A comprehensive tutorial on the OAI-PMH protocol and functions. You may skip the "Quick Quiz Questions" :-).

OAI for Beginners - the Open Archives Forum online tutorial

This tutorial is an introduction to the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). Originally developed as a means for metadata dissemination of preprint and e-print servers, OAI-PMH has become a widely known solution to connecting distributed electronic repositories of many kinds, and owes much of its acceptance to its simplicity and the comparatively very low costs of its implementation. Working through this tutorial you will: gain an overview of the history behind the OAI-PMH and an overview of its key features; achieve a deeper technical insight into how the protocol works; learn something about some of the main implementation issues; find some useful starting points and hints that will help you as an implementer. *The* **Overview**, **History and Development of OAI-PMH**, *and* **Glossary** *may be used on their own to gain information about OAI and OAI-PMH without going into technical implementation details.*

Acknowledgements
This tutorial is based on the tutorials given in conjunction with the second and third Open Archives Forum workshops. They were prepared and presented by **Uwe Müller** of Humboldt University Berlin and **Andy Powell** of UKOLN at University of Bath (Lisbon, December 2002), and Uwe Müller of Humboldt and **Pete Cliff** of UKOLN (Berlin, March 2003). **Herbert Van de Sompel**, **Carl Lagoze**, **Michael Nelson**, and **Simeon Warner** first developed many of the slides used in the presentation of those tutorials, and their contributions are gratefully acknowledged. Good graphics and parts of presentations have been shared generously throughout the Open Archives Initiative community, and the developers of this tutorial would like to thank anyone whose original work has turned up here without our knowledge of the original source.

Open Archives Forum (and hence the original development of this tutorial) was funded for two years from October 2001 through September 2003 as an accompanying measure within the Information Societies Technology (IST) Programme, a theme of the European Union's Fifth Framework Programme managed by the Information Society Directorate-General of the European Commission.

Contents

Contents | **1** | **2** | **3** | **4** | **5** | **6** | Previous | **Next**   **Back to OA-Forum tutorial entry**

Last modified: 14 Oct 2003 16:36
Authored in **CALnet**

1. OAI for Beginners: Overview

Contents of this part of *OAI for Beginners, the Open Archives Forum online tutorial*

How to use this tutorial

Top

This tutorial is intended for those who are interested in more technical aspects of the OAI-PMH, although the **Overview** and the **History and Development of OAI-PMH**, together with the **Glossary**, are suitable for those who simply require some general background information. Each part builds on the material in the earlier parts, so a good approach is to work through the parts in order, referring to the glossary as required. In addition to the Glossary, you will find key terms defined within each part of the tutorial. Sets of quick quiz questions for the introductory sections help you to check whether you've picked up key points along the way.

Overview (this part) introduces the basic concepts underlying the OAI and the OAI-PMH. Use this part to gain an understanding of what the OAI-PMH is, and what it does and does not provide. **History and Development of OAI-PMH** covers the emergence of the Open Archives Initiative, showing how it grew from roots in several earlier initiatives, and discussing the nature of the problems for which it aims to provide solutions. This part also surveys the development of the protocol (including the evolving nature, aims and technical components) from the Santa Fe Convention, through OAI-PMH v.1.0/1.1, to OAI-PMH v.2.0.

The rest of the tutorial contains more technical material. **The Main Technical Ideas of OAI-PMH** introduces and explains in some detail the key technical elements of the protocol. **Implementing OAI-PMH** outlines implementation issues for Data Providers and Service Providers; it explains how to implement OAI-PMH as a Data Provider and as a Service Provider, including both the necessary steps for a local implementation and several examples of freely available and adaptable tools for implementations. **XML Schemas and Record Formats** provides an overview of the implementation of a Data Provider metadata set, including coverage of XML schema and how to support multiple record formats.

Basic OAI concepts and features

Top

--- **Open Archives Initiative (OAI)** ---

The essence of the open archives approach is to enable access to Web-accessible material through interoperable repositories for metadata sharing, publishing and archiving. It arose out of the e-print community, where a growing need for a low-barrier interoperability solution to access across fairly heterogeneous repositories lead to the establishment of the Open Archives Initiative (OAI). The OAI develops and promotes a low-barrier interoperability framework and associated standards, originally to enhance access to e-print archives, but now taking into account access to other digital materials. As it says in the OAI mission statement "The Open Archives Initiative develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content."

Many communities are beginning to or potentially could benefit from the open archives approach. The Internet and the growing mass of material in digital format have broadened the potential clientele of many repositories of information. Material can be accessed more widely and also exploited for purposes different from those that originally motivated the creation of the repositories. Moreover, the possibility of accessing multiple repositories enables the construction of new kinds of services that can better serve the needs of the users. An additional incentive is the potential for cost-saving inherent in new models of the scholarly communication process that could be supported with this approach.

As an organisation, the OAI has included an Executive for management, and Steering and Technical Committees for policy direction and evaluation of protocol developments. The Digital Library Federation (DLF), the Coalition for Networked Information (CNI), and the National Science Foundation (NSF) have funded the OAI. While the Executive and the funders are USA-based, the success of the OAI is firmly grounded in the participation of a community of people from around the world, particularly Europe as well as North America. Now that there is a well-developed and stable second version of the protocol, the need to keep control in the hands of a very small number of people who can take independent and speedy decisions may be less important when weighed against the perception of stability and authority conferred by control through a standards body such as ISO, and this possibility has been discussed within the OAI.

--- **OAI Protocol for Metadata Harvesting (OAI-PMH)** ---

The OAI-Protocol for Metadata Harvesting (OAI-PMH) defines a mechanism for harvesting records containing metadata from repositories. The OAI-PMH gives a simple technical option for data providers to make their metadata available to services, based on the open standards HTTP (Hypertext Transport Protocol) and XML (Extensible Markup Language). The metadata that is harvested may be in any format that is agreed by a community (or by any discrete set of data and service providers), although unqualified Dublin Core is specified to provide a basic level of interoperability. Thus, metadata from many sources can be gathered together in one database, and services can be provided based on this centrally harvested, or "aggregated" data. The link between this metadata and the related content is not defined by the OAI protocol. It is important to realise that OAI-PMH does not provide a search across this data, it simply makes it possible to bring the data together in one place. In order to provide services, the harvesting approach must be combined with other mechanisms.

Much promise is seen for the use of the protocol within an open archives approach. Support for a new pattern for scholarly communication is the most publicised potential benefit. Perhaps most readily achievable are the goals of surfacing 'hidden resources' and low cost interoperability. Although the OAI-PMH is technically very simple, building coherent services that meet user requirements remains complex. The OAI-PMH protocol could become part of the infrastructure of the Web, as taken-for-granted as the HTTP protocol now is, if a combination of its relative simplicity and proven success by early implementers in a service context leads to widespread uptake by research organisations, publishers, and "memory organisations".

Seven key definitions

Top

Open Archive Initiative (OAI)
OAI is an initiative to develop and promote interoperability standards that aim to facilitate the efficient dissemination of content.

Archive
The term "archive" in the name Open Archives Initiative reflects the origins of the OAI in the e-prints community where the term archive is generally accepted as a synonym for repository of scholarly papers. Members of the archiving profession have justifiably noted the strict definition of an ?archive? within their domain; with connotations of preservation of long-term value, statutory authorization and institutional policy. The OAI uses the term ?archive? in a broader sense: as a repository for stored information. Language and terms are never unambiguous and uncontroversial and the OAI respectfully requests the indulgence of the professional archiving community with this broader use of ?archive?.
(OAI definition quoted from FAQ on OAI Web site)

OAI Protocol for Metadata Harvesting (OAI-PMH)
OAI-PMH is a lightweight harvesting protocol for sharing metadata between services.

Protocol
A protocol is a set of rules defining communication between systems. FTP (File Transfer Protocol) and HTTP (Hypertext Transport Protocol) are examples of other protocols used for communication between systems across the Internet.

Harvesting
In the OAI context, harvesting refers specifically to the gathering together of metadata from a number of distributed repositories into a combined data store.

Data Provider
A Data Provider maintains one or more repositories (web servers) that support the OAI-PMH as a means of exposing metadata.
(OAI definition quoted from FAQ on OAI Web site)

Service Provider
A Service Provider issues OAI-PMH requests to data providers and uses the metadata as a basis for building value-added services.
(OAI definition quoted from FAQ on OAI Web site)
A Service Provider in this manner is "harvesting" the metadata exposed by Data Providers

Sources of further information

Top

The rest of this tutorial.

Open Archives Initiative (OAI official Web site)
http://www.openarchives.org/

Open Archives Forum (OA-Forum Web site)
http://www.oaforum.org/

Quick Quiz Questions

Top

Answer and 'Mark' each question separately. Feedback is provided for each marked answer. Once you have marked a question, you can get a further 'Explanation' of the answers. When you have finished, check your total marks for the questions you tried. The marks are provided only to you, they are not stored when you leave this page.

Q1. What is the OAI?

(Select one answer)

(a) ○ The OAI is an initiative to provide open access to the output of scientific and other scholarly research.
(b) ○ The OAI is an initiative supporting archives for the preservation of and access to digital resources.

(c) ○ The OAI is an initiative to develop and promote interoperability standards that aim to facilitate the efficient dissemination of content.

Mark Explain

Q2. What is the OAI-PMH?

(Select one answer)

(a) ○ The OAI-PMH is a protocol for sharing metadata.
(b) ○ The OAI-PMH is a low-barrier protocol for searching across repositories and retrieving resources from them.

Mark Explain

Q3. What is the primary source of technical information about the OAI and OAI-PMH?

(Select one answer)

(a) ○ The Web site of the Open Archives Forum (http://www.oaforum.org/)
(b) ○ The Web site of the BOAI (http://www.soros.org/openaccess/)
(c) ○ The Web site of the Open Archives Initiative (http://www.openarchives.org/)

Mark Explain

Click to view your total score for all the above questions that you have attempted.

Total Score

Contents | 1 | 2 | 3 | 4 | 5 | 6 | Previous | Next Back to OA-Forum tutorial entry

2. History and development of OAI-PMH

Contents of this part of *OAI for Beginners, the Open Archives Forum online tutorial*

Roots of the OAI-PMH

Top

The roots of OAI lie in the development of e-print repositories (so-called archives). E-print repositories were established in order to communicate the results of ongoing scholarly research prior to peer review and journal publication. The earliest of these was xxx (later arXiv), which began with high energy physics in 1991 and expanded to cover the field of physics plus related fields of mathematics, nonlinear sciences and computer science. CogPrints followed for psychology, linguistics and neuroscience. The Networked Computer Science Technical Reference Library (NCSTRL) provided access to computer sciences technical reports deposited either in xxx or in departmental repositories of cooperating research bodies. Similarly, RePEc provided authors in the field of economics with the option to submit working papers to their departmental archive or, if there was none, to the EconWPA archive at Washington University. In addition, the Networked Digital Library of Theses and Dissertations (NDLTD) built a digital library of electronic theses and dissertations (ETDs) authored by students of member institutions.

The mechanism for filling these repositories, in all cases, was by author deposit. (Within the OAI, and for the purposes of this tutorial, an "e-print" is defined as an author self-archived document.) Web interfaces allowed people to interact with these repositories and some finding aids were provided. Different interfaces were designed for different repositories, so end users were forced to learn diverse interfaces in order to access the various repositories and finding aids. The "Guildford protocol" supported interoperability between the RePEc archives, while NCSTRL repositories implemented the Dienst protocol. These protocols made possible a variety of end-user services, including those supporting end-user search and browse across repositories in each grouping. NDLTD created a workflow for submitting material, and developed an XML DTD (Document Type Description) for electronic ETDs, as well as supporting a ETD digital library. However, little or no autonomous metadata sharing was supported across this diverse environment, and still further separate initiatives in the area of new means of scholarly communication were taking shape. Certain key players in these developments came to see interoperability as an increasingly important issue to be addressed by the e-print community.

Santa Fe meeting

Top

> ??the joint impact of these and future initiatives can be substantially higher when interoperability between them [e-print archives] can be established??
> (Ginsparg, Luce, Van de Sompel, UPS Call, July 1999)

Two key interoperability problems were identified as impairing the impact of e-print archives: end users were faced with multiple search interfaces making resource discovery harder, and there was no machine-based way of sharing the metadata. Solutions that were being explored included cross-searching of archives (on one hand) and harvesting archive metadata (on the other hand) in order to provide centralised search services. In July 1999 Paul Ginsparg, Rick Luce, and Herbert Van de Sompel of the Los Alamos National Laboratory issued a call to a restricted group of technical experts to attend a meeting Santa Fe, New Mexico in October of the same year. Ginsparg was involved with arXiv, and Van de Sompel was also still associated with the University of Ghent at that time. They proposed the creation of a universal service for author self-archived scholarly literature (the Universal Preprint Service, or UPS). The UPS would be "the fundamental and free layer of scholarly information, above which both free and commercial services could flourish". The first steps toward establishing it would be the identification or creation of interoperable technologies and frameworks for the dissemination of e-prints. This was announced to a wider audience under the headline "The Open Archives initiative aimed at the further promotion of author self-archived solutions".

