate an unreliable harvester continually requesting very large data sets.

An OAIA solves this problem by mirroring DP's metadata, so a few large aggregators (efficiently implemented) can serve many other OAI-PMH services, including downstream aggregators.

6.2 Aggregation

When deciding which repositories to harvest from, an SP must consider whether the repository content is relevant, whether the repository is reliable, and how often the SP should check that repository for updates.

A hierarchical OAIA structure can reduce this problem by avoiding duplication of the efforts of many SPs. For example, an SP may want to provide an index of all the music manuscript repositories of a given country. That SP can then expose the aggregated collection to an international SP, saving the international SP the effort of harvesting from every repository in every country.

6.3 Advantages Over HTTP Caching

An OAIA is similar to an HTTP cache; specifically, they both distribute load away from the server (the DP) and closer to the client (the SP).

An OAIA is, however, an active cache - it requests new records from the known repositories in advance. This means a repository's records will always be available from the cache to downstream harvesters, even if the repository itself is unavailable.

An important role for an aggregator is providing quick access to many, smaller collections. By prefetching records from its source repositories, an OAIA can provide a downstream harvester with all the aggregated records in one session.

6.4 Datestamping

Incremental harvesting in OAI-PMH uses datestamping; that is, a harvester only needs to request records that are new or have changed since the last time it checked the repository.

With hierarchical harvesting the OAIA must update the datestamp when it harvests a record - because the record is "new" to the OAIA. When a downstream harvester harvests from the OAIA, it will receive all the new records in the OAIA, even if the original datestamp of the record was before the date of harvest.

6.5 Identifiers

An OAIA can either maintain or change the *oai-identifiers* for records that it harvests (and reexports).

By maintaining the record's *oai-identifier*, the OAIA can become a nearly transparent layer in a hierarchical system (nearly transparent, because it introduces a delay between a record being created by DPs, and it being harvested from the OAIA). Maintaining *oai-identifiers* allows a harvester to change sources without causing inconsistent records.

OAI-PMH 2.0 introduces a provenance schema for use in the optional "about" field of records ("about fields" are for things that describe the metadata record). This schema allows the history of the record to be recorded: it stores the details of each OAI-PMH service that the record passes through as it goes down the hierarchy, from repository to eventual end-user service.

Provenance can be used to check the originality of the record, identify the change history of the record when it travels around the system, and to extract the original datestamp.

6.6 Identifier Collisions

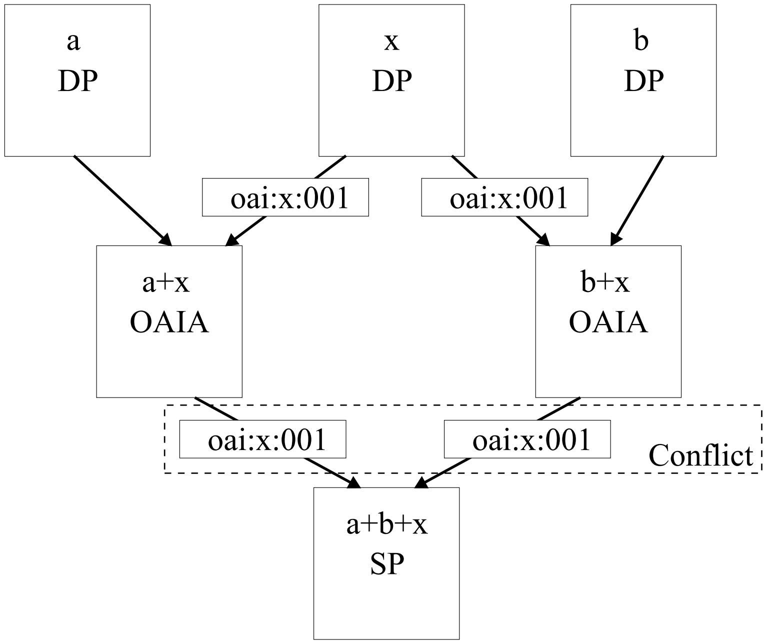

Figure 4: Identifier Conflict in Hierarchical Harvesting

When there is more than one path from a DP to an SP, the SP may need to resolve a collision between two or more records with the same *oai-identifier*.

Figure 4 shows how one record (with a unique *oaiidentifier*), may appear twice to a harvester. Three repositories, a, b, and x, are being harvested by two aggregators, a+x and b+x. When the service provider a+b+x harvests from a+x and b+x, it will get duplicates for every record from the data provider x.

