Extending The OAI Protocol as the Data Integration Framework for the Digital Library Network in the Third World

Ismail Fahmi
Ismail@itb.ac.id
Knowledge Management Research Group
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung, Indonesia

Ruri Muharto
rurie@transport.itb.ac.id
Knowledge Management Research Group
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung, Indonesia

Ismail Khalil Ibrahim
Ismail.khalil-ibrahim@scch.at
Software Competence Center Hangenberg
Hauptstrasse 99, A4232
Hangenberg, Austria

Abstract

The Open Archives Initiatives (OAI) Metadata Harvesting Protocol is developed to supply and promote an application-independent framework to solve problems of digital library interoperability. In this paper, we elaborate the design and implementation of the OAI protocol to solve the data integration problem found in the digital library network at the third world. The typical limitations are bandwidth and availability of the internet connection among the institutions. By extending the OAI protocol with the metadata posting mechanism we can show that the problem can be eliminated. The implementation case will be taken from the Indonesian Digital Library Network that currently integrating more than 30 digital library nodes. They are using dedicated and dial-up internet connection that in the most circumtance the bandwidth is low.

1. Introduction

Information is power and digital libraries are built to provide a unified infrastructure for supporting the creation of information sources, facilitating the movement of information across global networks and allowing effective and efficient interaction among knowledge producers, librarians, and information seekers [1].

A digital library is a vast collection of entities stored and maintained by multiple information sources including databases, image banks, file systems, email systems, the Web, and applications providing structured or semi-structured data.

The dramatic growth of digital libraries in recent years has not only simplified the access to existing information sources but also initiated the creation of numerous new sources. Paradoxically, this growth has made the task of finding, extracting and aggregating relevant information not easier.

This is because most of the information systems underlying digital libraries are physically distributed, heterogeneous in the way how information is stored, organized and managed, and comprise heterogeneous software and hardware platforms on which they reside. Additionally, they are autonomous in the sense that the content and format of data are determined by the organization owning the data not by the user [9].

Furthermore, the data in these information sources is mainly of composite multimedia nature comprising different media types such as text, video, images, and audio, and dynamic in the sense that the sources update both the content and the form of the data on their own, possibly on unknown schedule.

The goal of data integration in digital library network is to provide users with a uniform interface to access, relate, and combine data stored in multiple, autonomous, and possibly heterogeneous information sources.

The first crucial decision has to make is the selection of a method to achieve basic interoperability among repositories, with special emphasis placed on the ability to do cross-archival searching, and the most important is the system will be implemented at the third world where network connection is very expensive, temporary unavailable, slow, etc. It is generally considered that there are two major approaches to accomplish this: harvesting and federation.

Harvesting is when the digital library collects metadata from remote repositories, stores it locally and then performs searches on the local copy of the metadata. Federation is where the digital library sends the search criteria to multiple remote repositories and the results are gathered, combined, and presented to the user [16]. These are good for digital libraries that are connected through broadband and dedicated connection.

2. Motivation

Dedicated internet connection is a must if we want to develop a digital library network. Unfortunately, most of the institutions in the third world didn't have it. Many of them only have temporary or dial-up connection which is very slow for the digital library network. How can we search and access information from them, and how their students can search other digital library servers from their locations easily?

When they can access the digital library servers, it is typical that information searching will take very long time and expensive because of the very slow connection. Responses to their queries usually are disappointed. They will be reluctant to visit the sites again, and the digital library servers will have little visitors.

However, we can design the data integration framework for the digital library servers that does not require dedicated connection, broadband network, which is simple, and effective in the implementation. We can take a look at a new effort in digital library network: the Open Archive Initiatives (OAI). Its framework is considered to be able to solve these problems. Between both approaches, OAI opted for harvesting, primarily as a means of lowering the barrier to interoperability and data integration for providers of data.

In the OAI concept, there are Data Provider and Service Provider. Data Provider is the repository or digital library server that maintain the digital collections. Service Provider is an entity that harvests metadata from Data Providers.

To reduce complexity and transaction frequency at the Data Providers, we will make a central Service Provider that will harvest or receive metadata from all Data Providers in the digital library network. In this case, the merging of metadata to the central Service Provider will not only be done by harvesting, but also by **posting** the metadata especially by Data Providers that temporarily connected to the internet. The merged metadata then provided to be downloaded by all Data Providers that

want to keep it locally. Thus, the Data Provider will also act as local Service Provider for their local users.