The aim of the meeting in Santa Fe was to discuss interoperability issues, agree to begin work on a promotional prototype digital library service based on the main existing e-print repositories, and establish a forum for continuing work on interoperability of self-archiving solutions. In preparation for the meeting, some foundational work was undertaken. Van de Sompel initiated a project that simulated some aspects of interoperable distributed e-print archives. Thomas Krichel (University of Surrey & RePEc) experimented with converting data from existing e-print initiatives into the ReDIF metadata format used in RePEc. Michael Nelson (NASA Langley) took this data and used it to create various archives architected along the lines of his Smart Object Dumb Archives concepts. Data used came from sources including CogPrints, NASA, NCSTRL, RePEc and xxx. The aim of this work was not to make statements about the architectural directions that the UPS should take, but rather to facilitate discussions about this at the October meeting.

Challenges and proposed solutions

Top

Cross search or harvest?

Choice of a general direction to take in developing the architectural framework for a UPS was a key issue at this early stage. Two possible approaches were cross-searching multiple archives based on a protocol such as Z39.50, or else harvesting metadata into one or more "central" services in a bulk move of data that would bring it closer to the user interface.

Digital library experience suggested that cross searching does not scale well, at least partly because the search service degrades to the level of the slowest and least reliable server in the cross search set. For example, NCSTRL found that distributed searching of a small number of nodes was viable, but that performance was very bad over 100 nodes. In the UK, the Resource Discovery Network (RDN) was finding that even with only five subject gateways in its cross search there were problems of poor performance and in the provision of a browse interface, and developers were looking for a feasible centralised database solution. The more servers are cross-searched, the higher are the chances of encountering one or more slow or unreliable servers.

There is also the problem of knowing which target servers to use in any particular cross search. Collection descriptions – where they are available at all – may be inconsistent across repositories, were not designed for machine-to-machine communication and require time-consuming examination by end-users. Differences in query language syntax and search attribute variation (between servers and over time) introduce barriers of complexity, either for the end user or the cross-search software, or both. Ranked merging of results from distributed servers presents further technical and user-interface problems, and different size and types of targets can skew results. A browse interface is very difficult to build when the metadata to be browsed is distributed across a number of repositories. It was suggested that a solution would be to get all the metadata records together in one place.

The UPS prototype brought to the Santa Fe meeting demonstrated a cross-archive digital library providing services based on a collection of metadata harvested from multiple archives. Its architecture drew on NCSTRL and a modified version of the Dienst protocol. In this way, the number of nodes being searched could be reduced to one, giving significant performance benefits. A service could be provided using one query language, set of search attributes and ranking algorithm. In addition, an awareness of the data makes browse structures easier to build.

Data and Service Providers

The UPS architecture identified two logical roles: "Data Providers" and "Service Providers". Data Providers handle the deposit and publishing of resources in a repository and "expose" for harvesting the metadata about resources in the repository. They are the creators and keepers of the metadata and repositories of resources. Service Providers harvest metadata from Data Providers. They use the harvested metadata for the purpose of providing one or more services across all the data. The types of services that may be offered include a search interface, peer-review system, etc. Note that one 'provider' organisation can play both roles, offering both data for harvesting and end-user services. The key architectural shift was the move away from only supporting human end-user interfaces for each repository, to supporting both human end-user interfaces *and* machine interfaces for harvesting.

The dawn of a protocol

Top

The name UPS (Universal Preprint Service) was quickly changed, partly in order to avoid potential difficulties related to the fact that UPS is an established brand name for a commercial parcel delivery service, and partly because it was recognised that not all e-prints were preprints. The framework within which this universal service would be developed was now designated the Open Archives initiative – shortened to OAi, later OAI – a phrase that had gained currency in early discussions.

It was clear from discussions and experiments that in order to facilitate metadata harvesting there must be agreement on:

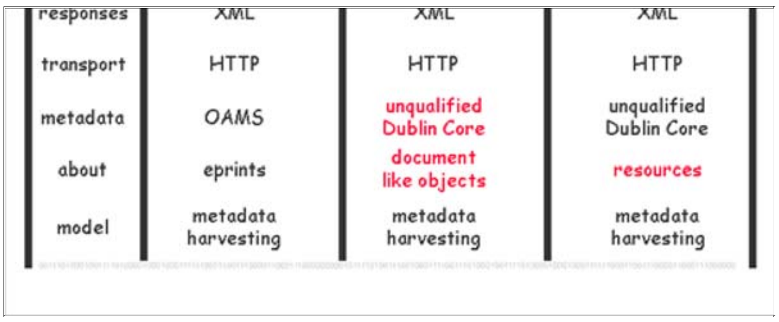
- a transport protocol – HTTP or FTP for example
- a metadata format – Dublin Core or MARC for example
- a basis for metadata quality assurance – mandatory element set, naming and subject conventions, etc.
- intellectual property and usage rights – *who* can do *what* with *what*?

An initial agreement in key areas made it possible to develop a protocol for metadata harvesting, named the Santa Fe Convention in honour of the meeting where this agreement was reached.

OAI-PMH version history

Top

The Santa Fe Convention was the first incarnation of the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). It drew upon the UPS Prototype, the RePEc/SODA Service/Data provider model, the Dienst Protocol, and the work of the Santa Fe group. The focus of the Santa Fe Convention was to ?optimise the discovery of e-prints?.



responses	XML	XML	XML
transport	HTTP	HTTP	HTTP
metadata	OAMS	unqualified Dublin Core	unqualified Dublin Core
about	eprints	document like objects	resources
model	metadata harvesting	metadata harvesting	metadata harvesting

The OAI-PMH 1.0 introduced the unqualified Dublin Core element set as a baseline for metadata interoperability. It drew upon the Santa Fe Convention, Digital Library Federation meetings, work at Cornell University, and feedback from alpha-testers. The focus expanded to facilitating the discovery of ?document-like objects". It was a low barrier interoperability specification, based on a metadata harvesting model. It was HTTP based, using HTTP GET / POST requests and XML responses. Note that it is not a search protocol, rather, it is based on the metadata harvesting model. OAI-PMH 1.1 was a revision of the 1.0 specification taking account of changes to the emerging XML Schema specification. Both v.1.0 and 1.1 were experimental in nature.

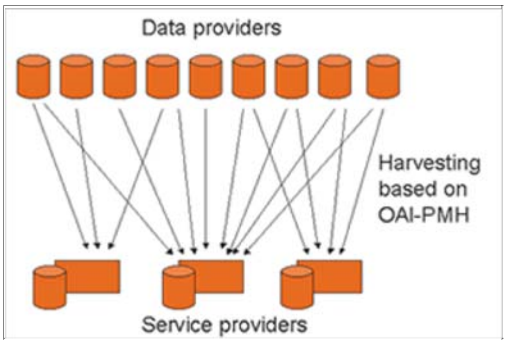
The OAI-PMH 2.0 is a major revision of the protocol, and is not compatible with v.1.x. It drew upon OAI-PMH 1.x, feedback from OAI Implementers List, OAI Tech deliberation, and feedback from alpha-testers. Once again the focus of the protocol expanded; now it was said to be about ?the recurrent exchange of metadata about ***resources*** between systems?. It is still a low barrier interoperability specification based around a metadata harvesting model. No longer experimental, v.2.0 is a stable protocol, and OAI has committed to making subsequent revisions of the protocol backwards compatible.

Flexible deployment: various says OAI-PMH may be deployed

Top

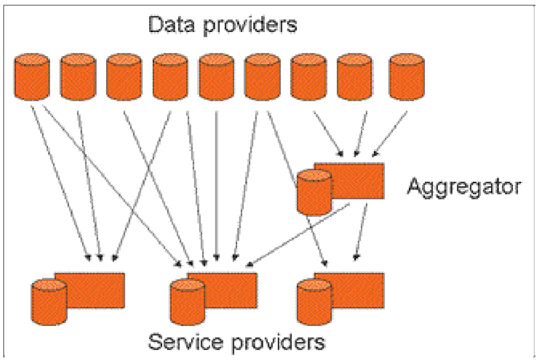
OAI-PMH enables flexible deployment. Because it is a simple protocol based on HTTP and XML, it allows for rapid deployment. A number of toolkits are available, as will be discussed later in this tutorial. Systems can be deployed in a variety of configurations, as illustrated in the following diagrams. Metadata and full-text resources are typically made freely available, but this is not a requirement. OAI-PMH can also be used between closed groups; for metadata-sharing only; and in commercial applications.

Multiple Service Providers



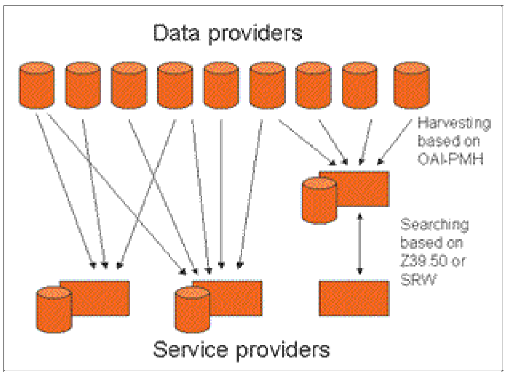
Multiple Service Providers can harvest from multiple Data Providers.

Aggregators



Aggregators can sit between Data Providers and Service Providers.

Harvesting combined with searching



The harvesting approach can be complemented with searching based, for example, on Z39.50 or SRW.

Summary

Top

Early movers were developing separate solutions, but the need for interoperability was recognised. In response, the Santa Fe Meeting led to substantial support for the OAI, which promotes interoperability via developing OAI-PMH as an open standard, and disseminating information about OAI-PMH. OAI-PMH is a low-cost mechanism for harvesting metadata records from one system to another – from Data Providers to Service Providers. Multiple Service Providers can harvest from multiple Data Providers ensuring a wider spread of metadata. OAI-PMH is not a search protocol, but its use can underpin search-based services; it is a base layer on which to build other services.

Development over the last two to three years has seen a move from the specific to the generic – from discovery of e-prints to sharing descriptions of any resources. Although unqualified Dublin Core is specified for baseline interoperability, OAI-PMH can be extended to any metadata format that can be encoded in XML. Based on HTTP for requests (and access-control, compression, error codes, etc.) and on XML for responses, it is Web-friendly, and therefore firewall friendly. It allows Service Providers to say 'give me some or all of your records', where 'some' is based on date-stamps, sets, or metadata formats. Simple, and built on existing technology, OAI-PMH is easy to deploy, with many toolkits that can hide the protocol from developers.

Seven key definitions

Top

E-print
An e-print is an author self-archived document. In the sense that the term is ordinarily used, the content of an e-print is the result of scientific or other scholarly research.

Document-like object
A document-like object is a digital data unit that is comparable to a paper document. The term designates a relatively simple stable resource, and would not cover, for example multimedia artifacts or interactive services.

Resource
A resource is anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., today's "weather report for Los Angeles"), and a collection of other resources. Not all resources are network "retrievable"; e.g., human beings, corporations, and bound books in a library can also be considered resources. (Definition from **Guidelines for implementing Dublin Core in XML** by Andy Powell and Pete Johnston)

XML
XML is the acronym for Extensible Markup Language. XML is a language for creating other languages. It defines a means of describing data. XML can be validated against a DTD or schema setting out the elements of the language created. XML mappings exist for a number of metadata record formats.

DTD
DTD is the acronym for Document Type Definition. A DTD is a formal specification of the structure of a document.

Dublin Core
Dublin Core (DC) is a metadata format defined on the basis of international consensus. The **Dublin Core** Metadata Element Set defines fifteen elements for simple resource description and discovery, all of which are recommended, and none of which are mandatory. DC has been extended with further optional elements, element qualifiers and vocabulary terms.
(Definition draws on UKOLN's **metadata glossary** and **Metadata in a nutshell** by Michael Day)

Interoperability
Interoperability is the ability of systems, services and organisations to work together seamlessly toward common or diverse goals. In the technical arena it is supported by open standards for communication between systems and for description of resources and collections, among others. Interoperability is considered here primarily in the context of resource discovery and access.