To resolve collisions, a service provider can either store both records or attempt to discard one. The following are some possible policies for record discarding:

Duplicate Records If colliding records are the same, or close, the duplicates could be safely discarded.

Trusted Sources The SP in Figure 4 may, for example, trust OAIA b+x more than OAIA a+x, in which case the SP could discard or overwrite any colliding records harvested from OAIA a+x.

Most Recent It may be possible to distinguish the most recent (and hence most authoritative) record using the datestamps given by the aggregator's provenance data (e.g., OAIAs a+x and b+x in Figure 4).

7 OAI-PMH-Gateway, Value-Added Services

A gateway between two resource discovery systems translates operations from one system into operations in another system. An OAI-PMH Gateway is responsible for converting OAI-PMH for use by other applications and vice-versa. Unlike the OAI-PMH Cache and Proxy, the Gateway service does not necessarily retain the original data or OAI-PMH interface. The objective of a Gateway is to extend OAI-PMH-compliant repositories to other protocols or applications; for example:

Protocol Broker A protocol broker could convert HTTP-based OAI-PMH requests to SOAP messages, or extend OAI-PMH to a Web Service model.

Gateway for Crawlers A gateway for web crawlers could translate OAI-PMH-compliant repositories to a series of linked web pages, which allows web search engines that do not support the OAI-PMH to index the "Deep Web" contained within OAI-PMH-compliant repositories.

Value-Added Services A gateway could cache the full-text document, and then provide value-added services, such as citation extraction, which can then be re-exposed through its own OAI-PMH interface.

Subject Gateway A subject gateway could help build a topic-specific service by harvesting records and then exposing them by subject criteria.

A gateway service may create a large overhead for DPs, especially if the gateway is designed to serve machine-based applications (e.g., web crawlers). This situation is where the OAI-PMH Cache is relevant because the hierarchical structure of OAI-PMH Cache will reduce the overhead for the source DPs to a minimum. At the same time, the gateway service itself could use flow control mechanisms, such as HTTP-throttle software, to reduce the overhead.

8 End-user Service, Search/Citation, Cross Archive Linking Service

There are several considerations for implementing end-user services using the OAI-PMH:

Unique *oai-identifier* The globally unique *oai-identifier* could be a basis for SP/DP interoperability.

Cross-Archive Linking With various DPs providing metadata of different qualities and formats, cross-archive linking is necessary to integrate them into one linking environment.

Cross-Service Linking Many services may be implemented on the same harvested records. A combination of these services will be very useful.

Parallel Metadata Set While OAI-PMH defines Dublin Core as the mandatory metadata format, each provider/community may implement its own metadata sets; supporting this variety of metadata sets is a challenging issue.

Online XML Support DP supports for web presentation of proprietary XML metadata formats.

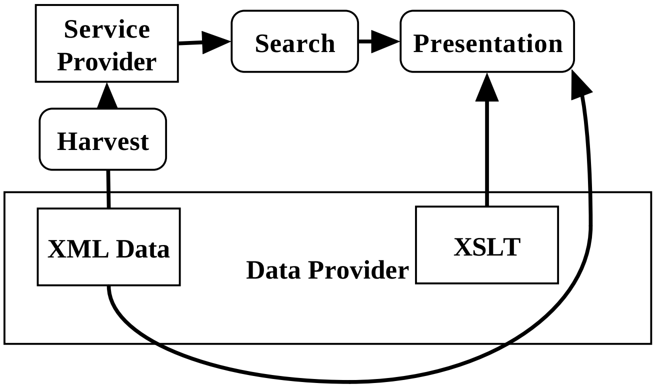

Figure 5: XSLT Processing

Figure 5 shows how to support parallel metadata presentation by XSLT processing. DPs will export their metadata (by OAI-PMH) and presentation format (by XSLT). An SP harvests the metadata and builds a search interface. The resource discovery is performed by the SP, and the final presentation of the data is accomplished by the DP's XSLT. With this mechanism, DPs may define an explicit method for presenting the metadata format, which is especially useful for rarely used or repository-specific metadata formats. OAI-PMH 2.0 introduces a "branding" mechanism for the SP to render the metadata in the stylesheet specified by the DP.

For research literature OAI-PMH can provide a transport mechanism for the bibliographies, as well as metadata. The references can then be harvested by "reference-aware" services that can provide linking services (e.g., by generating a URL to the full-text from the bibliographic reference). Further services can be built on the linked references, such as citation analysis. By aggregating many archives of research literature, one linking service will be able to provide reference links across many small archives.