Local Service Providers can harvest the whole metadata from central Service Provider, or only download specific metadata that related to interest of their users by applying filtering rules. The filtering will be based on subject or keyword of the metadata; type of metadata (gray literature, theses, or bibliography); and date of availability.

By storing the merged metadata into Data Provider's database, searching access to the metadata by local users will be responded quickly, because they didn't have to send the query out of their local area network. Users only have to connect to the Data Providers through internet if they want to download the files.

3. Approaches to Data Integration in Digital Libraries

A lot of research has been conducted in the field of data integration systems, and researchers have approached the problem from multiple points of view to provide a global schema that facilitates transparent access to distributed data, to integrate views, to facilitate interoperability without creating new objects, and to develop a new application with new objects that encompass existing applications [13].

The integration of data sources in digital libraries poses many challenges due to the differences in the data management systems (e.g., different vendors), in the data models (e.g., relational, network, ER, object-oriented etc.), in the query and data manipulation languages, in the data types (e.g., text, graphics, multimedia, hypermedia, etc.), in the format (e.g., structured, semistructured), and in the semantics.

Database interoperability [2, 12, 15, 14] is the ability of distributed, heterogeneous databases, which are independently created and administrated and have different semantics and schemas to cooperate and interoperate in a transparent way to the user while maintaining their autonomy and objectives.

The requirements and objectives for database interoperability are stated to be distribution transparency, heterogeneity transparency, no change to the existing database systems and applications, easy evolution of the system, execution of retrieval and updates, and performance comparable to homogenous distributed systems.

Different approaches and techniques [3, 20] have been proposed by the research community for data integration and several systems have been built with the goal of answering queries using a multitude of data sources.

Two common approaches have been advocated to building data integration systems [18]. The first approach is refereed to as virtual approach (Figure 1) to data integration. In the virtual approach, the user or the application poses the query. The data integration system accept the query, determine which set of information sources is capable to answer the query and generate the appropriate query plans for each information source. On obtaining the results from the information sources, the data integration system performs the appropriate translation, filtering and merging of the information and return the final answer to the user or application.

This process also may be referred to as a mediated approach, since the part that decomposes queries and combines results is often called the mediator.

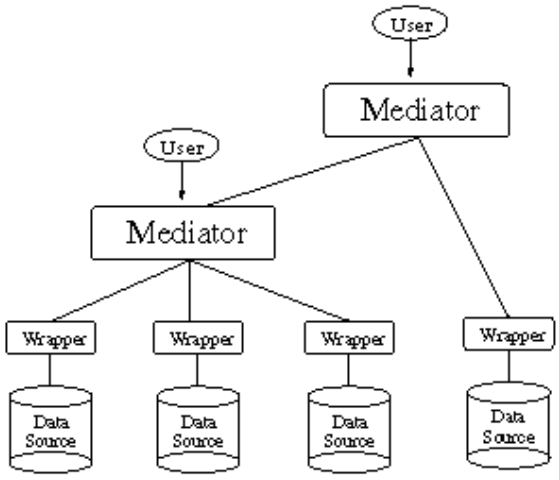

Figure 1. The virtual approach architecture

The second approach is called materialization approach (Figure 2) to data integration. In this approach information from each source that may be of interest to specific users or applications is extracted in advance, translated and filtered as appropriate, merged with relevant information from other sources and stored in a (logically) centralized repository.

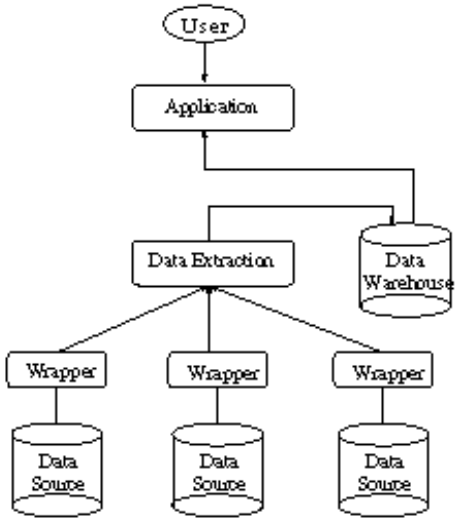

Figure 2. Materialization approach architecture

When a user or an application poses a query, the query is evaluated directly at the repository without accessing the original information sources. This approach is also referred to as data warehousing since the repository serves as a warehouse storing the data of interest.