Sources of further information

Top

Articles

Lynch, C.A. Metadata Harvesting and the Open Archives Initiative. *ARL Bimonthly Report 217*, August 2001

http://www.arl.org/newsltr/217/mhp.html

Van de Sompel, Herbert, Krichel, T., Nelson, M. L. and others. The UPS Prototype: An Experimental End-User Service across E-Print Archives. *D-Lib Magazine*, vol.6, no. 2. February 2000.
http://www.dlib.org/dlib/february00/vandesompel-ups/02vandesompel-ups.html

Van de Sompel, H., Lagoze, C. The Santa Fe Convention of the Open Archives Initiative. *D-Lib Magazine*, vol.6, no.2. February 2000.
http://www.dlib.org/dlib/february00/vandesompel-oai/02vandesompel-oai.html

Web site

OAI Web site: **http://www.openarchives.org/**

Contents | **1** | **2** | **3** | **4** | **5** | **6** | **Previous** | **Next** **Back to OA-Forum tutorial entry**

Contents | 1 | 2 | 3 | 4 | 5 | 6 | Previous | Next Back to OA-Forum tutorial entry

3. Main Technical Ideas of OAI-PMH

Contents of this part of *OAI for Beginners, the Open Archives Forum online tutorial*

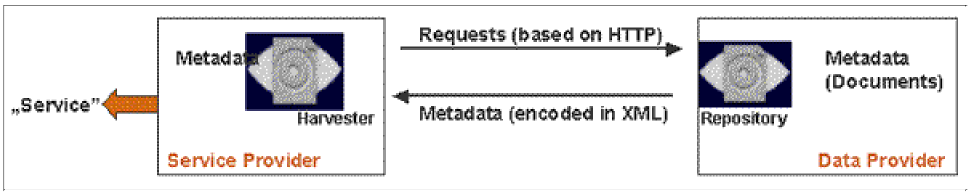
The Open Archives Initiative (OAI)

Top

Recap of the main ideas of OAI

- world-wide consolidation of scholarly archives
- free access to the archives (at least: metadata)
- consistent interfaces for archives and service provider
- low barrier protocol / effortless implementation (e.g., because based on HTTP, XML, DC)

Basic functioning of OAI-PMH



OAI: general assumptions

Top

There are two groups of 'participants': Data Providers and Service Providers.

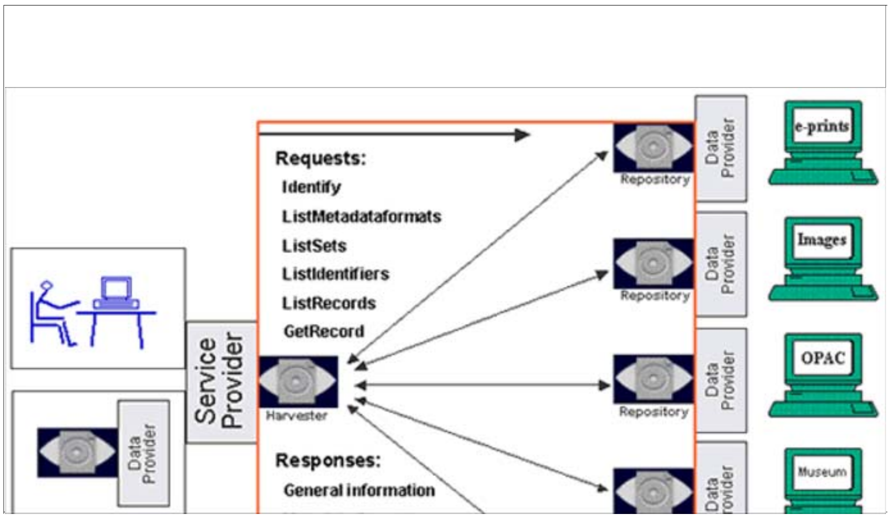
Data Providers (open archives, repositories) provide free access to metadata, and may, but do not necessarily, offer free access to full texts or other resources. OAI-PMH provides an easy to implement, low barrier solution for Data Providers.

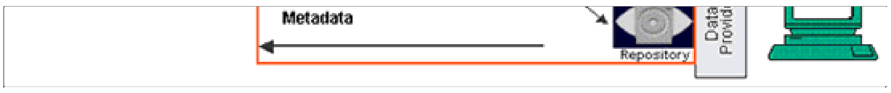
Service Providers use the OAI interfaces of the Data Providers to harvest and store metadata. Note that this means that there are no live search requests to the Data Providers; rather, services are based on the harvested data via OAI-PMH. Service Providers may select certain subsets from Data Providers (e.g., by set hierarchy or date stamp). Service Providers offer (value-added) services on the basis of the metadata harvested, and they may enrich the harvested metadata in order to do so.

OAI-PMH: overview and structure model

Top

The OAI-PMH protocol is based on HTTP. Request arguments are issued as **GET** or **POST** parameters. OAI-PMH supports six request types (known as "verbs"), e.g.,





`http://archive.org?verb=ListRecords&from=2002-11-01`.

Responses are encoded in XML syntax. OAI-PMH supports any metadata format encoded in XML. Dublin Core is the minimal format specified for basic interoperability.

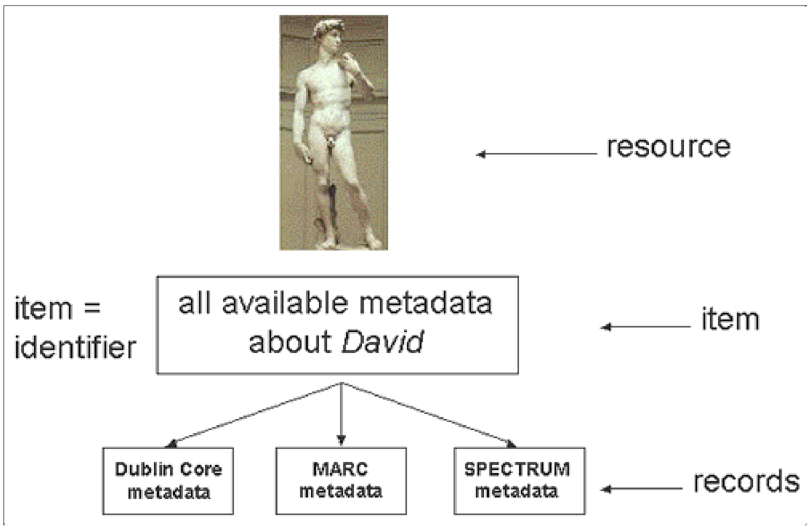
Error messages are HTTP-based.

Data Providers may define a logical set hierarchy to support levels of granularity for harvesting by Service Providers. Date stamps flag the last change of the metadata set, and thus provide further support for granularity of harvesting.

OAI-PMH supports flow control.

Seven key definitions

Top



Harvester:
client application issuing OAI-PMH requests

Repository:
network accessible server, able to process OAI-PMH requests correctly

Resource:
object the metadata is "about", nature of resources is not defined in the OAI-PMH – resources may be digital or non-digital

Item:
component of an repository from which metadata about a resource can be disseminated; has an unique identifier

Record:
metadata in a specific metadata format

Identifier:
unique key for an item in a repository

Set:
optional construct for grouping items in a repository

Protocol details

Top

-- Records --

A record is the metadata of a resource in a specific format. A record has three parts: a header and metadata, both of which are mandatory, and an optional about statement. Each of these is made up of various components as set out below.

header (mandatory)
  `identifier` (mandatory: 1 only)
  `datestamp` (mandatory: 1 only)
  `setSpec` elements (optional: 0, 1 or more)
  status attribute for deleted item

metadata (mandatory)
  XML encoded metadata with root tag, namespace
  repositories must support Dublin Core, may support other formats

about (optional)
  rights statements
  provenance statements

-- Datestamps --

A datestamp is the date of last modification of a metadata record. Datestamp is a mandatory characteristic of every item. It has two possible levels of granularity:
YYYY-MM-DD or YYYY-MM-DDThh:mm:ssZ.

The function of the datestamp is to provide information on metadata that enables selective harvesting using **`from`** and **`until`** arguments. Its applications are in incremental update mechanisms. It gives either the date of creation, last modification, or deletion. Deletion is covered with three support levels: no, persistent, transient.

-- Metadata schema --

OAI-PMH supports dissemination of multiple metadata formats from a repository. The properties of metadata formats are:

– id string to specify the format (**`metadataPrefix`**)
– metadata schema URL (XML schema to test validity)
– XML namespace URI (global identifier for metadata format)

Repositories must be able to disseminate unqualified Dublin Core. Further arbitrary metadata formats can be defined and transported via the OAI-PMH. Any returned metadata must comply with an XML namespace specification. The Dublin Core Metadata Element Set contains 15 elements. All elements are optional, and all elements may be repeated.

The Dublin Core Metadata Element Set:

Title	Contributor	Source
Creator	Date	Language
Subject	Type	Relation
Description	Format	Coverage
Publisher	Identifier	Rights

-- Sets --

Sets enable a logical partitioning of repositories. They are optional archives do not have to define Sets. There are no recommendations for the implementation of Sets. Sets are not necessarily exhaustive of the content of a repository. They are not necessarily strictly hierarchical. It is important and necessary to have negotiated agreements within communities defining useful sets for the communities.

- function: selective harvesting (**set** parameter)
- applications: subject gateways, dissertation search engine, and others
- examples
  - publication types (thesis, article, ?)
  - document types (text, audio, image, ?)
  - content sets, according to DNB (medicine, biology, ?)

-- Request format --

Requests must be submitted using the **`GET`** or **`POST`** methods of HTTP, and repositories must support both methods. At least one key=value pair: verb=RequestType (where RequestType is some type of request such as **`ListRecords`**) must be provided. Additional key=value pairs depend on the request type.

example for **`GET`** request: **`http://archive.org/oai?verb=ListRecords&metadataPrefix=oai_dc`**

The encoding of special characters must be supported; for example, "**:**" (host port separator) becomes "**%3A**"

-- Response --

Responses are formatted as HTTP responses. The content type must be text/xml. HTTP-based status codes, as distinguished from OAI-PMH errors, such as 302 (redirect) and 503 (service not available) may be returned. Compression codes are optional in OAI-PMH, only identity encoding is mandatory. The response format must be well-formed XML with markup as follows:

1. XML declaration
   (**`<?xml version="1.0" encoding="UTF-8" ?>`**)
2. root element named **`OAI-PMH`** with three attributes

(**xmlns**, **xmlns:xsi**, **xsi:schemaLocation**)
3. three child elements
   1. **responseDate** (UTC datetime)
   2. **request** (the request that generated this response)
   3. a) **error** (in case of an error or exception condition)
      b) element with the name of the OAI-PMH request

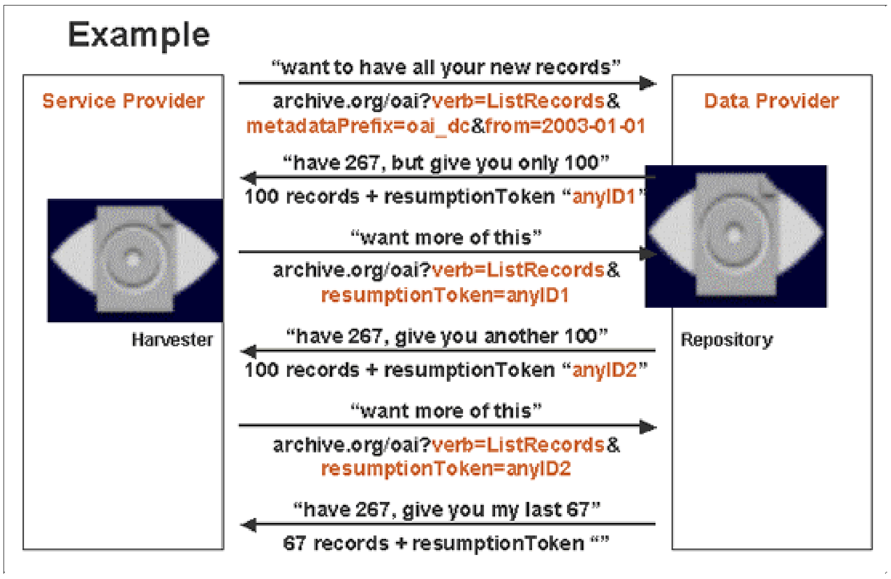
-- Flow control --

Four of the request types return a list of entries. Three of them may reply with 'large' lists.

OAI-PMH supports partitioning. Those managing a repository make the decisions on partitioning: whether to partition and how.

The response to a request includes:
  incomplete list
  resumption token
    + expiration date,
      size of complete list,
      cursor (optional)

For a new request with same request type:
  resumption token as parameter
  all other parameters omitted!


