ISSN No. 0976-5697

Volume 1, No. 2, July-August 2010

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Exploring OAI-PMH: Open Archives Initiative Protocol for Metedata Harvesting

Shruti Sharma*
Dept of CE
YMCA, Faridabad, India
Shruti.mattu@yahoo.co.in

J.P.Gupta
JIIT, Noida, India
jp_gupta/jiit@jiit.ac.in

A.K.Sharma
Dept of CE
YMCA, Faridabad, India
ashokkale2@rediffmail.com

Abstract: There are factually hundreds of billions of highly valuable documents hidden in searchable databases that cannot be retrieved by conventional search engines. Searching on the Internet today can be compared to dragging a net across the surface of the ocean. There is a wealth of information that is deep, and therefore missed. Institutional repositories and digital libraries are adopting the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) to expose their belongings of white papers. The Open Archives Initiative has its roots in an effort to enhance access to e-print archives as a means of increasing the availability of scholarly communication. Continued support of this work remains a cornerstone of the Open Archives program. The fundamental technological framework and standards that are developing to support this work are, however, independent of the both the type of content offered and the economic mechanisms surrounding that content, and promise to have much broader relevance in opening up access to a range of digital materials

Keywords: Search Engines, Crawlers, Deep web, Digital Libraries, OAI-PMH

I INTRODUCTION

The World Wide Web has grown into an enormous collection of resources that may be utilized fully by its user. Due to the explosion in the size of the www, search engines are becoming progressively more imperative tool in locating relevant information. Such search engines rely on massive collections of web pages that are retrieved with the help of web crawlers, which traverse the web by following the hyperlinks thereafter storing them in a large database that is later indexed for efficient execution of the user queries. Substantial attention shall be given to augment the competence of web crawlers through more precise estimation of updates. This difficulty arises from the actuality that http does not support semantics of the form “what resources have changed since 2010-07-26?”

The oai-pmh [2, 6, 15] is a protocol to selectively harvest from data repositories. The protocol has a considerable impact in the field of digital libraries but it has yet to be embraced by the general web community. The OAI is an initiative to develop and promote interoperability standards that aim to facilitate the efficient dissemination of content. OAI-PMH provides an application-independent interoperability framework based on metadata harvesting. There are two classes of participants in the OAI-PMH framework as shown in Figure 1:

- Data Providers administer systems that support the OAI-PMH as a means of exposing metadata and

- Service Providers use metadata harvested via the OAI-PMH as a basis for building value-added service

Data providers handle the deposit and publishing of resources in a repository and render it 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' organization can play both roles, offering both data for harvesting and end-user services.

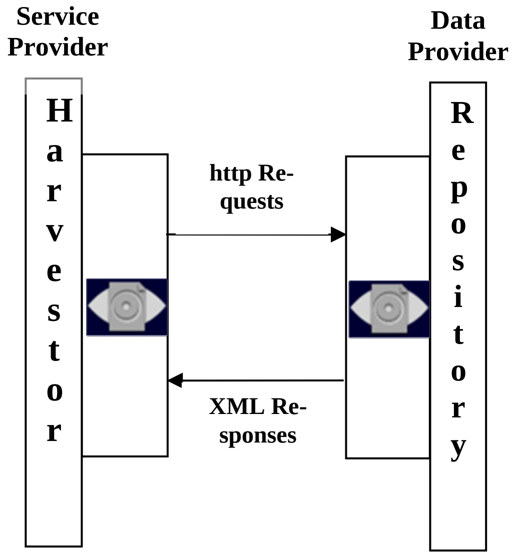

Figure 1. Functional Block diagram of OAI-PMH

II DEFINITIONS &CONCEPTS

OAI-PMH 2.0 [1, 13, 15, 18] is a low-barrier, HTTP-based protocol designed to allow incremental harvesting of XML metadata.

An OAI-PMH repository is a network accessible server that can process the six OAI-PMH protocol requests.

By issuing an OAI-PMH request to an OAI compliant repository, a harvester can obtain an XML-encoded list of all the repository's metadata records.

A *harvester* [7, 16, 18] operated by a service provider as a means of collecting metadata from repositories, is a client application that issues OAI-PMH requests. A *repository* [1, 19, 22] is a network accessible server that can process the six OAI-PMH requests managed by a data provider to expose metadata to harvesters. To allow various repository configurations, the OAI-PMH distinguishes between three following distinct entities related to the metadata as shown in Figure 2

- resource - A resource is the object that metadata is "about". The nature of a resource, whether it is physical or digital, or whether it is stored in the repository or is a constituent of another database.
- item - An item is a constituent of a repository from which metadata about a resource can be disseminated. . An item is conceptually a container that stores or dynamically generates metadata about a single resource in multiple formats, each of which can be harvested as **records** via the OAI-PMH. Each item has an **identifier** that is unique within the scope of the repository of which it is a constituent