The virtual approach to integration is appropriate when (a) the number of information sources is very large, (b) the data is changing very rapidly, (c) for clients with unpredictable needs and (d) for queries that operate over vast amounts of data from very large numbers of information sources (e.g., the World Wide Web). However, the virtual approach may incur inefficiency and delay in query processing, especially when (a) queries are issued multiple times, (b) information sources are slow, expensive or periodically unavailable and (c) significant processing is required for the translation, filtering and merging steps. In cases where information sources do not permit ad hoc queries, the virtual approach is simply not feasible. In the warehousing approach, the integrated information is available for immediate querying and analysis by clients. Thus, the warehousing approach is appropriate for (a) clients requiring specific predictable portions of the available information, (b) clients requiring high query performance but not necessarily over the most recent state of the information; (c) environments in which native applications at the information sources require high performance (large multi-source queries are executed at the warehouse instead) (d) clients wanting access to private copies of the information so that it can be modified, annotated, summarized and so on and (e) clients wanting to save information that is not maintained at the sources However, the data warehousing approach may incur that the warehouse should be updated each time the data is changed.

For these reasons, most of the recent research has focused on the virtual approach in building data integration systems and especially in building web data integration systems.

Figure (3) illustrates the different components of this system [19].

Users of data integration systems do not pose queries directly in the schema in which the data is stored. Instead the user poses queries on a *mediated schema* (often referred to as a *global schema*), which describes the contents of data sources and exposes the aspects of the data that might be of interest to the user [11].

A mediated schema is a set of virtual relations (in the sense that they are not actually stored anywhere), which are designed for a particular data integration application. As a consequence, the data integration system must first reformulate the user query into a query that refers directly to the schemas in the sources. In order for the system to be able to do this, it needs to have a set of source descriptions, specifying the semantic mapping between the relations in the sources and the relations in the mediated schema. These descriptions specify the relationship between the relations in the mediated schema and those in the local schemas of the sources. The description of a data source specifies its contents (contains technical reports in our motivational example), attributes (titles, subjects), constraints on its contents (access methods), completeness and reliability, and finally its query processing capabilities (can perform selections or can answer arbitrary SQL queries).

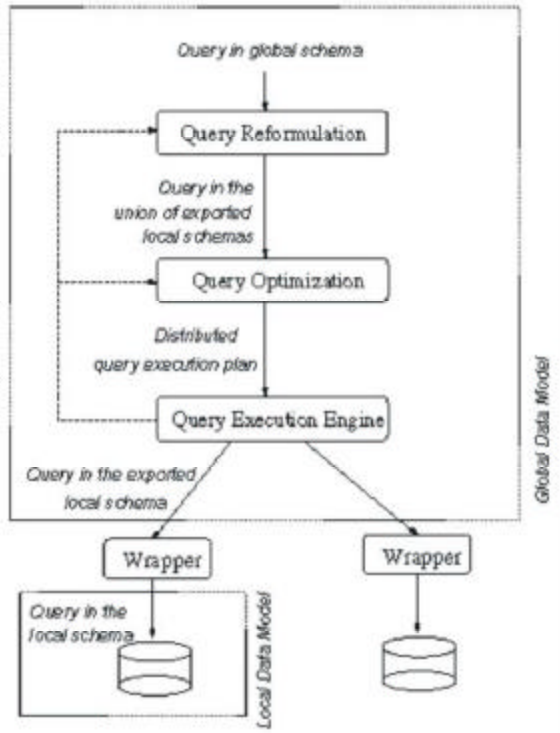

Figure 3. Components of data integration system

After the minimal set of data sources has been selected for a given query, a key problem is to find the optimal query execution plan for this query. The query execution plan is an imperative program that specifies exactly how to evaluate the query. In particular, the plan specifies the order in which to perform the different operations in the query (join, selection, projection), a specific algorithm to use for each operation (for example sort? merge join, hash? join) and the scheduling of different operators. Typically, the optimizer selects a query execution plan by searching a space of possible plans and comparing their estimated cost. To evaluate the cost of a query execution plan the optimizer relies on extensive statistics about the underlying data, such as sizes of relations, sizes of domains and selectivity of predicates. Finally, the query execution plan is passed to the query execution engine, which evaluates the query.

The system communicates with the remote data sources through wrappers. A wrapper is a program that is specific to a data source, whose task is to translate data from the format of the source to a format that can be manipulated by the data integration system. For example, if the data source is a Web site, the task of the wrapper is to translate the query to the source's interface and when the answer is returned as an HTML document, it needs to extract a set of tuples from that document.

4. Metadata Harvesting as a Simple and Effective Framework

Giving two approaches above, we can take a conclusion that the materialization or harvesting approach is the best fit with our need for digital library network. We need a fast query response; information changes are not rapid; and we can predict the need of our users.