Example
Service Provider
Data Provider
Harvester
Repository
"want to have all your new records"
archive.org/oai?verb=ListRecords&
metadataPrefix=oai_dc&from=2003-01-01
"have 267, but give you only 100"
100 records + resumptionToken "anyID1"
"want more of this"
archive.org/oai?verb=ListRecords&
resumptionToken=anyID1
"have 267, give you another 100"
100 records + resumptionToken "anyID2"
"want more of this"
archive.org/oai?verb=ListRecords&
resumptionToken=anyID2
"have 267, give you my last 67"
67 records + resumptionToken ""

The response includes the next (which may be the last) section of the list and a resumption token. That resumption token is empty if the last section of the list is enclosed.

-- Errors and exceptions --

Repositories must indicate OAI-PMH errors by the inclusion of one or more **error** elements. The defined error identifiers are:

badArgument
badResumptionToken
badVerb
cannotDisseminateFormat
idDoesNotExist
noRecordsMatch
noMetaDataFormats
noSetHierarch

Request types

Top

There are six different request types:

Identify
ListMetadataFormats
ListSets
ListIdentifiers
ListRecords
GetRecord

A harvester is not required to use all types. However, a repository must implement all types. There are required and optional arguments, depending on request types. Each request type will now be described.

-- Identify --

function
  description of an archive

example
  archive.org/oai-script?**verb=Identify**

parameters
none

errors / exceptions
badArgument (e.g. archive.org/oai-script?verb=Identify**&set=biology**)

response format

Element	*Example*	*Ordinality* ‡
repositoryName	My Archive	1
baseURL	http://archive.org/oai	1
protocolVersion	2.0	1
earliestDatestamp	1999-01-01	1
deleteRecords	no, transient, persistent	1
granularity	YYY-MM-DD, YYYY-MM-DDThh:mm:ssZ	1
adminEmail	oai-admin@archive.org	+
compression	deflate, compress	*
description	oai-identifier, eprints, friends, …	*

‡ Ordinality: 1 = mandatory, 1 only; + = mandatory, 1 only; * = optional, 0 or more

-- ListMetadataFormats --

function
retrieve available metadata formats from archive

example
archive.org/oai-script?**verb=ListMetadataFormats**&
identifier=oai:HUBerlin.de:3000218

parameters
identifier (optional)

errors / exceptions
badArgument
idDoesNotExist
e.g. archive.org/oai-script?verb=ListMetadataFormats
&identifier=really-wrong-identifier
noMetadataFormats

-- ListSets --

function
retrieve set structure of a repository

example
archive.org/oai-script?**verb=ListSets**

parameters
resumptionToken (exclusive)

errors / exceptions
badArgument
badResumptionToken
e.g. archive.org/oai-script?verb=ListSets
&resumptionToken=any-wrong-token
noSetHierarchy

-- ListIdentifiers --

function
abbreviated form of **ListRecords**, retrieving only headers

example
archive.org/oai-script?**verb=ListIdentifiers**&

        metadataPrefix=oai_dc&**from=2002-12-01**

parameters
  from (optional)
  until (optional)
  metadataPrefix (required)
  set (optional)
  resumptionToken (exclusive)

errors / exceptions
  badArgument (e.g. ?**&from=2002-12-01-13:45:00**)
  badResumptionToken
  cannotDisseminateFormat
  noRecordsMatch
  noSetHierarchy

-- ListRecords --

function
  harvest records from a repository

example
  archive.org/oai-script?**verb=ListRecords**&
        metadataPrefix=oai_dc&**set=biology**

parameters
  from (optional)
  until (optional)
  metadataPrefix (required)
  set (optional)
  resumptionToken (exclusive)

errors / exceptions
  badArgument
  badResumptionToken
  cannotDisseminateFormat
  noRecordsMatch
  noSetHierarchy

-- GetRecord --

function
  retrieve individual metadata record from a repository

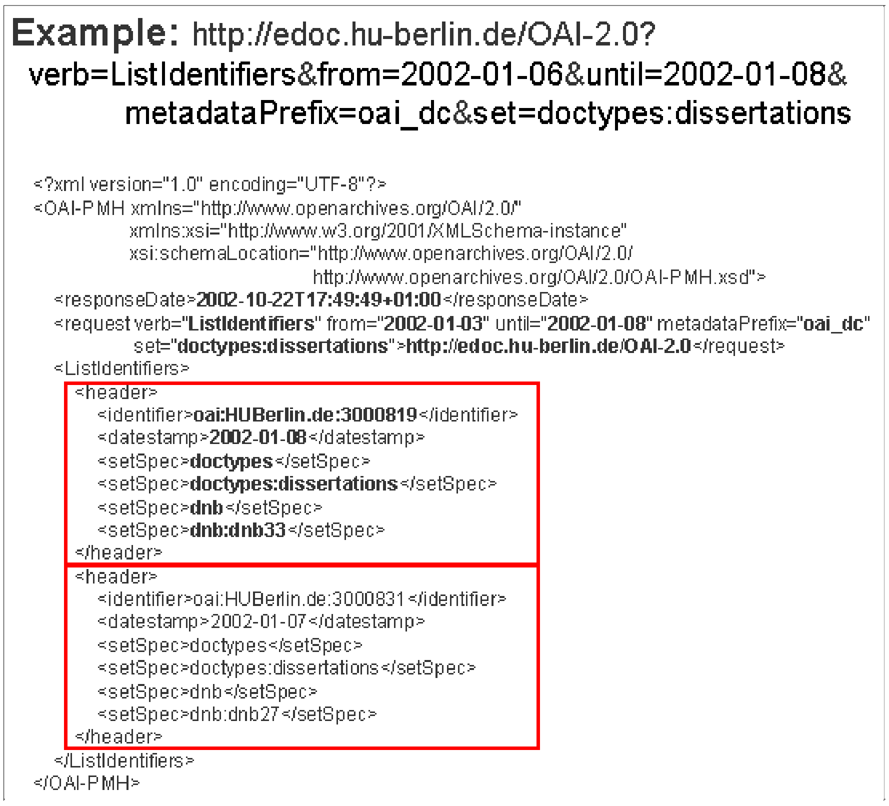
example
  archive.org/oai-script?**verb=GetRecord**&
        identifier=oai:HUBerlin.de:3000218&
        metadataPrefix=oai_dc

parameters
  identifier (required)
  metadataPrefix (required)

errors / exceptions
  badArgument
  cannotDisseminateFormat
  idDoesNotExist

Example 1: response to ListIdentifiers request Top

This example shows the response to a ListIdentifiers request that specifies a date range, a metadata format, a set, and a Data Provider.

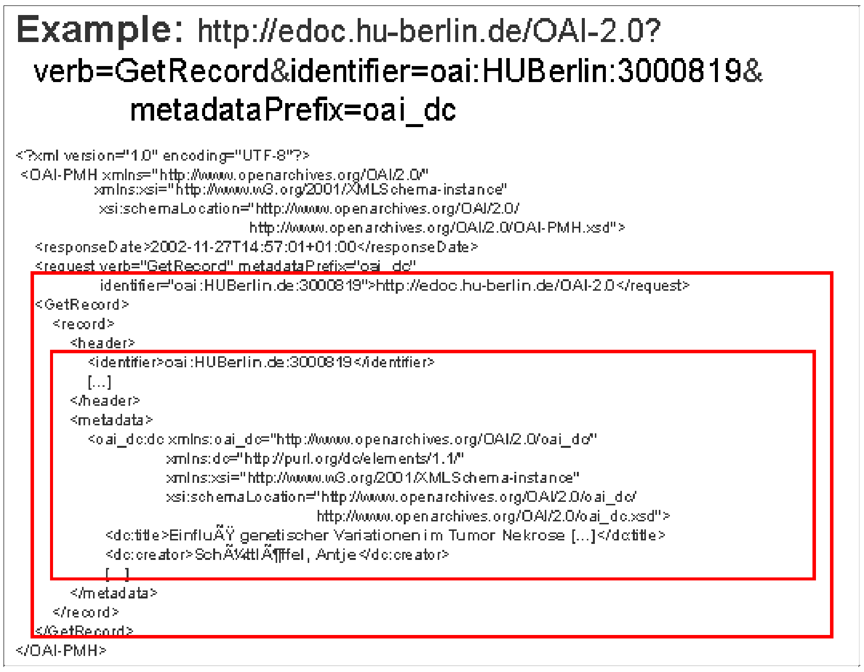


Example: http://edoc.hu-berlin.de/OAI-2.0?
verb=ListIdentifiers&from=2002-01-06&until=2002-01-08&
metadataPrefix=oai_dc&set=doctypes:dissertations

```
<?xml version="1.0" encoding="UTF-8"?>
<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">
  <responseDate>2002-10-22T17:49:49+01:00</responseDate>
  <request verb="ListIdentifiers" from="2002-01-03" until="2002-01-08" metadataPrefix="oai_dc"
           set="doctypes:dissertations">http://edoc.hu-berlin.de/OAI-2.0</request>
  <ListIdentifiers>
    <header>
      <identifier>oai:HUBerlin.de:3000819</identifier>
      <datestamp>2002-01-08</datestamp>
      <setSpec>doctypes</setSpec>
      <setSpec>doctypes:dissertations</setSpec>
      <setSpec>dnb</setSpec>
      <setSpec>dnb:dnb33</setSpec>
    </header>
    <header>
      <identifier>oai:HUBerlin.de:3000831</identifier>
      <datestamp>2002-01-07</datestamp>
      <setSpec>doctypes</setSpec>
      <setSpec>doctypes:dissertations</setSpec>
      <setSpec>dnb</setSpec>
      <setSpec>dnb:dnb27</setSpec>
    </header>
  </ListIdentifiers>
</OAI-PMH>
```

Example 2: response to GetRecord request Top

This example shows a response to a **`GetRecord`** request for an individual record specified by identifier.



Example: http://edoc.hu-berlin.de/OAI-2.0?
verb=GetRecord&identifier=oai:HUBerlin:3000819&
metadataPrefix=oai_dc

```
<?xml version="1.0" encoding="UTF-8"?>
 <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">
  <responseDate>2002-11-27T14:57:01+01:00</responseDate>
  <request verb="GetRecord" metadataPrefix="oai_dc"
          identifier="oai:HUBerlin.de:3000819">http://edoc.hu-berlin.de/OAI-2.0</request>
  <GetRecord>
    <record>
      <header>
        <identifier>oai:HUBerlin.de:3000819</identifier>
        [...]
      </header>
      <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>EinfluÃŸ genetischer Variationen im Tumor Nekrose [...]</dc:title>
          <dc:creator>SchÃ¼ttlÃ¶ffel, Antje</dc:creator>
          [...]
      </metadata>
    </record>
  </GetRecord>
</OAI-PMH>
```

Sources of further information Top

-- **Web sites and email lists** --

Open Archives Initiative (OAI) official site
http://www.openarchives.org/

> **OAI-PMH protocol specification**
> http://www.openarchives.org/OAI/openarchivesprotocol.html
>
> **OAI general mailing list**
> http://www.openarchives.org/mailman/listinfo/OAI-general/

OA-Forum expert reports and reviews of organisational and technical issues
Links from http://www.oaforum.org/documents/

Dublin Core
http://dublincore.org/

Contents | **1** | **2** | **3** | **4** | **5** | **6** | **Previous** | **Next** **Back to OA-Forum tutorial entry**

4. Implementing OAI-PMH

Contents of this part of *OAI for Beginners, the Open Archives Forum online tutorial*

General: first questions

Top

Before implementing OAI-PMH, you must consider a number of organisational decisions that will affect your implementation.

Data Provider

- Which data do I want to deliver?
- Which Service Providers do I want to provide with data?

Service Provider

- Which service, or services, do I want to provide, and to whom?
- From which Data Providers do I get the metadata?
- In what way will the metadata have to be processed in order to support service provision?

Data Provider & Service Provider

- Which aspects do we have to agree upon among Data Providers and Service Providers?
  - Update frequency?
  - Metadata formats?
  - Sets?
  - Subject schemes?
  - Acceptable use?

General: metadata formats

Top

As has already been noted, unqualified Dublin Core (DC) is the metadata format required for basic interoperability. However, within some subject areas and communities other metadata specifications may be required. It may be necessary to describe resources with complex structures in a specialised way, as is the case, for example, in the archival community. Whichever metadata format is chosen, agreement between Data Providers and Service Providers on its use must be reached, and the definition of an XML schema must be made publicly available for validation. The procedure for defining and declaring an XML schema is described in the final part of this tutorial. An OAI DC XML schema has already been made available. In addition, you may find another XML schema already in place that meets your requirements.

NOTE: At the time of preparation of this tutorial, a discussion had been initiated in the OAI community that may result in the status of unqualified DC from "mandatory" to "recommended". See the OAI official site for the latest news on this issue.