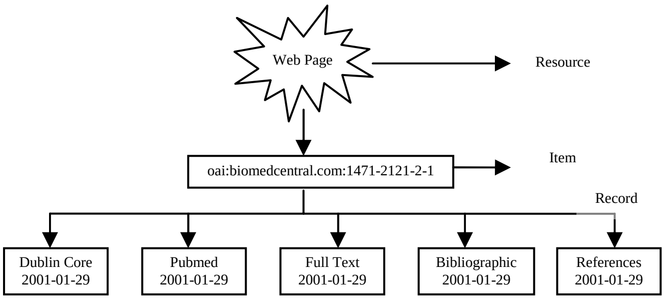

Figure 2. Data Model of OAI-PMH

- record - A record is metadata in a specific format returned as an XML-encoded byte stream in response to a protocol request to disseminate a specific metadata format from a constituent item. A record is identified unambiguously by the combination of the **unique identifier** of the item from which the record is available, the `metadataPrefix` identifying the metadata format of the record, and the datestamp of the record

III DATA PROVIDERS & SERVICE PROVIDERS

Components of Data Providers [15, 16, 17, 18, 19]

- 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.

Figure 3 illustrates an example architecture for a Data Provider.

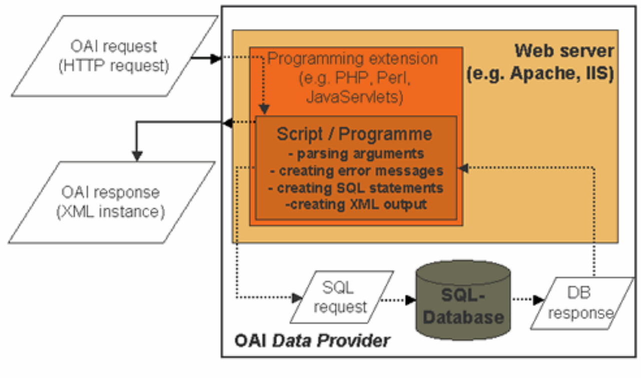

Figure 3. Architecture of Data Providers

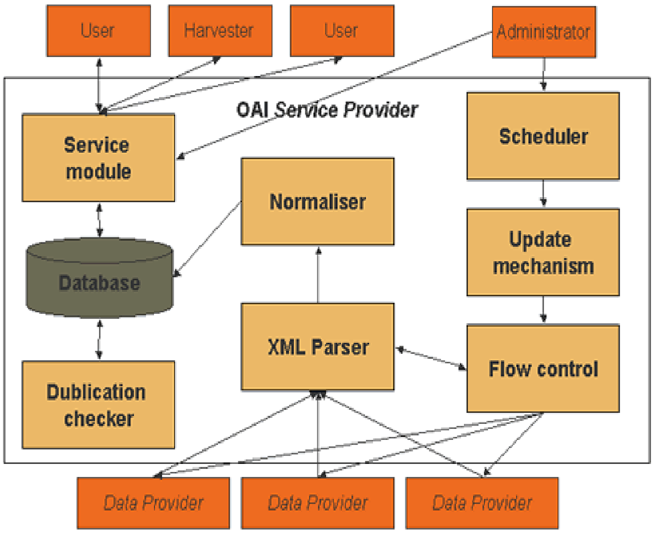

Figure 4. Architecture of Service Provider

Components of Service Providers [15, 16, 17, 18, 19]

- 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 realizes 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 realizes 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.
- Formalizer transforms data in different metadata formats into a homogenous structure. It harmonizes representation of, for example, date, author, language code. It may map between or translate different languages.
- Database receives the output of the normalizes 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.

Figure 4 illustrates example architecture for a Data Provider

IV FLEXIBLE DEPLOYMENT

OAI-PMH enables flexible deployment [15, 17]. Because it is a simple protocol based on HTTP and XML, it allows for rapid deployment. Systems can be deployed in a variety of configurations, as illustrated in the following diagrams (Figure 5, Figure 6 & Figure 7). Metadata and full-text resources are typically made freely available. OAI-PMH can also be used between closed groups; for metadata-sharing only; and in commercial applications.

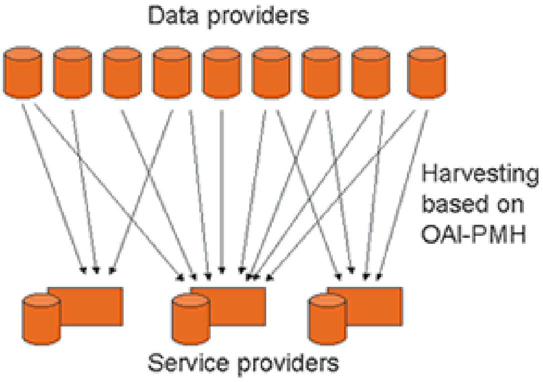

Figure 5: Multiple Service Providers can harvest from multiple Data Providers.

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 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.