Virtual or federation approach is a more expensive mode of operation in terms of network and search system constraints since each repository has to support a complex search language and fast real-time responses to queries. Harvesting requires only that individual archives be able to transfer metadata to the central DL. The frequency of queries, quantity of metadata, and availability of network resources also factor into this comparison but, in general, federation places a greater burden on the remote sites while harvesting reduces the demand on remote sites and concentrates the processing at the central DL site [16].

5. Basic Concept of OAI Metadata Harvesting Protocol

The OAI Metadata Harvesting Protocol (or referred as the OAI protocol in the remainder of this document) is to support and promote application-independent interoperability framework that can be used by a variety of communities who are engaged in publishing content on the Web. The OAI protocol described in this document permits *metadata harvesting*. The result is an interoperability framework with two classes of participants [17]:

- *Data Providers* administer systems that support the OAI protocol as a means of exposing metadata about the content in their systems;
- *Service Providers* issue OAI protocol requests to the systems of data providers and use the returned metadata as a basis for building value-added services.

The data flow between Data Providers and Service Providers is described by figure 4.

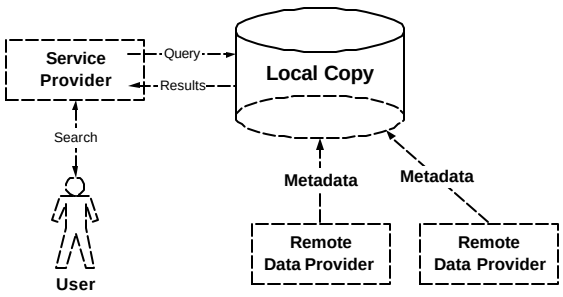

Figure 4. Data flow between Service Provider and Data Provider

5.1. Definitions and Concepts

The OAI protocol uses term *repository* to refer a network accessible server to which OAI protocol requests can be submitted. The requests are embedded in HTTP. Any OAI-compliant repositories can decode the requests and give outputs – metadata - in the form of records.

A record is an XML-encoded byte stream that is returned by a repository in response to an OAI protocol request for metadata from an item in that

repository. The OAI records are organized into *header, metadata,* and *about*.

Header is necessary for the harvesting process, and consists of two parts: *unique-identifier,* the key for extracting metadata from an item in a repository; and *datestamp* of creation, deletion, and last date of modification. *Metadata* is a single manifestation of a metadata from an item. The OAI protocol supports multiple format of metadata. *About* is an optional container to hold data about the metadata of the record, such as rights information, term and conditions for usage, etc[17].

Example:

```
<header>
  <identifier>oai:gdl:0109200
    </identifier>
  <datestamp>2001-09-01</datestamp>
</header>
<metadata>
 <dc xmlns= "http://purl.org/dc/
elements/1.1/"
      xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance"
      xsi:schemaLocation= "http://purl.org/
dc/elements/1.1/
http://openarchives.org/OAI/1.1/dc.xsd">
  <title>Ganesha Digital Library is Born to
Struggle with the Digital Divide</title>
  <creator>Ismail Fahmi</creator>
  <description>This paper describe an
effort of developing  digital library
network in Indonesia. The goal is to manage
local content to improve the information
literacy</description>
  <date>2001-09-01</date>
  <type>e-print</type>
  <identifier>
http://gdlhub.indonesiadln.org/go.php?id=01
09200 </identifier>
 </dc>
</metadata>
<about>
   <ea xmlns="http://www.arXiv.org/eprints-
about"
    xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
    xsi:schemaLocation= "
http://www.arXiv.org/eprints-about
http://www.arXiv.org/eprints-about.xsd">
    <archive>KMRG ITB</archive>
    <usage>Verbatim copying is
allowed</usage>
   </ea>
</about>
```

There is one more terminology used by the OAI protocol, set. A *set* is an optional construct for grouping items in a repository for the purpose of selective harvesting of records. Each node in the hierarchy is a Set which has: `a setTag`, `a setName`, and `a setSpec`. Because there is no pre-defined semantic for what constitutes a set so any use of set must have an explicit agreement between Data Providers and Service Providers [16].