General: sets

Top

Support for the **`Set`** construct is optional within OAI. However, sets have been found to be particularly useful in supporting the provision of specialised services based on selectively harvested metadata. Sets define groups of metadata within a repository, and metadata may be grouped by any characteristic that provides sensible partitioning for selective harvesting. Examples of sets that have been defined within organisations and communities include those based on journal or series titles, on subject classifications or categories, on collections (e.g. topic or discipline based), on resource types and on authors of works. Individual metadata records may be included in one set, more than one set, or no sets at all. A repository may support Sets based on as many different defined groupings as it finds useful.

OAI sets may be hierarchical. In this case, members of child sets are harvested as part of their parent set. The meaning of sets or of set hierarchies is not defined in the OAI protocol. The definition of a set or set hierarchy may be internal to a repository, but is often based on an agreement between Data Providers or between Data Providers and Service Providers.

General: organisational structure

Top

An aggregator may set between Service Providers and some Data Providers. Where this is the case, Service Providers must be aware of the identity of the Data Providers that have been aggregated. This will enable Service Providers to avoid duplication that would arise from harvesting both the aggregator and the original Data Providers.

Service Providers that provide subject gateways will be able to implement selective harvesting if corresponding sets have been defined and implemented by the Data Providers they harvest.

General: tools for implementing OAI-PMH

Top

While you may want to develop your own OAI software, this is by no means necessary, as a number of software tools are available, many of them under open source license (or similar) terms. The OAI maintains a list of OAI software tools (http://www.openarchives.org/tools/) with links for the sources of the tools. At time of preparation of this tutorial, the list included Repository Explorer for interactive exploration and validation of OAI repositories, as well as tools with specific support for OAI-PMH, such as GNU EPrints (eprints.org) from Southampton University, DSpace from HP Labs and MIT Libraries, and the PHP OAI Data Provider from Oldenburg University. The tools you choose will depend on such considerations as the type of repository or service you are implementing and the technical skills available to you in-house. For example, if you are setting up an e-print archive then you may want to consider using the GNU EPrints software package, while DSpace provides a digital asset management framework that includes preservation considerations, and the advantage offered by PHP OAI Data Provider is support for on-the-fly output compression aiming at a significant reduction in data transfer load. A number of general software tools relating to XML and Unicode that may be useful for implementing OAI-conformant data and service providers are also listed by OAI.

In addition, about thirty OAI-related tools are described in the OA-Forum *Final Report on Technical Issues* (download from http://www.oaforum.org/documents/). This report also includes a detailed comparison of GNU EPrints and DSpace.

Data Provider: prerequisites

Top

These are the things you must, should, or may have in place in order to implement OAI-PMH as a Data Provider:

- **metadata** on resources ("items")
  These should be stored in a database (such as an SQL database). A file system may be necessary. It is necessary to have a unique identifier for each item.
- **Web server**, accessible via the Internet, e.g. Apache, IIS
- **programming interface / API**
  - e.g. Perl, PHP, Java-Servlet
  - web server extension
  - access to database (or filesystem)
  - not needed: session management
- **archive identifier / base URL**
- **unique identifier for each item**
- **metadata format** (one or more; at least: unqualified Dublin Core)
- **datestamps for metadata** (created / last modified)
- **logical set hierarchy** (may have)
  This is most usefully by agreement within communities, especially subject communities
- **flow control** by implementation of resumption token (optional, but 'larger' repositories should have it)

Data Provider: components and architecture

Top

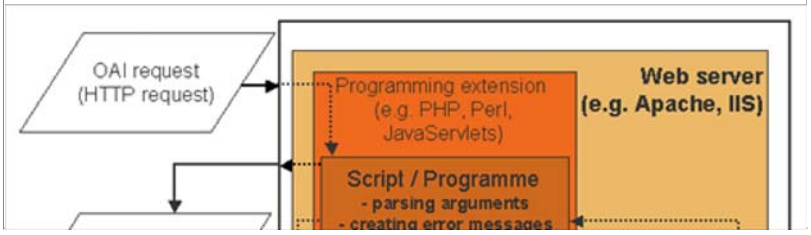
Components

Argument Parser validates OAI requests.

Error Generator creates XML responses with encoded error messages.

Database Query / Local Metadata Extraction retrieves metadata from the repository, according to the required metadata format.

XML Generator / Response Creation creates



XML responses with encoded metadata information.

Flow Control realises incomplete list sequences for 'larger' repositories. It uses resumption token as the control mechanism.

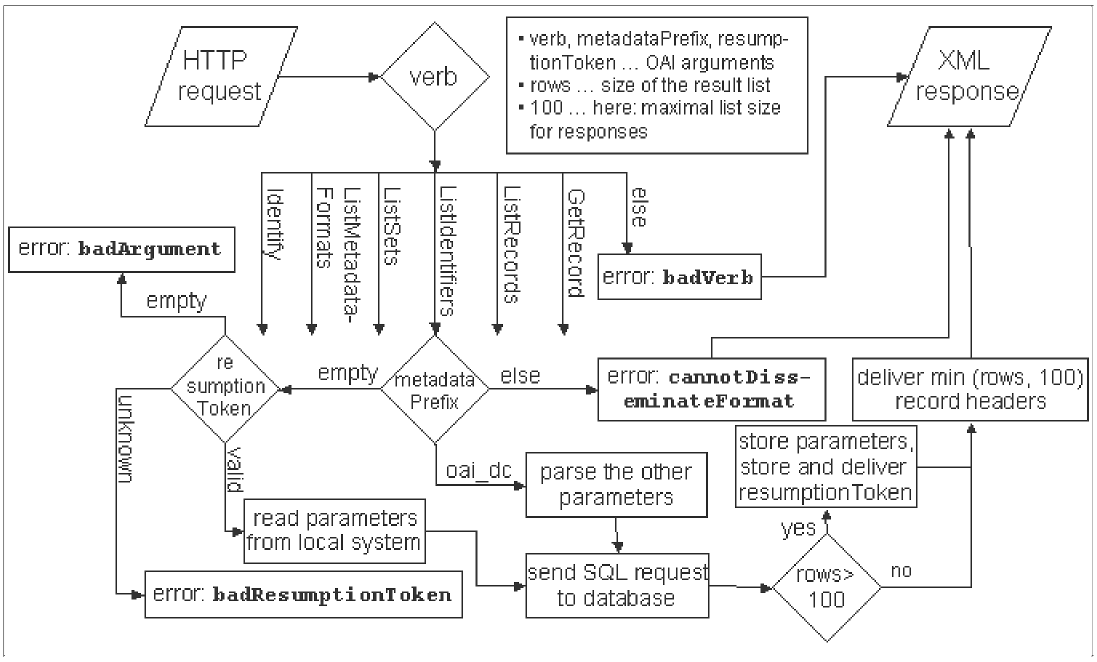
SQL request
SQL-Database
DB response
OAI Data Provider

This diagram illustrates an example architecture for a Data Provider.

Data Provider: example flow chart

Top

This diagram charts the flow from receiving an HTTP request to issuing an XML response to the request. It is an example showing the processing of one particular request type.


HTTP request
verb
• verb, metadataPrefix, resumptionToken ... OAI arguments
• rows ... size of the result list
• 100 ... here: maximal list size for responses
XML response
Identify
ListMetadata-Formats
ListSets
ListIdentifiers
ListRecords
GetRecord
else
error: badArgument
error: badVerb
empty
re sumption Token
empty
metadata Prefix
else
error: cannotDisseminateFormat
deliver min (rows, 100) record headers
unknown
valid
oai_dc
parse the other parameters
store parameters, store and deliver resumptionToken
read parameters from local system
yes
send SQL request to database
rows> 100
no
error: badResumptionToken

This is a flow chart for the processing of an example request within the Data Provider. In general, the diamonds represent conditions, and within the rectangles actions are described informally. When receiving an OAI request the Data Provider has to parse the query and firstly has to decide which of the six valid request types is issued or if the request type is illegal. The latter case (verb parameter has a nonstandard value) results in an error message to the Service Provider (**badVerb**). In case the issued request type is **ListIdentifers** the next parameter the parser has to check is **metadataPrefix** because this argument is mandatory for the request type **ListIdentifiers**.

If the parameter has not been provided the only possibility for the request to be valid is to have a **resumptionToken** parameter which has to be known to the Data Provider. In this case the Data Provider reads the locally stored parameters representing the arguments of the original request and cursor information indicating how many identifiers have already been delivered to the Service Provider. If the **resumptionToken** argument is emtpy as well or has an unkown value error messages have to be generated.

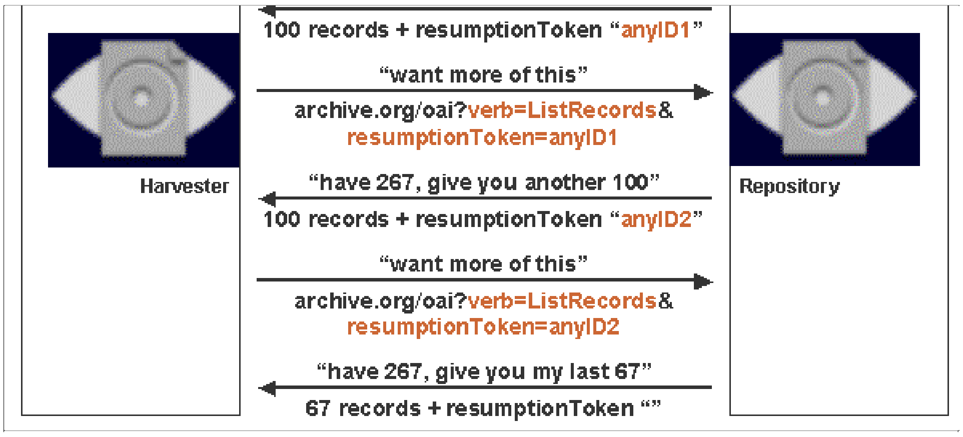
The only valid value for the **metadataPrefix** parameter is **oai_dc** because the example Data Provider assumed here can only deliver metadata sets in the unqualified Dublin Core schema. If this is the case the other optional parameters have to be parsed, which in the chart has been described informally for reasons of simplicity. The possible parameters are **from**, **until** and **set**. In this process, error messages have to be issued if the parameters have illegal values or if the query contains other parameters not allowed for this request type.

Subsequently, the given parameters received by the query or - in case of a resumed resumptionToken query - read from the local system have to be assembled to an SQL query which then has to be issued to the database. If this results in more than 100 records (100 in the example is the maximum number of delivered indentifiers at once) the Data Provider has to generate a new resumptionToken and to store it locally together with the query parameters and the cursor information. The resumptionToken has to be included in the XML response to the service provider as well. Of course, the XML response also contains the identifiers returned by the database.

Data Provider: resumption token

Top

This diagram illustrates the use of Resumption Token in controlling the flow of the dialogue between a Service Provider and Data Provider.


100 records + resumptionToken "anyID1"
"want more of this"
archive.org/oai?verb=ListRecords&
resumptionToken=anyID1
Harvester
"have 267, give you another 100"
100 records + resumptionToken "anyID2"
Repository
"want more of this"
archive.org/oai?verb=ListRecords&
resumptionToken=anyID2
"have 267, give you my last 67"
67 records + resumptionToken ""

Resumption Token should be implemented for handling "large" lists. It is initiated by Data Provider, and is used to store parameters (such as **`set`** or **`from`**) and the number of already delivered records.

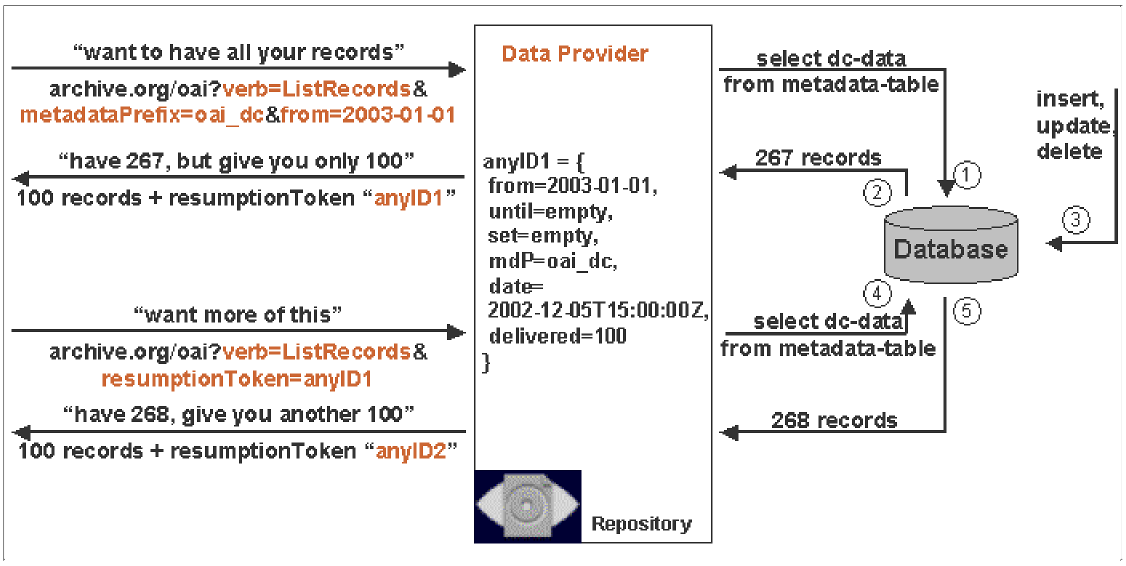
The Resumption Token may contain the following optional XML elements giving the Service Provider hints on the total length of the list to be expected:

- **`expirationDate`** (date after which the data provider does not guarantee the possibility of resuming the list)
- **`completeListSize`**
- **`cursor`** (already delivered records)

The protocol requires the ability of Data Providers to answer correctly if the most recent Resumption Token of a query is reissued. This feature allows Service Providers to recover from network errors without having to reissue the complete list from the beginning.