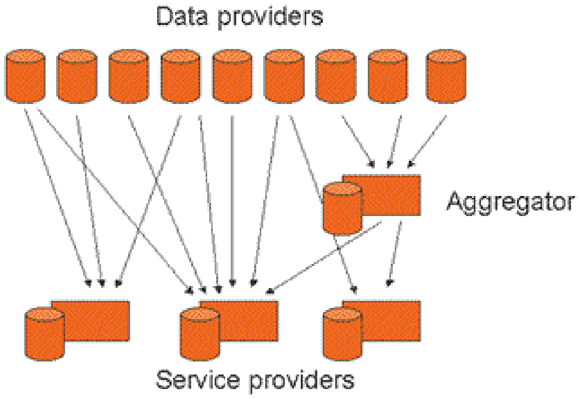

Figure 6: Aggregators can sit between Data Providers and Service Providers.

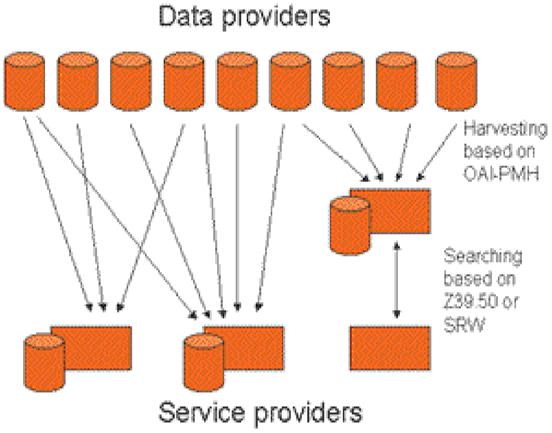

Figure 7: Harvesting combined with searching

An aggregator as shown in Figure 6, 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.

V ELEMENTS OF OAI-PMH

OAI-PMH **requests** are expressed as **HTTP** requests [19, 20]. A typical implementation uses a standard Web server that is configured to dispatch OAI-PMH requests to the software handling these requests. Request arguments are issued as **`GET`** or **`POST`** parameters. OAI-PMH supports six request types (known as "verbs"). Responses are encoded in XML syntax. OAI-PMH supports any metadata format encoded in XML.

A. HTTP Request Format

OAI-PMH requests can be submitted using either the HTTP `GET` or `POST` methods. `POST` has the advantage of imposing no limitations on the length of arguments. Repositories support both the `GET` and `POST` methods. There is a single base URL for all requests. The base URL specifies the Internet host and port, and **optionally** a path, of an HTTP server acting as a repository. Repositories expose their base URL as the value of the `base URL` element in the `Identify` response. In addition to the base URL, all requests consist of a list of *keyword arguments*, which take the form of `key=value` pairs. Arguments may appear in any order and multiple arguments **can** be separated by ampersands [&]. Each OAI-PMH request **can** have at least one `key=value` pair that specifies the OAI-PMH request issued by the harvester:

`key` is the string `'verb'`;

`value` is one of the defined OAI-PMH requests.

The number and nature of additional `key=value` pairs depends on the arguments for the individual request

a) Encoding an OAI-PMH request in a URL for an HTTP GET

URLs for `GET` requests have keyword arguments appended to the base URL, separated from it by a question mark [?]. For example, the URL of a GetRecord request to a repository with a base URL that is `http://an.oa.org/OAI-script` might be:

```
http://an.oa.org/OAI-script?
verb=GetRecord&identifier=oai:arXiv.org:
hep-th/9901001&metadataPrefix=oai_dc
```

b) Encoding an OAI-PMH request in an HTTP POST

Keyword arguments are carried in the message body of the HTTP `POST`. The `Content-Type` of the request will be `application/x-www-form-urlencoded`. For example, submitting the same request as above using the `POST` method would use just the base URL as the URL, with the format of the `POST` being:

```
POST http://an.oa.org/OAI-script
HTTP/1.0
Content-Length: 82
Content-Type: application/x-www-form-
urlencoded
verb=GetRecord&identifier=oai%3AarXiv.or
g%3Ahep-
th%2F9901001&metadataPrefix=oai_dc
```

B. XML Response Format

All responses to OAI-PMH requests [15, 17] will be well-formed XML instance documents. Encoding of the XML use the UTF-8 representation of Unicode. Character references are used since they allow XML responses to be treated as stand-alone documents that can be manipulated without dependency on entity declarations external to the document. The XML data for all responses to OAI-PMH requests **are** validated against the XML Schema. As can be seen from that schema, responses to OAI-PMH requests have the following common markup:

a) The first tag output is an XML declaration where the version is always 1.0 and the encoding is always UTF-8, eg: <?xml version="1.0" encoding="UTF-8" ?>

b) The remaining content is enclosed in a root element with the name OAI-PMH. This element has three attributes that define the XML namespaces used in the remainder of the response and the location of the validating schema:

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

c) For all responses, the first two children of the root element are:

- `responseDate` -- a <u>UTCdatetime</u> indicating the time and date that the response was sent. These are expressed in UTC.
- `request` -- indicating the protocol request that generated this response. The rules for generating the `request` element are as follows:
- The content of the `request` element is always be the **base URL** of the protocol request;
- The only valid attributes for the `request` element are the `keys` of the `key=value` pairs of protocol request. The attribute values must be the corresponding `values` of those `key=value` pairs;
- In cases where the request that generated this response did not result in an **error or exception condition**, the attributes and attribute values of the `request` element match the `key=value` pairs of the protocol request;
- In cases where the request that generated this response resulted in `a badVerb` or `badArgument` **error condition**, the repository returns the **base URL** of the protocol request only.

d) The third element is either:

- an `error` element that was used in case of an **error or exception condition**;
- an element with the same name as the verb of the respective OAI-PMH request.