5.2. Protocol Features

All of the OAI protocol requests are expressed as HTTP requests either using GET or POST method. The format of the requests is:

```
BASE-URL?key=value&key=value…
```

`BASE-URL` specifies internet host and port of a repository, and a path at the server as the handler of the OAI protocol requests. For example:

```
http://gdlhub.indonesiadln.org/OAI/resp
onse.php
```

All requests also consist of a list of *keyword argument* in the form of *key=value* pairs that depend on the arguments for the individual protocol request. For example:

key = string '`verb`'
value=one of defined requests

Using HTTP GET method, the keyword arguments appended to `BASE-URL` separated by a question mark [?]:

```
http://gdlhub.indonesiadln.org/OAI/respo
nse.php?verb=GetRecord&identifier=OAI:GDLHU
B:2001-31&metadataPrefix=oai_dc
```

Responses to protocol requests are formatted as HTTP responses, with appropriate HTTP header fields. Every OAI protocol request returns a Content-

Type of text/xml. Encoding of the XML is done using the UTF-8 representation of Unicode.

For example, the reply to the GetRecord protocol request shown above will be of the form:

```
<?xml version="1.0" encoding="UTF-8" ?>
<GetRecord
  xmlns="http://www.openarchives.org/OAI/
   1.1/OAI_GetRecord"
  xmlns:xsi="http://www.w3.org/2001/XMLSche
ma-instance"
  xsi:schemaLocation="http://www.openarchiv
es.org/OAI/1.1/OAI_GetRecord
     http://www.openarchives.org/OAI/1.1/OA
I_GetRecord.xsd">
  <responseDate>2001-09-01T08:14:23-
06:00</responseDate>
  <requestURL>http://gdlhub.indonesiadln.or
g/go.php?verb=GetRecord
    &amp;identifier=oai%3Agdl%3A0109200
    &amp;metadataPrefix=oai_dc</requestURL>
     list of records
</GetRecord>
```

6. The OAI Protocol Requests and Responses

There are 6 OAI service requests that can be made to a repository through HTTP request. The responses will be encapsulated using XML format. The following are the basic service requests [16,17]:

- *GetRecord* retrieves the metadata for a single object in a specified metadata format.
- *Identify* is a request for information about the repository as a whole. Returned is such information as the name of the repository, the version of the protocol, and the email address of the administrator.
- *ListIdentifiers* lists identifiers for all objects or within a given date range and/or within a given set.
- *ListMetadataFormats* will return the list of all metadata formats supported by the archive.
- *ListRecords* lists complete metadata for all objects or within a given date range and/or within a given set.
- *ListSets* lists the sets (and subsets, recursively) contained within the repository.

7. Extending the OAI Protocol for Implementation at the Indonesian Digital Library Network

Indonesian Digital Library Network (IndonesiaDLN) is a new digital library community in Indonesia, launched on June 2001. Its main purpose is to manage the local content information from any level of institution. The collections are not only theses and dissertation, but also gray literature, clipping, course material, distance learning material, etc. Currently there are more than 20 institutions joined the network, mostly from academic institutions. More than 30 institutions are in progress of developing their digital library and join the network [8].

Its Data Provider members are from individual (personal digital library), internet café, and institution (research, education, NGO, government, business, etc).

Figure 5 shows the distribution map of the IndonesiaDLN nodes (per March 2002)[6].