Data Provider: resumption token and database changes **Top**

There is a problem regarding the implementation of Resumption Token if the database changes during the course of a harvesting operation, as illustrated in the diagram below. The chart shows the possible case that between the first request and the resumed request the content of the Data Provider's database has been changed. If the Data Provider only remembers the the total number of already delivered records to the request the combination of the resumed lists may have inconsistencies.
There are two possible solutions. One solution is to duplicate data in a "request table". The other is to store the date of first request with the other parameters and use it like additional **`until`** argument.



Data Provider: data representation **Top**

Data Providers should use the following recommended data representations.

- Dates
  - Do use: 2002-12-05
  - Not: 2002-xx-xx, 2002, 05.12.2002
- Language code
  - Do use: eng, ger, ...
  - Not: en, de,
  - Not: english, german

Data Providers should use one separate XML element for each entity in the case of multiple values for a data element, as in the following examples.

- Author
  - Do use: **`<dc:creator>Smith, Adam</dc:creator>`**
    `<dc:creator>Nash, John</dc:creator>`

- Not: **`<dc:creator>Smith, Adam; Nash, John</dc:creator>`**

Data Provider: compression **Top**

Compression is a method to reduce traffic and enhance performance. It is optional for both Data Providers and Service Providers. Compression is handled on the HTTP level.

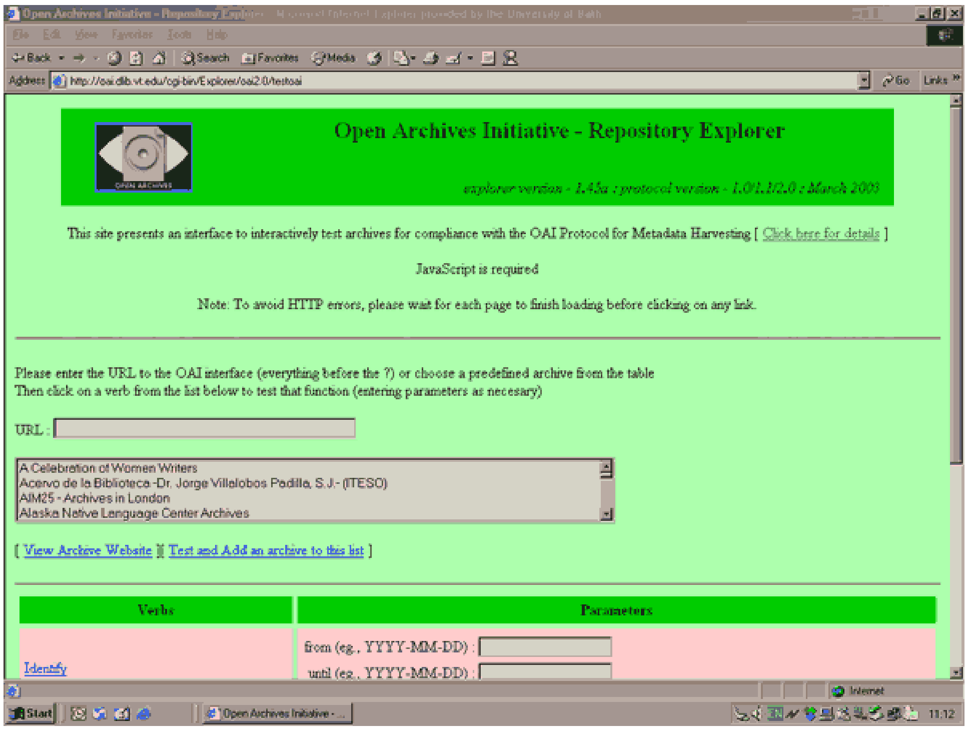
Harvesters may include an **`Accept-Encoding`** header in their requests for specifying compression preferences. Harvesters without an **`Accept-Encoding`** header always receive uncompressed data.

Repositories *must* support HTTP **`identity`** encoding. Repositories *should* specify supported encodings by including **`compression`** elements in the identify response.

Data Provider: testing **Top**

When you think your implementation is ready to run, create some OAI-PMH requests, send them to your OAI interface and check the results.

You can use the **Repository Explorer** at Vermont University to do this



Open Archives Initiative - Repository Explorer

explorer version - 1.45a : protocol version - 1.0/1.1/2.0 : March 2003

This site presents an interface to interactively test archives for compliance with the OAI Protocol for Metadata Harvesting [Click here for details]

JavaScript is required

Note: To avoid HTTP errors, please wait for each page to finish loading before clicking on any link.

Please enter the URL to the OAI interface (everything before the ?) or choose a predefined archive from the table
Then click on a verb from the list below to test that function (entering parameters as necesary)

URL :

A Celebration of Women Writers
Acervo de la Biblioteca -Dr. Jorge Villalobos Padilla, S.J.- (ITESO)
AIM25 - Archives in London
Alaska Native Language Center Archives

[View Archive Website][Test and Add an archive to this list]

Verbs	Parameters

Identify	from (eg., YYYY-MM-DD) :
	 until (eg., YYYY-MM-DD) :

Open Archives Initiative - Repository Explorer - Microsoft Internet Explorer provided by the University of Bath
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://oai.dlib.vt.edu/cgi-bin/Explorer/oai2.0/testoai
Go Links
Internet
Start
Open Archives Initiative - ...
11:12

(http://oai.dlib.vt.edu/cgi-bin/Explorer/oai2.0/testoai/) by browsing through your repository. Repository Explorer is an interactive, automatic compliance tester. It allows you to provide arguments via HTML forms. The responses are validated as conformant with OAI-PMH.

You can check your repository against each of the OAI-PMH verbs in turn, setting parameters where required for date ranges, metadataPrefix, identifier, set, and resumption token. Thus, all aspects of the protocol can be tested, and the results of queries are checked for conformance with the expected syntax.

Repository Explorer supports the following languages: Chinese, English, Spanish, French, German, Korean and Portuguese. You can choose how the results are displayed, either as parsed or raw XML, or both. There is also some provision for schema validation.

Data Provider: registration **Top**

Once you have assured yourself your implementation is working as expected, you can register it at the **official registration site** for Data Providers. (http://www.openarchives.org/data/registerasprovider.html)

The registration requires that you provide the base URL for your Data Provider implementation. OAI then performs an extensive **conformance** test (including tests for error conditions, among others), and information on incorrect behaviour (if any) that was found will be notified to you. In the case of conformance, your Data Provider implementation will be added to the official list. OAI performs regular checks on registered Data Providers to confirm that all is well.

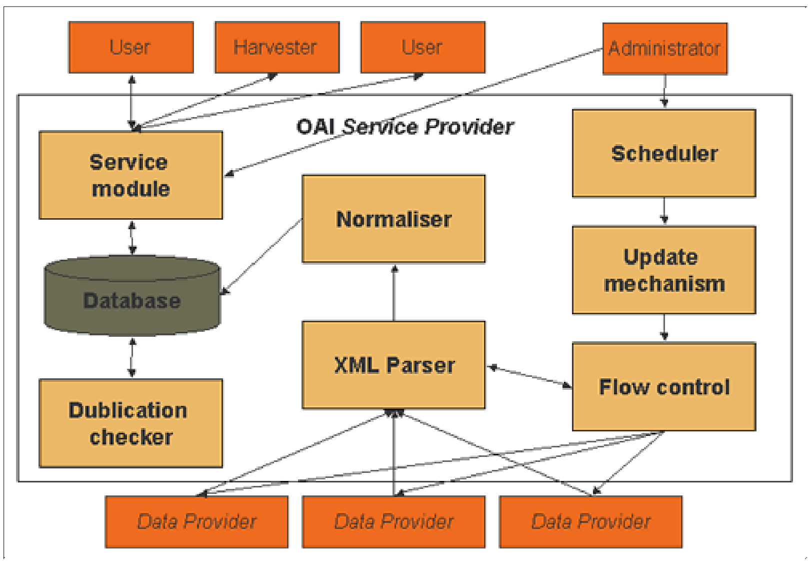
Service Provider: prerequisites **Top**

There are three technical infrastructure prerequisites for implementing an OAI-PMH Service Provider that will harvest metadata from Data Providers via OAI-PMH:

- an Internet-connected server
- a database system
  (relational or XML)
- a programming environment.
  (The programming environment must be one that can issue HTTP requests to web servers, can issue database requests, and includes an XML parser.)

Service Provider: components and architecture

Top



Archive management involves the selection of repositories to be harvested. Entries to your list of repositories to be harvested may be made manually or you can automatically add or remove archives using the official registry.

Request Component creates HTTP requests and sends them to OAI repositories (Data Provider). It demands metadata using the allowed verbs of the OAI-PMH. It may do selective harvesting using the **set** parameter.

Scheduler realises timed and regular retrieval of the associated archives. The simplest case would be manual initiation of the jobs, but this can be automated, e.g., as a cron job.

Flow Control is implemented via resumption token, partitioning of the result list into incomplete sections with a new request to retrieve more results. An HTTP error 503 (service not available) allows analysis of the response to extract a “retry-after” period.

Update Mechanism realises the consolidation of metadata which have been harvested earlier (merge old and new data). The easiest case would be to delete all ‘old’ metadata from each repository before harvesting it again. A reasonable alternative is to do an incremental update (**from** parameter) – insert *new* metadata and overwrite *changed / deleted* metadata (assignment using the unique identifiers).

XML Parser analyses the responses received from the repositories, with validation using the XML schema, and transforms the metadata encoded in XML into the internal data structure.

Normaliser transforms data in different metadata formats into a homogenous structure. It harmonises representation of, for example, date, author, language code. It may map between or translate different languages.

Database receives the output of the normaliser mapping the XML structure of the metadata into a relational database that will handle multiple values of elements. An alternative is to use an XML database.

Duplication Checker merges identical records from different data providers. One possibility for implementing this is by the unique identifier for each item (for example, by URN). However, this solution is often not easily practicable and is not risk or error free.

Service Module provides the actual service to the 'public'. The basis for a service provided is the harvested and stored records of the associated archives. That is, it uses only the local database for requests etc., and thus it does not make calls on the Data Providers during operation.

Service Provider: resumption token

Top

Resumption Token is optional from the Data Provider’s point of view. However, it is mandatory for Service Providers in order to retrieve complete lists in response to protocol requests that return lists (**ListRecords**, **ListIdentifiers**, **ListSets**) in order to resume sequences of incomplete lists. In order to provide flow control, Resumption Token must 'recognise' that the response from the Data Provider contains an incomplete list and then reissue the OAI request to the Data Provider in order to get next part of the list.

Service Provider: test and registration

Top

Test your OAI-PMH implementation by harvesting registered (as OAI conformant) Data Providers. There is a list of registered Data Providers on the OAI Web site, linked from the **Community page**, under the Interoperability Participants heading. Depending on your service, you may be planning to harvest from Data Providers that are not listed here. However, testing your implementation with some of the registered Data Providers will ensure that you have a working implementation of an OAI-PMH harvester.

This tutorial does not deal with implementing your end-user services, only with the implementation to harvest the records on which these services will, at least in part, be based. (Several examples of different types of OAI-based services are given at the end of this part of the tutorial.) Once you have tested the behaviour of your Service Provider harvesting, and once you have assured yourself your end-user service implementation is working as expected, you can register as a Service Provider at the **official registration site** (http://www.openarchives.org/service/registerasprovider.html). You will be asked to provide The full name of your service, a description of the type of service, the coverage offered (i.e., subject domain or topic), the URL of a Web page to be associated with your service, the email address of a contact person for your service, and a list of the Data Providers you harvest. This will be in the form of a Web page that you will put on your own server for ease of updating, and for which an HTML table template is provided. You simply have to email the URL for the page to OAI for addition to the list of Service Providers.