C. Error and Exception Conditions

In event of an error or exception condition, repositories indicate OAI-PMH errors [5, 6], distinguished from **HTTP** `Status-Codes`, by including one or more `error` elements in the response. The following example demonstrates error handling in the case of an illegal verb argument. All request URLs shown from now on will be wrapped to make them more readable.

Request
http://arXiv.org/oai2? verb=nastyVerb

Response

```
<?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>2002-05-01T09:18:29Z</responseDate>
  <request>http://arXiv.org/oai2</request>
  <error code="badVerb">Illegal OAI verb</error>
</OAI-PMH>
```

While one `error` element is sufficient to indicate the presence of the error or exception condition, repositories report all errors or exceptions that arise from processing the request. Each `error` element has a `code` attribute that is from the following Table I; each `error` element also has a free text string value to provide information about the error that is useful to a human reader.

Table I. Error Description

Error Codes	*Description*	*Applicable Verbs*
`badArgument`	The request includes illegal arguments, is missing required arguments, includes a repeated argument, or values for arguments have an illegal syntax.	*all verbs*

`badResumptionToken`	The value of the `resumptionToken` argument is invalid or expired.	`ListIdentifiers`
		`ListRecords`
		`ListSets`

`badVerb`	Value of the `verb` argument is not a legal OAI-PMH verb, the verb argument is missing, or the `verb` argument is repeated.	*N/A*

`cannotDisseminateFormat`	The metadata format identified by the value given for the `metadataPrefix` argument is not supported by the item or by the repository.	`GetRecord`
		`ListIdentifiers`
		`ListRecords`

`idDoesNotExist`	The value of the `identifier` argument is unknown or illegal in this repository.	`GetRecord` `ListMetadataFormats`
`noRecordsMatch`	The combination of the values of the `from`, `until`, `set` and `metadataPrefix` arguments results in an empty list.	`ListIdentifiers` `ListRecords`
`noMetadataFormats`	There are no metadata formats available for the specified item.	`ListMetadataFormats`

`noSetHierarchy`	The repository does not support sets.	`ListSets`
		`ListIdentifiers`
		`ListRecords`

V PROTOCOL REQUESTS AND RESPONSES

This section lists the requests, or `verbs,` defined in the OAI-PMH. Arguments to the verbs are of three types:

- *required*, the argument **must** be included with the request (the `verb` argument is always *required*, as described in HTTP Request Format).
- *optional*, the argument **may** be included with the request.
- *exclusive*, the argument **may** be included with request, but **must** be the only argument (in addition to the `verb` argument).

A GetRecord

This verb is used to retrieve an individual metadata record [5,15, 22] from a repository. Required arguments specify the identifier of the item from which the record is requested and the format of the metadata that should be included in the record.

c) Arguments

- **`identifier`** a *required* argument that specifies the unique identifier of the item in the repository from which the record is disseminated.
- **`metadataPrefix`** a *required* argument that specifies the `metadataPrefix` of the format that should be included in the metadata part of the re-

turned record . A record should only be returned if the format specified by the `metadataPrefix` can be disseminated from the item identified by the value of the identifier argument. The metadata formats supported by a repository and for a particular record can be retrieved using the ListMetadata-Formats request.

d) Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments or is missing required arguments.
- **`cannotDisseminateFormat`** - The value of the `metadataPrefix` argument is not supported by the item identified by the value of the `identifier` argument.
- **`idDoesNotExist`** - The value of the `identifier` argument is unknown or illegal in this repository.

e) Examples

a) Request

Request a record in the Dublin Core metadata format
http://arXiv.org/oai2?verb=GetRecord&identifier=oai:arXiv.org:cs/0112017&metadataPrefix=oai_dc

b) Response

```
<?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>2002-02-08T08:55:46Z</responseDate>
  <request          verb="GetRecord"          identi-
fier="oai:arXiv.org:cs/0112017"
          metadataPre-
fix="oai_dc">http://arXiv.org/oai2</request>
  <GetRecord>
   <record>
    <header>
      <identifier>oai:arXiv.org:cs/0112017</identifier>
      <datestamp>2001-12-14</datestamp>
      <setSpec>cs</setSpec>
      <setSpec>math</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>Using Structural Metadata to Localize Experi-
ence of
              Digital Content</dc:title>
       <dc:creator>Dushay, Naomi</dc:creator>
       <dc:subject>Digital Libraries</dc:subject>
       <dc:description>With the increasing technical sophisti-
cation of
          both information consumers and providers, there is
          increasing demand for more meaningful experiences
of digital
          information. We present a framework that separates
digital
          object experience, or rendering, from digital object
storage
          and manipulation, so the rendering can be tailored to
          particular communities of users.
       </dc:description>
       <dc:description>Comment: 23 pages including 2 appen-
dices,
          8 figures</dc:description>
       <dc:date>2001-12-14</dc:date>
     </oai_dc:dc>
    </metadata>
  </record>
 </GetRecord>
</OAI-PMH>
```