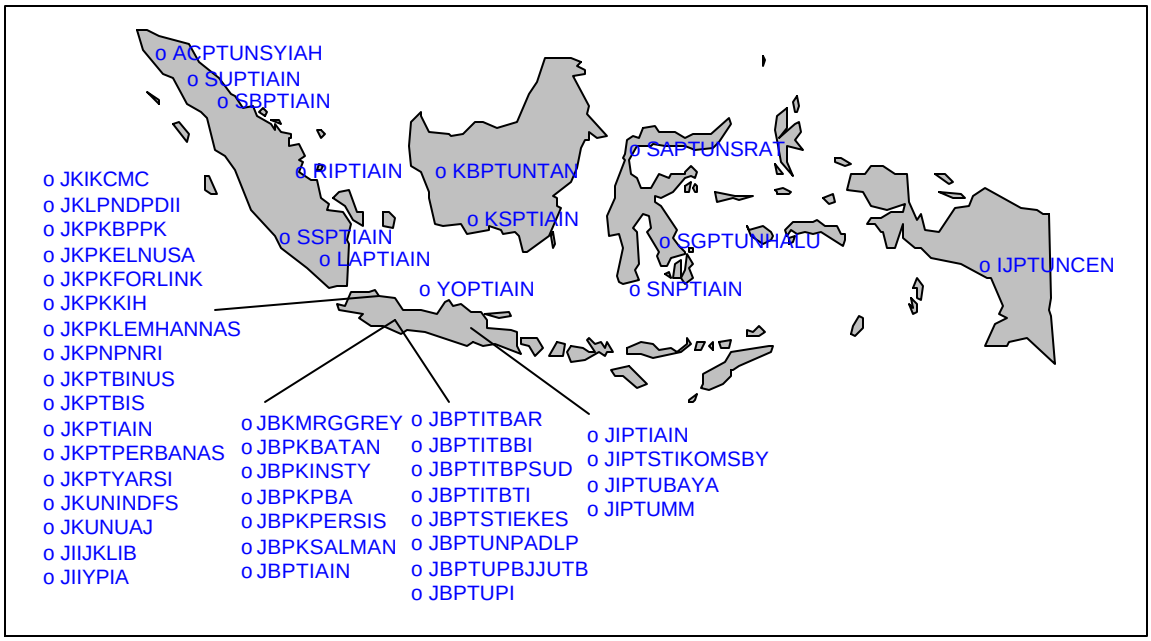

Figure 5. The Distribution map of the Indonesian Digital Library Network nodes

In addition to the implementation of OAI protocol at IndonesiaDLN, we define new service requests namely:

?? *Connect*,
?? *Disconnect*,
?? *PutRecord*,
?? *PutListRecords*,
?? *PutFileFragment*, and
?? *MergeFileFragments*.

These service requests are required, because requests and accesses only allowed to the registered institutions and users [4].

Repository registration is done at the hub or center Service Provider (http://gdlhub.indonesialdn.org), and each repository has its `PUBLISHER_ID` using national library code standard. The following is a sample of repository server configuration:

```
$DC_PUBLISHER_ID = "GDLHUB";
$DC_PUBLISHER_SERIALNO="IDLN-20010524-
024419-1";
$DC_PUBLISHER_TYPE = "INSTITUTION";
$DC_PUBLISHER_CONNECTION = "DEDICATED";
$DC_PUBLISHER_APPS = "GDL";
$DC_PUBLISHER = "GaneshaDL Central Hub";
$DC_PUBLISHER_ORGNAME="Knowledge Management
Research Group ITB";
$DC_PUBLISHER_HOSTNAME =
"gdlhub.indonesiaDLN.org";
$DC_PUBLISHER_IPADDRESS  = "167.205.23.27";
$DC_PUBLISHER_ADMIN =
"donfau@kmrg.lib.itb.ac.id";
$DC_PUBLISHER_CKO =
"cko@kmrg.lib.itb.ac.id";
$GDL_SERVER  = "gdlhub.indonesiaDLN.org";
```

Before sending any service requests, a Data Provider should be successfully connected to the central Service Provider using *Connect* service request. This request contains keyword argument containing `PUBLISHER_ID` and its serial number.

Information about the repositories above (Data Provider) except the serial number will be disseminated from the central Service Provider to the Data Providers for name resolution purposes. The repositories information request is sent by the Data Providers using *ListRecords* service request to the central Service Provider.

Users have to complete a free registration form in order to get full access to the repositories, including search, explore/browse, upload, and download. Registration can be done at any Data

Provider once. In order to they can use the same account to access repositories other than the one they do registration, the users information should be registered to the central Service Provider by the Data Providers using *PutRecord* or *PutListRecords* service requests. We also use the service requests to post local own metadata collections to the central Service Provider.

When user log into other repositories, a user authentication service request will be sent to the central Service Provider and a response will be returned whether information entered by user is true or false. This request is made using *GetRecord* service request.

Because some Data Providers are connected to internet temporarily (for example using dial-up connection) or even located behind proxy (e.g. in internet café), we need to store the files from such temporary-available Data Providers to the central Service Provider. This will ensure users to able to get the files referred by the metadata.

The file transfer can be done using off-line method such as using CD-ROM, or on-line method using two defined service requests, *PutFileFragment* and *MergeFileFragments*. *PutFileFragment* uses HTTP POST method and is used to upload a part of file that has been sliced into several small size fragments (e.g. 10 KB each). This service of course also support upload resume mechanism. After all fragments of a file have been uploaded, a *MergeFileFragments* service request will be sent to central Service Provider to merge the fragments into one file as originally stored at the temporary-available Data Provider. Using this service request, a dial-up connected Data Provider will be able to send as big as file size to the central Service Provider.

A *Disconnect* service request will close connection to the central Service Provider.

Figure 6 bellow shows all of the service requests between Data Provider and Service Provider in IndonesiaDLN.