Seven key definitions

Top

Metadata
Structured information about resources (including both digital and non-digital resources). Metadata can be used to help support a wide range of operations on those resources. In the context of services based on metadata harvested via OAI-PMH, the most common operation is discovery and retrieval of resources.

Acceptable use
Terms and conditions setting out which Service Providers can do what with metadata harvested from a particular Data Provider or group of Data Providers. At the Cornell meeting (September 2000) where the foundations for the OAI protocol were agreed upon, an explicit choice was been made to hand over acceptable use issues to communities implementing the OAI protocol.

The OAI-PMH does not address issues of acceptable use of harvested metadata, although it does allow for the inclusion of an "about" container attached to each harvested metadata record. Typically such an "about" container could be used to specify the terms and conditions of the usage of a metadata record. In this way, individual communities can express terms and conditions regarding metadata use at the level of individual records. In addition to that, at the level of a repository, the response to the **`Identify`** verb allows for the inclusion of an open-ended "description" container. Communities could use this container to include terms and condition information for all metadata records in the repository. From a technical perspective, these provide hooks are there to allow communities to specify terms and conditions for the usage of metadata harvested from their repositories.

Aggregator
An OAI aggregator is both a Service Provider and a Data Provider. It is a service that gathers metadata records from multiple Data Providers and then makes those records available for gathering by others using the OAI-PMH.

Flow control
The management of the flow of data between Data Provider and Service Provider in order to assure that neither end of the transaction suffers overload.

Data representation
In this context, the format in which data of a particular type is set out in order to provide interoperability across repositories.

Value-added service
A service that is based on harvested metadata, and adds value for its users by means which may include normalisation and enriching of the harvested metadata for example. Types of services which may be offered include search services, citation linking, overlay journals, and peer-review services, among others.

Conformant
A repository is deemed to be OAI conformant if upon protocol testing by OAI it responds to each of the protocol requests with a response that validates with its XML schema, and also responds to malformed requests with the appropriate errors and exception conditions.

Sources of further information

Top

-- **Web sites and email lists** --

Open Archives Initiative (OAI) official site
http://www.openarchives.org

OAI-PMH protocol specification
http://www.openarchives.org/OAI/openarchivesprotocol.html

OAI-PMH implementation guidelines
http://www.openarchives.org/OAI/2.0/guidelines.htm

OAI tools
http://www.openarchives.org/tools/

OAI general mailing list
http://www.openarchives.org/mailman/listinfo/OAI-general/

OAI implementers discussion list
http://www.openarchives.org/mailman/listinfo/OAI-implementers/

Open Archives Forum

http://www.oaforum.org

OA-Forum **Review of Technical Issues**
Linked from http://www.oaforum.org/documents/

OA-Forum Information Resource
http://www.oaforum.org/oaf_db/

Dublin Core
http://dublincore.org

-- **Examples of Tools** --

Repository Explorer
http://oai.dlib.vt.edu/cgi-bin/Explorer/oai2.0/testoai/

GNU EPrints
http://software.eprints.org/

DSpace
http://www.dspace.org/

PHP OAI Data Provider
http://physnet.uni-oldenburg.de/oai/

-- **Examples of Service Providers** --

ARC - A Cross Archive Search Service (experimental research service)
http://arc.cs.odu.edu/

Dokumenten- und Publikationsserver der Humboldt-Universität zu Berlin (search service, German language user interface)
http://edoc.hu-berlin.de/oaisearch/

iCite (citation index)
http://icite.sissa.it/

NCSTRL—Networked Computer Science Technical Reference Library (search engine)
http://www.ncstrl.org/

my.OAI (value-added search interface to a selected list of metadata databases)
http://www.myoai.com/

Physnet (simple search interface to an experimental OAI harvester)
http://physnet.uni-oldenburg.de/oai/query.php

ProPrint (printing-on-demand service, German and English language user interfaces offered)
http://www.proprint-service.de/

Contents | **1** | **2** | **3** | **4** | **5** | **6** | **Previous** | **Next** **Back to OA-Forum tutorial entry**

5. XML Schemas and Support for Multiple Record Formats in OAI-PMH

Contents of this part of *OAI for Beginners, the Open Archives Forum online tutorial*

Basics of XML schemas for OAI-PMH

Top

OAI-PMH uses XML Schemas to define record formats. You can exchange any metadata you like using OAI-PMH as long as you can encode it as XML and define an XML Schema for it. OAI-PMH mandates the **oai_dc** schema as a minimum standard for interoperability.

OAI-PMH documentation also describes the use of XML schema for other formats, and provides additional XML schemas for:

- **rcf1807** (for RFC 1807 format metadata)
- **marc21** (recommended for MARC21 metadata, provided by the Library of Congress)
- **oai_marc** (for MARC format metadata)

Closer look at oai_dc, the mandated XML schema for OAI-PMH

Top

oai_dc is the simple metadata schema (based on unqualified Dublin Core) used as the mandatory ?Lowest Common Denominator? metadata record format in OAI-PMH. It defines a container schema that is OAI-specific, and is hosted on the OAI Web site. It imports a generic DCMES (DC Metadata Element Set) schema. The generic DCMES schema is hosted on the DCMI (Dublin Core Metadata Initiative) Web site.

The same model could be used for a qualified Dublin Core schema; that is, a container schema hosted by OAI and referencing the generic schema hosted by DCMI.

oai_dc – an example from a record

This is an example **oai_dc** record, as viewed via the **Repository Explorer**, showing the beginning of a full **GetRecord** response.

```
<?xml version="1.0" encoding="UTF-8"?>
 <OAI-PH 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">
   <responseDate>2003-03-15T16:16:51+01:00</responseDate>
   <request verb="GetRecord" metadataPrefix="oai_dc"
identifier="oai:HUBerlin.de:3000476">http://edoc.hu-berlin.de/OAI-2.0</request>
  <GetRecord>
    <record>
      <header>
        <identifier>oai:HUBerlin.de:3000476</identifier>
        <datestamp>1997-07-18</datestamp>
        <setSpec>pub-type</setSpec>
</header>
      <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>Melanchthon in seiner Zeit. In: Philipp Melanchthon 1497-1997</dc:title>
        <dc:creator>Selge, Kurt-Victor</dc:creator>
…
```

Three important things to notice picked out above:

The namespace for the oai_dc format:

xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"

The namespace for DCMES elements:
xmlns:dc="http://purl.org/dc/elements/1.1/"

The container schema associated with the oai_dc namespace:
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
    http://www.openarchives.org/OAI/2.0/oai_dc.xsd"

Thus, the **oai_dc** container schema for the http://www.openarchives.org/OAI/2.0/oai_dc/ namespace imports the DCMES schema from **http://dublincore.org/schemas/xmls/simpledc20021212.xsd**. It also defines a container element called 'dc' that lists the elements within the 'dc' container (from the DCMES namespace / schema) that are allowed in **oai_dc**.

Other metadata schemas may be used

Top

oai_dc is a simple format providing baseline interoperability. There are a number of reasons why it may not be suitable for your repository, service or community to share only **oai_dc**.

- **The 15 DCMES elements may not include enough of the elements you need.** In this case you can create a new schema incorporating the additional required elements along with those those in DCMES that you use.
- **The elements oai_dc does have may not be sufficiently precise for your metadata records**, as DCMES is an 'unqualified' metadata encoding schema. In this case you can get greater precision by creating a new schema adding 'encoding schemes' to existing DCMES elements.
- **DC may not be the metadata format you need.** In your community you may want to exchange metadata in another format, for example, in IMS/IEEE LOM for eLearning metadata or in ODRL (Open Digital Rights Language).

Adding new elements when oai_dc is not enough

Top

Creating a new schema by extending the **oai_dc** schema to add new elements involves the following tasks:

1. Create a name for the new schema
2. Create namespaces
3. Create the schema for the new elements
4. Create a 'container schema'
5. Validate your schema / records
6. Add to your repository?s "**ListMetadataFormats**"
7. Add to your repository?s other verbs
8. Test it worked and is valid

Next, we'll use a simple scenario to demonstrate these eight tasks step-by-step. Suppose we have a test repository containing some photos:

http://www.ukoln.ac.uk/metadata/oa-forum/workshop-photos/oai/nph-oai2.cgi

Currently the repository has metadata using oai_dc. We want to add an "Equipment Used" element, as this is not part of the DCMES.

Step 1: Name your format

The new metadata format needs a name. In this case, we'll choose the name "**wp_dc**" - following OAI's naming of "**oai_dc**" as a convention. (The two-letter code, 'wp', is short for 'workshop photos'.) However, the name could be anything you like. In this case alternative possibilities would be, for example, **wpdc** or **WP**

Step 2: Create Namespaces

We need two namespaces:

1. a namespace for the new format (**wp_dc**) that mixes both standard DC elements and any new ones
2. a namespace for the new **wp_dc** metadata element (the property "Equipment Used") that we will use in this format

Namespaces are declared as URIs. We will use:

- http://www.ukoln.ac.uk/metadata/oa-forum/workshop-photos/wp_dc/ (for 1, above)
- http://purl.org/oaforum/wpterms/ (for 2, above)

Note that the use of PURL for the elements namespace follows DCMI usage, but is not mandatory. However, both these namespace URIs should be under your control to ensure uniqueness and prevent re-use in the future. Namespace URIs do not need to resolve to anything.

Step 3: New terms schema

Next, we must create an XML schema for the new term. We will do this at:

http://www.ukoln.ac.uk/metadata/oa-forum/workshop-photos/wp_dc/20030818/wpterms.xsd

Notice the datestamp built in to the directory structure. This makes it easier to enhance the schema without breaking things that use the old one.

The schema for the new term defines the new element "**equipmentUsed**" and adds it to the **dc:any** group. It also defines a new container type "**wpterms:elementContainer**".

Step 4: Container Schema

We must also create a container schema for the **wp_dc** record format. In this case the schema is available at:

http://www.ukoln.ac.uk/metadata/oa-forum/workshop-photos/wp_dc/20030818/wp_dc.xsd

(Note again the use of a date stamp incorporated in the directory structure.) This simply imports the wpterms schema and then defines a container element '**wp_dc**' of type **wpterms:elementContainer**.

Step 5: Validate

In order to validate the records using our new schema, we next create some test records (or modify our existing ones) including all the elements we want to use. For ease of managing our validation process, we put these in a datestamped directory and use a meaningful file naming convention, such as

http://www.ukoln.ac.uk/metadata/oa-forum/workshop-photos/oai/wp_dc/20030818/test.xml

Now we can validate the records and schema with the XML schema validator at

http://www.w3.org/2001/03/webdata/xsv/

Step 6: ListMetadataFormats

The OAI-PMH verb **ListMetadataFormats** needs an awareness of the new format. Therefore, we need to modify our repository software (source code and/or configuration files) to support the new metadata format. We do this by adding information about the new format to our repository's response to the '**ListMetadataFormats**' request.For example:

```
...
<metadataFormat>
  <metadataPrefix>wp_dc</metadataPrefix>
  <schema>http://www.ukoln.ac.uk/metadata/oa-forum/workshop-photos/wp_dc/20030818/wp_dc.xsd</schema>
  <metadataNamespace>http://www.ukoln.ac.uk/metadata/oa-forum/workshop-photos/wp_dc/</metadataNamespace>
</metadataFormat>
...
```

Step 7: Other Verbs

We also need to ensure that the "**wp_dc**" format is available using the:

- **ListSets**
- **ListIdentifiers**
- **ListRecords**
- **GetRecord**

verbs. To do this, we must modify our repository's response to these verbs. Accept "MetadataPrefix" must be set to the new format name "**wp_dc**". Responses to requests will then return the appropriate records formatted according to the new schema when that is requested by a Service Provider.

Step 8: Testing - validate again

Finally, we use the **Repository Explorer** to test the new format. To do so, enter the following URL to the OAI interface of the repository

http://www.ukoln.ac.uk/metadata/oa-forum/workshop-photos/oai/nph-oai2.cgi

We must test to ensure that:

- all requests work with the new '**metadataPrefix**'

- the **oai_dc** format still works
- appropriate records are returned for each format
- responses validate correctly

Once all these conditions are met, we have a new format!

Summary - extending a format

- Decide on a name and some namespaces
- Develop XML schemas for the container and the new elements
- Create test records and validate
- Modify the repository (source code and/or configuration files) to support the new format
- Test and validate new repository output

When you want to use another metadata format

Top

You can take a similar approach with other metadata record formats. In the case of IMS/IEEE LOM and ODRL, XML schemas and namespaces have already been agreed. Deployment of these formats should be easier because you don't need to define you own schemas. However, XML schema specs are continually undergoing revisions at the time of preparing this tutorial, so sometimes it is difficult for applications like IMS to keep up with the changes.