Currently, there is no widely used XML metadata format in DPs that supports bibliographic references. Many archives, if they support references at all, will only export the text of the reference, and rely on add-on services to parse the text into bibliographic components. Other archives (e.g., Biomed Central), whose core document format is XML, will be able to provide fully marked-up references, so that only minimal processing will be required by linking services. The standardization of OpenURL, which is gathering growing support as an open and extensible system for reference linking using HTTP URLs, may result in a widely-used bibliographic-aware metadata format [3].

A citation-linking system is likely to be hierarchical because middle normalization layers will be needed to parse source archive reference lists into a common format that end-user services can link to bibliographic databases; furthermore references will also need to be provided in a format understood by the end-user.

9 Experiments

9.1 OAI-PMH-Proxy

Our first experiment is an OAI-PMH-specific proxy that takes a URL of the format:
`http://foo.org/OAIProxy/{repositoryid}?{oai verb}`
Its function is to filter XML encoding errors. This proxy relies on a preex-

isting mapping table between an OAI-PMH repository ID and a baseURL. When an OAI-PMH request is issued, the proxy forwards the request to the corresponding data provider. The XML response is parsed by a DOM parser; if any XML encoding errors exist, the proxy tries to delete bad records based on the detailed error message from the DOM parser. The proxy then returns the corrected XML response.

9.2 OAI Aggregation/Caching/Filtering

An OAI-PMH cache service has been explored in several experiments, including OAI Aggregator, Arc, and CiteBase. Both Arc and Citebase act as DPs disseminating Dublin Core metadata harvested from other DPs.

OAIA is specially designed to mirror OAI repositories. OAIA creates a duplicate of all available data from the source repositories, excluding the set hierarchy (with OAI 1.1, the set hierarchy can only be ascertained through exhaustive querying of each set).

OAIA is designed to facilitate OAI-PMH Gateway services, which rely on fast, reliable access to OAI repositories. For DP9, this consists of a few list identifiers requests, followed by large numbers of individual record requests (as a Web crawler requests the corresponding record's web page). Otherwise, this kind of behavior can lead to large numbers of (possibly unwanted) requests to source repositories exposed through DP9.

OAIA also acts as a gateway from legacy OAI-PMH implementations (1.x) to the most recent version of the OAI-PMH (2.0). As well as being able to harvest from any repository that has been OAI-PMH-compliant, OAIA converts the required Dublin Core metadata format to the most recent OAI-PMH version.

Given a new repository URL, OAIA issues an Identify request. The Identify response is stored so an SP can retrieve from OAIA the source repository's data policies, etc. A ListMetadataFormats request is issued to find out which metadata formats are supported. Each record is then requested for each metadata format (either using the batch command ListRecords, or GetRecord, depending on the repository's reliability). The metadata is stored as it was received from the source repository, ready for an SP to harvest. The record's datestamp is changed to the time the record was harvested by OAIA.

OAIA provides two views to harvesters of the records it has collected: the aggregated collection, or individual repositories by using a wrapped URL.

When an aggregated OAIA collection receives a ListMetadataFormats request, it lists all the metadata formats used by any of the harvested repositories (which may include variants of the same metadata format). When the same request is made to a wrapped repository, it lists only the metadata formats supported by that repository.

9.3 Gateway

DP9 is a gateway service that allows general search engines, (e.g. Google, Inktomi) to index OAI-compliant archives. DP9 does this by providing persistent URLs for records, and converting them to OAI-PMH queries against the appropriate repository when the URL is requested. This service allows search engines that do not support the OAI-PMH to index the ”Deep Web” contained within OAI-PMH-compliant repositories.

DP9 is a gateway service; it does not cache the records and only forwards requests to corresponding DPs. This process ensures DP9 records are always up to date; however, its quality of service is highly dependent on the availability of DPs. On the other hand, an aggressive crawler using DP9 can rapidly send requests without regard for the load it places on the DPs. The robot exclusion protocol [5] at the DP site will not be observed because the requests originate from DP9. The OAIA is used relieve the load on DPs.

Another gateway service is the reference extraction module in CiteBase. CiteBase extracts the bibliography from arXiv.org documents and exposes them by an additional OAI-PMH interface. These data are then harvested by ODU to build its own citation service.

CiteBase adds reference data by harvesting new records from a repository’s OAI-PMH interface, then separately downloading the full text for parsing. The parsed bibliography is added to the existing metadata, and is used by CiteBase or harvested by other services.