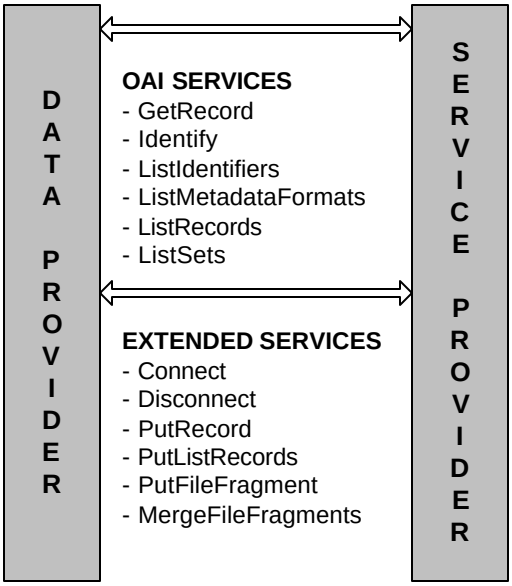

Figure 6. IndonesiaDLN service requests

At the present time, IndonesiaDLN has implemented successfully all of the service request functionalities in the Ganesha Digital Library software (GDL, http://gdl.itb.ac.id) that is distributed as open-source software [7], but hasn't used the OAI protocol standard. We are developing the next version of GDL to be fully compliant with the OAI protocol.

To reduce the complexity and frequency of cross-archival query and to lower the barrier of internet bandwidth at the Data Providers, we also put the functionality of the Service Provider to the Data Providers.

Using this scenario, every Data Providers can harvest or download the merged metadata records at the central Service Providers, and store them locally. Then, the Data Provider become a Service Provider for its local users for searching and browsing the whole or selected metadata from all Data Providers. Figure 7 bellow shows this scenario.

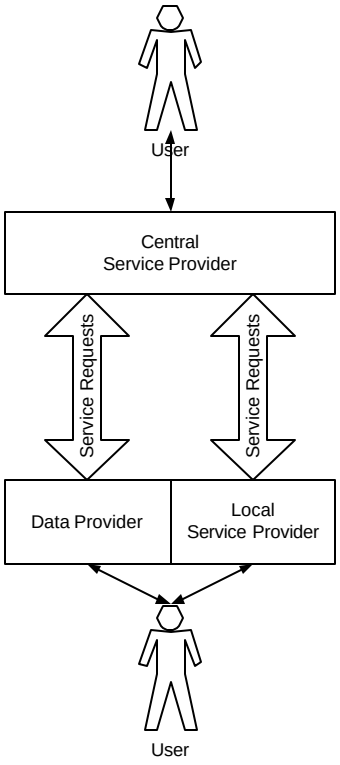

Figure 7. Scenario of interaction among user, Data Provider, and Service Provider

The following table shows current institutional members of IndonesiaDLN including their publisherid, name, connection status, and number of records stored at the central Service Provider [8].

Table 1. List of active digital library nodes in IndonesiaDLN

No	Node ID	#Metadata	Connection
1	JKPKELNUSA	1355	Dedicated
2	JBPTITBPP	1226	Dedicated
3	JKPKBPPK	348	Dedicated
4	SAPTUNSRAT	309	Dedicated
5	JIPTUMM	300	Dedicated
6	JBPTIPBMMA	118	Dedicated
7	JKPTBINUS	84	Dedicated
8	JBPEISMAIL	56	Dial-up
9	ACPTUNSYIAH	49	Dedicated
10	KSPTIAIN	46	Dial-up
11	SNPTIAIN	36	Dial-up
12	SSPTIAIN	35	Dial-up
13	SUPTIAIN	30	Dial-up
14	LAPTIAIN	28	Dial-up
15	IJPTUNCEN	26	Dial-up
16	JBKMRGGREY	24	Dedicated
17	GDLHUB	22	Dedicated
18	JKPKFORLINK	21	Dedicated
19	JKPKLEMHANNAS	19	Dedicated
20	JTPTIAIN	16	Dial-up
21	SBPTIAIN	15	Dial-up
22	KBPTUNTAN	11	Dial-up
23	JKPTIAINPP	9	Dial-up
24	JKPTPERBANAS	7	Dedicated
25	RIPTIAIN	7	Dial-up
26	JKPTYARSI	5	Dedicated
27	JKUNUAJ	4	Dedicated
28	SGPTUNHALU	3	Dial-up
29	JBPKINSTY	2	Dial-up
30	JKLPNDPDII	2	Dedicated
31	JBPTIAIN	2	Dial-up
32	JKPNPNRI	1	Dial-up
33	JKPEONNO	1	Dial-up
34	YOPTIAIN	1	Dial-up

From the 34 institutions above, 18 of them are using temporary internet connections, that usually through dial-up connections. The actual number of the institutional members could be more than such amount, because we believe there are some institutions that have been successfully installed and connected to the central Service Provider, but the administrators didn't send their records. Because they use temporary connection, we can't harvest their records. The only way is to suggest the administrators to upload the records periodically.