Implementing an existing format

Top

To implement an existing metadata format, modify the ?ListMetadataFormats? response to include the format you wish to support. For example, for IMS:

```
...
<metadataFormat>
 <metadataPrefix>ims</metadataPrefix>
 <schema>http://www.imsglobal.org/xsd/imsmd_v1p2p2.xsd</schema>
 <metadataNamespace>http://www.imsglobal.org/xsd/imsmd_v1p2</metadataNamespace>
</metadataFormat>
...
```

Extend the other verbs (**ListSets**, **ListIdentifiers**, **ListRecords**, and **GetRecord** requests) to accept the 'metadataPrefix' set to 'ims' and return records formatted appropriately.

Summary

Top

OAI-PMH allows for any metadata format, so long as it is encoded in XML with an XML Schema. All repositories *must* support oai_dc for a minimum level of interoperability. If oai_dc does not have enough elements, you can extend it. If oai_dc is not precise enough, a qualified Dublin Core schema can be used. If oai_dc is not the right schema for your community or purpose, then use something else as well.

Seven key definitions

Top

PURL
A PURL is a Persistent Uniform Resource Locator. Functionally a PURL is a URL. However, instead of pointing directly to the location of an Internet resource, a PURL points to an intermediate resolution service. The PURL resolution service associates the PURL with the actual URL and returns that URL to the client. The client can then complete the URL transaction in the normal fashion. In Web parlance, this is a standard HTTP **redirect**.
(Definition quoted from **PURL** at http://purl.org)

URI
URI is the acronym for Universal Resource Identifier. URIs are strings that identify things on the Web. URIs are sometimes informally called URLs (Uniform Resource Locators), although URLs are more limited than URIs. URIs are used in a number of schemes, including the HTTP and FTP URI schemes.

XML namespace
An XML namespace is a collection of names, identified by a URI reference [RFC2396], which are used in XML documents as element types and attribute names. XML namespaces differ from the "namespaces" conventionally used in computing disciplines in that the XML version has internal structure and is not, mathematically speaking, a set.
(Definition quoted from **W3C—Namespaces in XML** at http://www.w3.org/TR/REC-xml-names/)

XML schemas
XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML documents.
(Definition quoted from **W3C Architecture Domain—XML schema** at http://www.w3.org/XML/Schema)

container
Containers are places in OAI-PMH responses where XML complying with any external schema may be supplied. Containers are provided for extensibility and for community specific enhancements. The OAI Implementation Guidelines lists the existing optional containers and provides links to existing schemas.

DCMI (Dublin Core Metadata Initiative)
The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models. DCMI's activities include consensus-driven working groups, global workshops, conferences, standards liaison, and educational efforts to promote widespread acceptance of metadata standards and practices.
(Definition quoted from **Dublin Core Metadata Initiative** at http://dublincore.org/)

DCMES (Dublin Core Metadata Element Set)
The Dublin Core metadata element set is a standard for cross-domain information resource description. Here an information resource is defined to be "anything that has identity". This is the definition used in Internet RFC 2396, "Uniform Resource Identifiers (URI): Generic Syntax", by Tim Berners-Lee et al. There are no fundamental restrictions to the types of resources to which Dublin Core metadata can be assigned.
(Definition quoted from **Dublin Core Metadata Initiative—Dublin Core Metadata Element Set, Version 1.1: Reference Description** at http://dublincore.org/documents/dces/)

Sources of further information

Top

Dublin Core official site
http://dublincore.org/

DCMI term declarations represented in XML schema language
http://dublincore.org/schemas/xmls/

Guidelines for implementing Dublin Core in XML
http://dublincore.org/documents/dc-xml-guidelines/

W3 Schools XML tutorials include, among others, the following:

W3 Schools XML tutorial
http://www.w3schools.com/xml/

W3 Schools XML Schema Tutorial
http://www.w3schools.com/schema/

OAI official site
http://www.openarchives.org/

OAI-PMH protocol specification
http://www.openarchives.org/OAI/openarchivesprotocol.html

OAI general mailing list
http://www.openarchives.org/mailman/listinfo/OAI-general/

OAI implementers mailing list
http://www.openarchives.org/mailman/listinfo/OAI-implementers/

Contents | **1** | **2** | **3** | **4** | **5** | **6** | **Previous** | **Next** **Back to OA-Forum tutorial entry**

6. Glossary

Contents of this part of *OAI for Beginners, the Open Archives Forum online tutorial*

Acceptable use

Top

Terms and conditions setting out which Service Providers can do what with metadata harvested from a particular Data Provider or group of Data Providers. At the Cornell meeting (September 2000) where the foundations for the OAI protocol were agreed upon, an explicit choice was been made to hand over acceptable use issues to communities implementing the OAI protocol.

The OAI-PMH does not address issues of acceptable use of harvested metadata, although it does allow for the inclusion of an "about" container attached to each harvested metadata record. Typically such an "about" container could be used to specify the terms and conditions of the usage of a metadata record. In this way, individual communities can express terms and conditions regarding metadata use at the level of individual records. In addition to that, at the level of a repository, the response to the `Identify` verb allows for the inclusion of an open-ended "description" container. Communities could use this container to include terms and condition information for all metadata records in the repository. From a technical perspective, these provide hooks are there to allow communities to specify terms and conditions for the usage of metadata harvested from their repositories.

Aggregator

Top

An OAI aggregator is both a Service Provider and a Data Provider. It is a service that gathers metadata records from multiple Data Providers and then makes those records available for gathering by others using the OAI-PMH.

Archive

Top

The term "archive" in the name Open Archives Initiative reflects the origins of the OAI in the e-prints community where the term archive is generally accepted as a synonym for repository of scholarly papers. Members of the archiving profession have justifiably noted the strict definition of an ?archive? within their domain; with connotations of preservation of long-term value, statutory authorization and institutional policy. The OAI uses the term ?archive? in a broader sense: as a repository for stored information. Language and terms are never unambiguous and uncontroversial and the OAI respectfully requests the indulgence of the professional archiving community with this

broader use of ?archive?.
(OAI definition quoted from FAQ on OAI Web site)

Conformant Top

A repository is deemed to be OAI conformant if upon protocol testing by OAI it responds to each of the protocol requests with a response that validates with its XML schema, and also responds to malformed requests with the appropriate errors and exception conditions.

Container Top

Containers are places in OAI-PMH responses where XML complying with any external schema may be supplied. Containers are provided for extensibility and for community specific enhancements. The OAI Implementation Guidelines lists the existing optional containers and provides links to existing schemas.

Data Provider Top

A Data Provider maintains one or more repositories (web servers) that support the OAI-PMH as a means of exposing metadata.
(OAI definition quoted from FAQ on OAI Web site)

Data representation Top

In this context, the format in which data of a particular type is set out in order to provide interoperability across repositories.

DC (Dublin Core) Top

Dublin Core (DC) is a metadata format defined on the basis of international consensus. The **Dublin Core** Metadata Element Set defines fifteen elements for simple resource description and discovery, all of which are recommended, and none of which are mandatory. DC has been extended with further optional elements, element qualifiers and vocabulary terms.
(Definition draws on UKOLN's **metadata glossary** and **Metadata in a nutshell** by Michael Day)

DCMI (Dublin Core Metadata Initiative) Top

The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models. DCMI's activities include consensus-driven working groups, global workshops, conferences, standards liaison, and educational efforts to promote widespread acceptance of metadata standards and practices.
(Definition quoted from **Dublin Core Metadata Initiative** at http://dublincore.org/)

DCMES (Dublin Core Metadata Element Set) Top

The Dublin Core metadata element set is a standard for cross-domain information resource description. Here an information resource is defined to be "anything that has identity". This is the definition used in Internet RFC 2396, "Uniform Resource Identifiers (URI): Generic Syntax", by Tim Berners-Lee et al. There are no fundamental restrictions to the types of resources to which Dublin Core metadata can be assigned.
(Definition quoted from **Dublin Core Metadata Initiative—Dublin Core Metadata Element Set, Version 1.1: Reference Description** at http://dublincore.org/documents/dces/)

Document-like object Top

A document-like object is a digital data unit that is comparable to a paper document. The term designates a relatively simple stable resource, and would not cover, for example multimedia artifacts or interactive services.

DTD (Document Type Definition) Top

A DTD is a formal specification of the structure of a document.

E-print Top

An e-print is an author self-archived document. In the sense that the term is ordinarily used, the content of an e-print is the result of scientific or other scholarly research.

Flow control Top

The management of the flow of data between Data Provider and Service Provider in order to assure that neither end of the transaction suffers overload.

Harvester Top

In OAI-PMH a harvester is a client application issuing OAI-PMH requests.

Harvesting **Top**

In the OAI context, harvesting refers specifically to the gathering together of metadata from a number of distributed repositories into a combined data store.

Identifier **Top**

In OAI-PMH an identifier is a unique key for an item in a repository.

Item **Top**

In OAI-PMH an item is a component of an repository from which metadata about a resource can be disseminated. An item has an unique identifier.

Interoperability **Top**

Interoperability is the ability of systems, services and organisations to work together seamlessly toward common or diverse goals. In the technical arena it is supported by open standards for communication between systems and for description of resources and collections, among others. Interoperability is considered here primarily in the context of resource discovery and access.

Metadata **Top**

Structured information about resources (including both digital and non-digital resources). Metadata can be used to help support a wide range of operations on those resources. In the context of services based on metadata harvested via OAI-PMH, the most common operation is discovery and retrieval of resources.

OAI (Open Archives Initiative) **Top**

OAI is an initiative to develop and promote interoperability standards that aim to facilitate the efficient dissemination of content.

OAI-PMH (OAI Protocol for Metadata Harvesting) **Top**

OAI-PMH is a lightweight harvesting protocol for sharing metadata between services.

Protocol **Top**

A protocol is a set of rules defining communication between systems. FTP (File Transfer Protocol) and HTTP (Hypertext Transport Protocol) are examples of other protocols used for communication between systems across the Internet.

PURL **Top**

A PURL is a Persistent Uniform Resource Locator. Functionally a PURL is a URL. However, instead of pointing directly to the location of an Internet resource, a PURL points to an intermediate resolution service. The PURL resolution service associates the PURL with the actual URL and returns that URL to the client. The client can then complete the URL transaction in the normal fashion. In Web parlance, this is a standard HTTP *redirect*.
(Definition quoted from **PURL** at http://www.purl.org)

Record **Top**

In OAI-PMH a record is metadata in a specific metadata format.

Repository **Top**

In OAI-PMH a repository is a network accessible server that is able to process OAI-PMH requests correctly.

Resource **Top**

A resource is anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., today's "weather report for Los Angeles"), and a collection of other resources. Not all resources are network "retrievable"; e.g., human beings, corporations, and bound books in a library can also be considered resources.
(Definition from **Guidelines for implementing Dublin Core in XML** by Andy Powell and Pete Johnston)

In OAI-PMH a resource is an object the metadata is "about". The nature of resources is not defined in the OAI-PMH. Thus, resources may be digital or non-digital.

Service Provider **Top**

A Service Provider issues OAI-PMH requests to data providers and uses the metadata as a basis for building

value-added services.
(OAI definition quoted from FAQ on OAI Web site)
A Service Provider in this manner is "harvesting" the metadata exposed by Data Providers

Set

Top

In the OAI-PMH a Set is an optional construct for grouping items in a repository.

URI

Top

URI is the acronym for Universal Resource Identifier. URIs are strings that identify things on the Web. URIs are sometimes informally called URLs (Uniform Resource Locators), although URLs are more limited than URIs. URIs are used in a number of schemes, including the HTTP and FTP URI schemes.

Value-added service

Top

A service that is based on harvested metadata, and adds value for its users by means which may include normalisation and enriching of the harvested metadata for example. Types of services which may be offered include search services, citation linking, overlay journals, and peer-review services, among others.

XML (Extensible Markup Language)

Top

XML is a language for creating other languages. It defines a means of describing data. XML can be validated against a DTD or schema setting out the elements of the language created. XML mappings exist for a number of metadata record formats.

XML namespace

Top

An XML namespace is a collection of names, identified by a URI reference [RFC2396], which are used in XML documents as element types and attribute names. XML namespaces differ from the "namespaces" conventionally used in computing disciplines in that the XML version has internal structure and is not, mathematically speaking, a set.
(Definition quoted from **W3C—Namespaces in XML** at http://www.w3.org/TR/REC-xml-names/)

XML schemas

Top

XML Schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the structure, content and semantics of XML documents.
(Definition quoted from **W3C Architecture Domain—XML schema** at http://www.w3.org/XML/Schema)

Contents | **1** | **2** | **3** | **4** | **5** | **6** | **Previous** | Next **Back to OA-Forum tutorial entry**