The CiteBase concept could be extended from the current supported repository (arXiv.org [7]) to a general service for any full-text scientific repository - assuming tools can be developed to parse the bibliography.

9.4 End-User Services

Both Archon and CiteBase implement a cross-archive search interface; Archon focuses more on harvesting heterogeneous collections and builds an interactive search interface based on harvested metadata, and CiteBase concentrates on automatic reference extraction. Both applications may harvest from the same repository (e.g. arXiv.org [7]) and implement different services for the same record. With the quick adoption of OAI-PMH, we believe this will become a common situation. We implemented two prototypes for cross-linking between Archon and CiteBase (Figure 6).

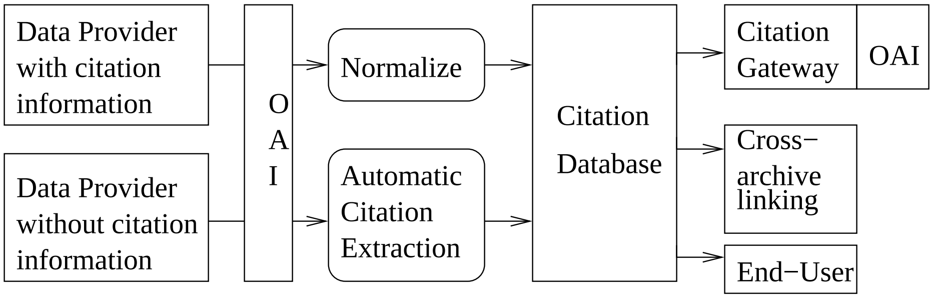
Data Provider with citation information
Data Provider without citation information
O A I
Normalize
Automatic Citation Extraction
Citation Database
Citation Gateway
OAI
Cross−archive linking
End−User

Figure 6: Cross Archive Citation Link

The first approach is to reexpose value-added metadata through an OAI-PMH interface. By this method Archon harvests citation data from CiteBase, APS, CERN and other sources, it then builds a cross archive linking service such as a citation in APS may lead to document in CiteBase and vice-versa. Another prototype is based on dynamic linking: both services link to a broker page, and the broker page dynamically checks whether a service exists for a specific record. If so, it adds a link to the corresponding SP. In order to know which records are available in advance, the broker issues an OAI GetRecord lookup to the target service (which has an OAI-PMH export). Based on the reply, the broker knows whether a record is harvested. We envision that a DP may also link to this broker page for additional services for its data.

10 Summary

The OAI-PMH is an important infrastructure for supporting distributed networked information services. This paper introduces the concepts of OAI-PMH proxies, OAI-PMH caches, OAI-PMH gateways as tools for the optimization of the functioning of the DP/SP model underlying the OAI-PMH. To demonstrate the usability of this framework, we have built several prototype services. These demonstration systems and source codes are available at the web sites of both ODU and the Southampton group.

11 Acknowledgements

Our thanks to Herbert Van de Sompel and Steve Hitchcock for providing valuable input and reviewing the manuscript.

References

[1] Archon: An OAI-compliant federated physics digital library. `http://archon.cs.odu.edu`.

[2] Cognitive science e-print archive. `http://cogprints.soton.ac.uk/`.

[3] Development of an OpenURL standard. `http://library.caltech.edu/openurl/`.

[4] eprints.org self-archiving FAQ. `http://www.eprints.org/self-faq/`.

[5] My.OAI. `http://www.myoai.com/`.

[6] Scirus - for scientific information. `http://www.scirus.com/`.

[7] arXiv.org. `http://arXiv.org/`.

[8] Michelle Baldonado, Chen-Chuan Chang, Luis Gravano, and Andreas Paepcke. The stanford digital library metadata architecture. *International Journal of Digital Libraries*, 1(2), 1997.

[9] Donna Bergmark. Automatic extraction of reference linking information from online documents. Technical report, Cornell University, 2000. `http://www.cs.cornell.edu/cdlrg/Reference\%20Linking/extraction.pdf`.

[10] C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz. The Harvest information discovery and access system. *Computer Networks and ISDN Systems*, 28(1–2):119–125, 1995. `http://citeseer.nj.nec.com/article/bowman95harvest.html`.

[11] Priscilla Caplan and William Y. Arms. Reference linking for journal articles. *D-Lib Magazine*, 5(7/8), 1999.

[12] Anawat Chankhunthod, Peter B. Danzig, Chuck Neerdaels, Michael F. Schwartz, and Kurt J. Worrell. A hierarchical internet object cache. In *USENIX Annual Technical Conference*, pages 153–164, 1996.