8. Conclusion

The key factors of the successfulness of the OAI protocol are laid on the selection of HTTP as the transport protocol, and the definition of the simple requests and responses mechanism using HTTP requests method and XML formatted responses. Other key factor is the selection of harvesting or materialization method over mediated, virtual, or federated method. In

Indonesia as a sample of the third world, to have a digital library server that integrated to the national digital library network, an institution doesn't have to have a dedicated internet connection. Institutions, individuals, or internet cafés using temporary connection (dial-up) can join IndonesiaDLN through GDL software. The similarity of current protocol used by IndonesiaDLN with the OAI protocol is lead to the possibility of migrating to the OAI protocol toward integration with the worldwide Open Archives Initiatives.

9. Bibliography

[1] Adam, N. R., Atluri, V., and Adiwijaya, I., 2000. SI in Digital Libraries, Communications of the ACM, vol. 43, no. 6, pages 64-72.

[2] Bell, D., and Grimson, J., 1994. Distributed Database Systems. International Computer Science Series, Addison Wesley, Wokingham, England.

[3] Elmagarmid, A., Rusinkiewicz, M., and Sheth, A., 1999. Management of Heterogeneous and Autonomous Database Systems. Morgan Kaufman, Los Altos, CA.

[4] Fahmi, Ismail, 2001. Mirroring Unified Metadata as the Solution for the Digital Library Network at Developing Countries. Proceeding 3rd IIWAS Conference, Lintz, Austria.

[5] Fahmi, Ismail, 2002. IndonesiaDLN-OAI Protocol. http://idln.lib.itb.ac.id/Open.html?target=tech/indonesiadln_oai.html

[6] Fahmi, Ismail, 2002. Perkembangan dan Tantangan Masa Depan The IndonesiaDLN. The 3rd IndonesiaDLN Meeting, Bandung, March 2002.

[7] Fahmi, Ismail, KMRG, 2001. Ganesha Digital Library web site. http://gdl.itb.ac.id.

[8] IndonesiaDLN, 2001. The Central Service Provider of GDL-Network. http://gdlhub.indonesiadln.org.

[9] Khalil-Ibrahim I., 2001. Semantic Query Transformation for the Intelligent Integration of Information Sources. Ph.D. thesis. Gadgah Mada University, Indonesia.

[10] Khalil-Ibrahim, Ismail and Wieland Schwinger. 2001. Data Integration in Digital Libraries: Approaches and Challenges. Proceeding IndonesiaDLN Seminar, Bandung, Indonesia.

[11] Levy , A. Y., 2000. Logic? Based Techniques in Data Integration. In Minker, J., Logic Based Artificial Intelligence, Kluwer Publishers.

[12] Litwin, W., and Abdellatif, A., 1986. Multidatabase Interoperability. IEEE Computer, vol. 19, no. 12, pages10?18.

[13] Navathe, S. B., Elmasri, R., and Larson, J., 1986. Integrating User Views in Database Design. IEEE Computer, pages 50?62.

[14] Reddy, M. P., Prasad, B. E., Reddy, P.G., and Gupta, A., 1994. A Methodology for Integration of Heterogeneous Databases. IEEE Transactions on Knowledge and Data Engineering, vol. 6, no. 6, pages920?933.

[15] Silberschatz, A., Stonebraker, M., and Ullman, J. D., 1991. Database Systems: Achievements and Opportunities. SIGMOD RECORD, vol. 19, no. 4, pages 6?22.

[16] Suleman, Hussein and Edward Fox. The Open Archives Initiatives: Realizing Simple and Effective Digital Library Interoperability. http://www.dlib.vt.edu/projects/OAI/reports/jla_2001_article_oai.pdf

[17] Van de Sompel, Herbert, Carl Lagoze, 2001. The Open Archives Initiatives Protocol for Metadata

Harvesting, http://www.openarchives.org/OAI_protocol/openarchivesprotocol.html

[18] Widom, J., 1995. Research Problems in Data Warehousing. Proceedings of the 4th Conference on Information and Knowledge Management (CIKM).

[19] Wiederhold, G., 1993. Intelligent Integration of Information. Proceedings of ACM SIGMOD, vol. 22, no. 2, pages 434?437.

[20] Zisman, A., 1998. Information Discovery for Interoperable Autonomous Database Systems. Ph.D. thesis. Department of Computing, Imperial College of Science, Technology, and Medicine, University of London.