B Identify

This verb is used to retrieve information about a repository. Some of the information returned is required as part of the OAI-PMH. Repositories also employ the Identify [12, 15] verb to return additional descriptive information.

a) Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments.

b) Response Format

The response include one instance of the following elements:

- `repositoryName` : a human readable name for the repository;
- `baseURL` : the base URL of the repository;
- `protocolVersion` : the version of the OAI-PMH supported by the repository;
- `earliestDatestamp` : a UTCdatetime that is the guaranteed lower limit of all datestamps recording changes, modifications, or deletions in the repository. A repository do **not** use datestamps lower than the one specified by the content of the `earliestDatestamp` element. `earliestDatestamp` must be expressed at the finest granularity supported by the repository.
- `deletedRecord` : the manner in which the repository supports the notion of deleted records. Legitimate values are `no` ; `transient` ; `persistent` with meanings defined in the section on deletion.
- `granularity:` the finest harvesting granularity supported by the repository. The legitimate values are `YYYY-MM-DD` and `YYYY-MM-DDThh:mm:ssZ` with meanings as defined in ISO8601.

The response also include one or more instances of the following element:

- `adminEmail` : the e-mail address of an administrator of the repository.
- `compression` : a compression encoding supported by the repository. The **recommended** values are those defined for the `Content-Encoding`

header in Section 14.11 of RFC 2616 describing HTTP 1.1.

- `description` : an extensible mechanism for communities to describe their repositories. For example, the `description` container could be used to include collection-level metadata in the response to the Identify request. Implementation Guidelines are available to give directions with this respect. Each `description` container is accompanied by the URL of an XML schema describing the structure of the description container.

c) Examples

a) Request
http://memory.loc.gov/cgi-bin/oai?
verb=Identify

b) Response

```
<?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>2002-02-08T12:00:01Z</responseDate>
  <request          verb="Identify">http://memory.loc.gov/cgi-
bin/oai</request>
  <Identify>
    <repositoryName>Library of Congress Open Archive Ini-
tiative
                  Repository 1</repositoryName>
    <baseURL>http://memory.loc.gov/cgi-bin/oai</baseURL>
    <protocolVersion>2.0</protocolVersion>
    <adminEmail>somebody@loc.gov</adminEmail>
    <adminEmail>anybody@loc.gov</adminEmail>
    <earliestDatestamp>1990-02-
01T12:00:00Z</earliestDatestamp>
    <deletedRecord>transient</deletedRecord>
    <granularity>YYYY-MM-DDThh:mm:ssZ</granularity>
    <compression>deflate</compression>
    <description>
      <oai-identifier
        xmlns="http://www.openarchives.org/OAI/2.0/oai-
identifier"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
        xsi:schemaLocation=
            "http://www.openarchives.org/OAI/2.0/oai-identifier
        http://www.openarchives.org/OAI/2.0/oai-
identifier.xsd">
        <scheme>oai</scheme>
        <repositoryIdenti-
fier>lcoa1.loc.gov</repositoryIdentifier>
        <delimiter>:</delimiter>
        <sampleIdenti-
fier>oai:lcoa1.loc.gov:loc.music/musdi.002</sampleIdentifier
>
      </oai-identifier>
    </description>
    <description>
      <eprints
        xmlns="http://www.openarchives.org/OAI/1.1/eprints"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:schemaLocation="http://www.openarchives.org/OAI/1.1/
eprints
        http://www.openarchives.org/OAI/1.1/eprints.xsd">
       <content>

<URL>http://memory.loc.gov/ammem/oamh/lcoa1_content.h
tml</URL>
         <text>Selected collections from American Memory at
the Library
               of Congress</text>
       </content>
       <metadataPolicy/>
       <dataPolicy/>
     </eprints>
   </description>
   <description>
     <friends

xmlns="http://www.openarchives.org/OAI/2.0/friends/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/f
riends/
        http://www.openarchives.org/OAI/2.0/friends.xsd">
      <baseURL>http://oai.east.org/foo/</baseURL>
      <baseURL>http://oai.hq.org/bar/</baseURL>
      <baseURL>http://oai.south.org/repo.cgi</baseURL>
    </friends>
  </description>
 </Identify>
</OAI-PMH>
```

C ListIdentifiers

This verb is an abbreviated form of `ListRecords` [2, 12, 15], retrieving only headers rather than records. Optional arguments permit selective harvesting of headers based on set membership and/or datestamp. Depending on the repository's support for deletions, a returned header **may** have a `status` attribute of "deleted" if a record matching the arguments specified in the request has been deleted.

a) Arguments

- **`from`** an *optional* argument with a UTCdatetime value, which specifies a lower bound for datestamp-based selective harvesting.
- **`until`** an *optional* argument with a UTCdatetime value, which specifies a upper bound for datestamp-based selective harvesting.
- **`metadataPrefix`** a *required* argument, which specifies that headers should be returned only if the metadata format matching the supplied `metadataPrefix` is available or, depending on the repository's support for deletions, has been deleted. The metadata formats supported by a repository and for a particular item can be retrieved using the `ListMetadataFormats` request.
- **`set`** an *optional* argument with a `setSpec` value , which specifies set criteria for selective harvesting.