[13] Chaomei Chen and Les Carr. Trailblazing the literature of hypertext: Author co-citation analysis (1989-1998). In *Hypertext '99*, 1999. `http://www.ecs.soton.ac.uk/~lac/ht99.pdf`.

[14] CiteBase search. `http://citebase.eprints.org/`.

[15] James Davis and Carl Lagoze. NCSTRL: Design and deployment of a globally distributed digital library. *Journal of the American Society of Information Science*, 51(3), 2000.

[16] Eprints.org self-archiving software. `http://www.eprints.org/`.

[17] C. Lee Giles, Kurt Bollacker, and Steve Lawrence. CiteSeer: An automatic citation indexing system. In *Digital Libraries 98 - The Third ACM Conference on Digital Libraries*, pages 89–98, June 23–26 1998.

[18] Stevan Harnad, Les Carr, and Tim Brody. How and why to free all refereed research from access- and impact-barriers online, now. *High Energy Physics Libraries Webzine*, 4, June 2001. `http://library.cern.ch/HEPLW/4/papers/1/`.

[19] Steve Hitchcock, Les Carr, Zhuoan Jiao, Wendy Hall, Carl Lagoze, and Stevan Harnad. Developing services for open eprint archives: globalisation, integration and the impact of links. In *5th ACM Conference on Digital Libraries*, 2000. `http://opcit.eprints.org/dl00/dl00.html`.

[20] Alan Kent. OAI harvester crawling status. `http://www.mds.rmit.edu.au/~ajk/oai/interop/summary.htm`.

[21] Tom M. Kroeger, Darrell D. E. Long, and Jeffrey C. Mogul. Exploring the bounds of web latency reduction from caching and prefetching. In *USENIX Symposium on Internet Technologies and Systems*, 1997.

[22] Xiaoming Liu, Kurt Maly, Mohammad Zubair, and Michael L. Nelson. Arc - an OAI service provider for digital library federation. *D-Lib magazine*, 7(4), 2001. `http://www.dlib.org/dlib/april01/liu/04liu.html`.

[23] Xiaoming Liu, Kurt Maly, Mohammad Zubair, and Michael L. Nelson. DP9 - an OAI gateway service for Web crawlers. In *Proceedings of the Second ACM/IEEE Joint Conference on Digital Libraries*, 2002.

[24] Kurt Maly, Mohammad Zubair, and Xiaoming Liu. Kepler - an OAI data/service provider for the individual. *D-Lib magazine*, 7(4), 2001. `http://www.dlib.org/dlib/april01/maly/04maly.html`.

[25] Lee McLoughlin. Mirror software. `http://sunsite.doc.ic.ac.uk/packages/mirror/`.

[26] Michael L. Nelson and B. Danette Allen. Object persistence and availability in digital libraries. *D-Lib magazine*, 8(1), 2002.

[27] The open citation project. `http://opcit.eprints.org/`.

[28] Norman Paskin. DOI: Current status and outlook. *D-Lib Magazine*, 5(5), 1999.

[29] Hussein Suleman and Edward A. Fox. A framework for building open digital libraries. *D-Lib magazine*, 7(12), 2001. `http://www.dlib.org/dlib/december01/suleman/12suleman.html`.

[30] Sam X. Sun and Laurence Lannom. Handle system overview. `http://www.handle.net/overview-current.html`.

[31] Herbert Van de Sompel and Oren Beit-Arie. Generalizing the OpenURL framework beyond references to scholarly works- the Bison-Fut model. *D-Lib magazine*, 7(7/8), 2001. `http://www.dlib.org/dlib/july01/vandesompel/07vandesompel.html`.

[32] Herbert Van de Sompel and Oren Beit-Arie. Open linking in the scholarly information environment using the OpenURL framework. *D-Lib magazine*, 7(3), 2001. `http://www.dlib.org/dlib/march01/vandesompel/03vandesompel.html`.

[33] Herbert van de Sompel, Thomas Krichel, Michael Nelson, Patrick Hochstenbach, Victor M Lyapunov, Kurt Maly, Mohammed Zubair, Xiaoming Liu, and Heath O'Connell. The UPS Prototype: An experimental end-user service across e-print archives. *D-Lib Magazine*, 6(2), February 2000.

[34] Martin Vesely, Tibor Simko, and Thomas Baron. Whitepaper on OAI "result set filtering" issue. `http://documents.cern.ch/ettdh/doc/public/OAIRSF.html`.