- **resumptionToken** an *exclusive* argument with a value that is the flow control token returned by a previous ListIdentifiers request that issued an incomplete list.

b) Error and Exception Conditions

- **badArgument** - The request includes illegal arguments or is missing required arguments.
- **badResumptionToken** - The value of the `resumptionToken` argument is invalid or expired.
- **cannotDisseminateFormat** - The value of the `metadataPrefix` argument is not supported by the repository.
- **noRecordsMatch**- The combination of the values of the `from`, `until`, and `set` arguments results in an empty list.
- **noSetHierarchy** - The repository does not support sets.

c) Examples

a) Request

List the headers of records in the oldArXiv metadata format that are added, modified or deleted since January 15, 1998 in the set physics:hep. [URL shown without encoding for better readability].

http://an.oa.org/OAI-script?
verb=ListIdentifiers&from=1998-01-15&metadataPrefix=oldArXiv&set=physics:hep

b) Response

```
<?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>2002-06-01T19:20:30Z</responseDate>
  <request verb="ListIdentifiers" from="1998-01-15"
          metadataPrefix="oldarXiv"
          set="physics:hep">http://an.oa.org/OAI-
script</request>
  <ListIdentifiers>
   <header>
    <identifier>oai:arXiv.org:hep-th/9801001</identifier>
    <datestamp>1999-02-23</datestamp>
    <setSpec>physic:hep</setSpec>
   </header>
   <header>
    <identifier>oai:arXiv.org:hep-th/9801002</identifier>
    <datestamp>1999-03-20</datestamp>
    <setSpec>physic:hep</setSpec>
    <setSpec>physic:exp</setSpec>
   </header>
   <header>
    <identifier>oai:arXiv.org:hep-th/9801005</identifier>
    <datestamp>2000-01-18</datestamp>
    <setSpec>physic:hep</setSpec>
   </header>
   <header status="deleted">
    <identifier>oai:arXiv.org:hep-th/9801010</identifier>
    <datestamp>1999-02-23</datestamp>
    <setSpec>physic:hep</setSpec>
    <setSpec>math</setSpec>
   </header>
   <resumptionToken                    expirationDate="2002-06-
01T23:20:00Z"
     completeListSize="6"
     cursor="0">xxx45abttyz</resumptionToken>
 </ListIdentifiers>
</OAI-PMH>
```

D ListMetadataFormats

This verb [8, 12, 15] is used to retrieve the metadata formats available from a repository. An optional argument restricts the request to the formats available for a specific item.

a) Arguments

- **identifier** an *optional* argument that specifies the unique identifier of the item for which available metadata formats are being requested. If this argument is omitted, then the response includes all metadata formats supported by this repository. Note that the fact that a metadata format is supported by a repository does *not* mean that it can be disseminated from all items in the repository.

b) Error and Exception Conditions

- **badArgument** - The request includes illegal arguments or is missing required arguments.
- **idDoesNotExist** - The value of the `identifier` argument is unknown or illegal in this repository.
- **noMetadataFormats** - There are no metadata formats available for the specified item.

c) Examples

a) Request

List the metadata formats that can be disseminated from the repository `http://www.perseus.tufts.edu/cgi-bin/pdataprov` for the item with unique identifier `oai:perseus.tufts.edu:Perseus:text:1999.02.0119` [URL shown without encoding for better readability].

http://www.perseus.tufts.edu/cgi-bin/pdataprov?

verb=ListMetadataFormats&identifier=oai:perseus.tufts.edu:Perseus:text:1999.02.0119

b) Response

The response shows that 3 metadata formats are supported for the given identifier: oai_dc, olac and perseus. For each of the formats, the location of an XML Schema describing the format, as well as the XML Namespace URI is given.

```
<?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>2002-02-08T14:27:19Z</responseDate>
  <request verb="ListMetadataFormats"
    identifier="oai:perseus.tufts.edu:Perseus:text:1999.02.0119">
    http://www.perseus.tufts.edu/cgi-bin/pdataprov</request>
  <ListMetadataFormats>
   <metadataFormat>
     <metadataPrefix>oai_dc</metadataPrefix>
     <schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd
       </schema>
     <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/
       </metadataNamespace>
   </metadataFormat>
   <metadataFormat>
     <metadataPrefix>olac</metadataPrefix>
     <schema>http://www.language-archives.org/OLAC/olac-
0.2.xsd</schema>
     <metadataNamespace>http://www.language-
archives.org/OLAC/0.2/
      </metadataNamespace>
   </metadataFormat>
   <metadataFormat>
     <metadataPrefix>perseus</metadataPrefix>
     <schema>http://www.perseus.tufts.edu/persmeta.xsd</schema>
     <metadataNamespace>http://www.perseus.tufts.edu/per
smeta.dtd
       </metadataNamespace>
   </metadataFormat>
 </ListMetadataFormats>
</OAI-PMH>
```

E ListRecords

This verb [2, 15] is used to harvest records from a repository. Optional arguments permit selective harvesting of records based on set membership and/or datestamp. Depending on the repository's support for deletions, a returned header may have a `status` attribute of "deleted" if a record matching the arguments specified in the request has been deleted. No metadata will be present for records with deleted status.

a) Arguments

- **`from`** an *optional* argument with a UTCdatetime value, which specifies a lower bound for datestamp-based selective harvesting.
- **`until`** an *optional* argument with a UTCdatetime value, which specifies a upper bound for datestamp-based selective harvesting.
- **`set`** an *optional* argument with a `setSpec` value , which specifies set criteria for selective harvesting.
- **`resumptionToken`** an *exclusive* argument with a value that is the flow control token returned by a previous `ListRecords` request that issued an incomplete list.
- **`metadataPrefix`** a *required* argument (unless the exclusive argument `resumptionToken` is used) that specifies the `metadataPrefix` of the format that should be included in the metadata part of the returned records. Records should be included only for items from which the metadata format matching the `metadataPrefix` can be disseminated. The metadata formats supported by a repository and for a particular item can be retrieved using the ListMetadataFormats request.

b) Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments or is missing required arguments.
- **`badResumptionToken`** - The value of the `resumptionToken` argument is invalid or expired.
- **`cannotDisseminateFormat`** - The value of the `metadataPrefix` argument is not supported by the repository.
- **`noRecordsMatch`** - The combination of the values of the `from`, `until`, `set` and `metadataPrefix` arguments results in an empty list.
- **`noSetHierarchy`** - The repository does not support sets.

c) Examples

a) Request

List the records expressed in `oai_rfc1807` metadata format, that have been added or modified since January 15, 1998 in the `hep` subset of the `physics` set [URL shown without encoding for better readability].

http://an.oa.org/OAI-script?
verb=ListRecords&from=1998-01-15&set=physics:hep&metadataPrefix=oai_rfc1807

b) Response

```
<?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>2002-06-01T19:20:30Z</responseDate>
 <request verb="ListRecords" from="1998-01-15"
         set="physics:hep"
         metadataPrefix="oai_rfc1807">
         http://an.oa.org/OAI-script</request>
 <ListRecords>
  <record>
   <header>
    <identifier>oai:arXiv.org:hep-th/9901001</identifier>
    <datestamp>1999-12-25</datestamp>
    <setSpec>physics:hep</setSpec>
    <setSpec>math</setSpec>
   </header>
   <metadata>
    <rfc1807 xmlns=
      "http://info.internet.isi.edu:80/in-
notes/rfc/files/rfc1807.txt"
```

```
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
     xsi:schemaLocation=
      "http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1807.txt
       http://www.openarchives.org/OAI/1.1/rfc1807.xsd">
       <bib-version>v2</bib-version>
       <id>hep-th/9901001</id>
       <entry>January 1, 1999</entry>
       <title>Investigations of Radioactivity</title>
       <author>Ernest Rutherford</author>
       <date>March 30, 1999</date>
    </rfc1807>
   </metadata>
   <about>
     <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:publisher>Los Alamos arXiv</dc:publisher>
       <dc:rights>Metadata may be used without restrictions as
                   long as the oai identifier remains attached
                   to it.</dc:rights>
     </oai_dc:dc>
   </about>
  </record>
  <record>
   <header status="deleted">
     <identifier>oai:arXiv.org:hep-th/9901007</identifier>
     <datestamp>1999-12-21</datestamp>
   </header>
  </record>
 </ListRecords>
</OAI-PMH>
```

F ListSets

This verb [2, 12, 16] is used to retrieve the set structure of a repository, useful for selective harvesting.

a) Arguments

- **`resumptionToken`** an *exclusive* argument with a value that is the flow control token returned by a previous `ListSets` request that issued an incomplete list.

b) Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments or is missing required arguments.
- **`badResumptionToken`** - The value of the `resumptionToken` argument is invalid or expired.
- **`noSetHierarchy`** - The repository does not support sets.

c) Examples

a) Request

http://an.oa.org/OAI-script?
verb=ListSets

b) Response

```
<?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>2002-08-11T07:21:33Z</responseDate>
 <request                verb="ListSets">http://an.oa.org/OAI-
script</request>
 <ListSets>
  <set>
    <setSpec>music</setSpec>
    <setName>Music collection</setName>
  </set>
  <set>
    <setSpec>music:(muzak)</setSpec>
    <setName>Muzak collection</setName>
  </set>
  <set>
    <setSpec>music:(elec)</setSpec>
    <setName>Electronic Music Collection</setName>
    <setDescription>
      <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:description>This set contains metadata describing
            electronic music recordings made during the 1950ies
            </dc:description>
       </oai_dc:dc>
    </setDescription>
   </set>
   <set>
    <setSpec>video</setSpec>
    <setName>Video Collection</setName>
   </set>
 </ListSets>
</OAI-PMH>
```

VI CONCLUSION & FUTURESCOPE

http does not provide semantics to allow web servers to answer questions of the form "what resources do you have?" and "what resources have changed since 2004-12-27? A number of approaches have been suggested to add update semantics to http servers, including conventions about how to store indexes as well-known URLs for crawlers and a combination of indexes and http extensions. The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [15] has a very powerful and general set of update

semantics and is the standard for metadata interchange within the digital library community. Packages for implementing OAI-PMH repositories for XML files have been focused on highly constrained scenarios, not general web content and they do not integrate directly into the web server. mod_oai is an Apache module that implements OAI-PMH functionality directly into the Apache web server. Search engines would benefit by being able to index more content, and digital libraries would benefit by being able to share their content with search engines without incurring Web crawling overhead. One of the attractive features of HTTP and OAIPMH is that neither protocol requires registration with a central authority. This makes it easy to establish HTTP and OAI-PMH servers in an uncoordinated fashion, but it makes it hard to catalog the number of servers and their respective resources.

VII REFERENCES

[1] Hochstenbach, P., Jerez, H., Van de Sompel, H. The OAI-PMH Static Repository and Stati cRepository Gateway. In Proceedings of the 2003 ACM/IEEE Joint Conference on Digital Libraries, pp. 210-220.

[2] Lagoze, C., Van de Sompel, H., Nelson, M. L., Warner, S. The Open Archives Initiative Protocol for Metadata Harvesting. http://www.openarchives.org/OAI/ openarchivesprotocol.html.

[3] Ntoulas, A., Zerfos, P., Cho, J. Downloading Textual Hidden Web Content by Keyword Queries, In Proceedings of the 2005 ACM/IEEE Joint Conference on Digital Libraries.

[4] Raghavan, S., Garcia-Molina, H. Crawling the Hidden Web. In Proceedings of VLDB 2001, pp. 129-138.

[5] Suleman, H. OAI-PMH2 XMLFile File-based Data Provider, 2002. http://www.dlib.vt.edu/projects/OAI/ software/xmlfile/xmlfile.html.

[6] Van de Sompel, H., Nelson, M. L., Lagoze, C., & Warner, S. Resource Harvesting within the OAI-PMH Framework. D-Lib Magazine, 10(12), 2004.

[7] V. Crescenzi, G. Mecca, and P. Merialdo. “Roadrunner: Towards Automatic Data Extraction from Large Web Sites,”

[8] M. L. Nelson, H. Van de Sompel, X. Liu, T. Harrison, N. McFarland, “mod_oai: An Apache Module for Metadata Harvesting,” In Proceedings of ECDL 2005, Vienna, Austria, pp. 509-510.

[9] A. Ntoulas, P. Zerfos, J. Cho, “Downloading Textual Hidden Web Content by Keyword Queries,” In Proceedings of the Joint Conference on Digital Libraries (JCDL), June 2005, pp. 100-109.

[10] S. Raghavan and H. Garcia-Molina, “Crawling the Hidden Web,” In Proceedings of VLDB ‘01, 2001, pp. 129-138.

[11] C. Lee Giles, Kurt Bollacker, and Steve Lawrence. CiteSeer: An automatic citation indexing system. In Digital Libraries 06—The Third ACM Conference on Digital Libraries, pages 89-98, June 23-26 2007.

[12] AMeGA, Automatic Metadata Generation Applications, Retrieved April, 2005 http://ils.unc.edu/mrc/amega.htm

[13] Li X, Cheng Z, Sheng F, Fan X, and Ng P. A Document Classification and Extraction System with Learning Ability. Proceedings of the Fifth World Conference on Integrated Design and Process Technology, Dallas, Texas, June 2000.

[14] Liu X, Maly K, Zubair M, Nelson M. Arc: an OAI service provider for cross-archive searching. JCDL 2001: 65-66

[15] http://www.openarchives.org/OAI/2.0/

[16] Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, and Simeon Warner. “Resource Harvesting within the OAI-PMH Framework,” D-Lib Magazine, 10, 12, December, 2004, Corporation for National Research Initiatives. http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.html

[17] Michael L. Nelson, Herbert Van de Sompel, and Simeon Warner, “Advanced Overview of Version 2.0 of the Open Archives Initiative Protocol for Metadata Harvesting, ” ACM/IEEE Joint Conference on Digital Libraries, Houston, Texas, May 27 2003. http://www.cs.odu.edu/~mln/jcdl03/oai-2.0-adv.ppt

[18] Hussein Suleman and Edward Fox, “Beyond Harvesting: Digital Library Components as OAI Extensions”http://www.husseinsspace.com/publications/cstr_2002_odl_1.pdf

[19] Automated Building of OAI Compliant Repository from Legacy Collection Jianfeng Tang, Kurt Maly, Steven Zeil, Mohammad Zubair.

[20] H. Van de Sompel, J. A. Young, and T. B. Hickey.Using the OAI-PMH ... differently. D-Lib Magazine, 2003.

[21] Shruti Sharma, J.P.Gupta and A.K.Sharma. A novel architecture using agents for crawling OAI resources, IJCSE 2010

[22] Search Engine Coverage of the OAI-pmh corpus, IEEE 2006.