RESOURCE AGGREGATION IN DIGITAL LIBRARIES
Static vs Dynamic Protocols

Pedro Almeida, Marco Fernandes
IEETA, University of Aveiro, Aveiro, Portugal

Joaquim Arnaldo Martins, Joaquim Sousa Pinto
IEETA/DETI, University of Aveiro, Aveiro, Portugal

Keywords: Digital Libraries, Metadata Aggregation, Distributed Systems.

Abstract: This article analyses development of static and dynamic protocols to aggregate metadata and resources from heterogeneous systems. In particular, it compares the advantages and drawbacks of both types of protocols and presents a case study of the University of Aveiro Information System as an example of the possibilities of dynamic resource aggregation systems.

1 INTRODUCTION

Resource harvesting has been a major issue in Digital Libraries systems.

Gathering information from distinct sources with different metadata schemas lead to the development of distributed search mechanisms and resource harvesting protocols. The main concept being explored behind these approaches is the development of interoperable and efficient federated search engines, which can access, search and retrieve metadata from different Digital Libraries catalogues.

As the content of Digital Libraries evolves to heterogeneous collections, composed by several types of documents, new standards are being developed and used to properly classify the information contained in Digital Libraries. While some years ago we could think only in text based searches, the reality nowadays is that new formats of queries are being developed, and the future and success of huge repositories depends on the ability to process text, image or sound based queries.

In order to provide federated search facilities over metadata from different repositories, two different strategies can be adopted: a static approach - harvest the content of the repositories, index the metadata and provide search mechanisms over it, or a real-time dynamic approach - provide programmatic interfaces to search and retrieve the metadata on the fly.

Each one of these approaches has advantages and drawbacks. In this article we analyze the implications in using either a static or a real-time dynamic aggregation protocol. In the end, we will present a case study of a dynamic integration system that was developed for the University of Aveiro's Institutional Repository (IR), a Digital Library that contains several documents of different formats (text, images, videos and sounds).

2 STATIC AGGREGATION AND DYNAMIC AGGREGATION

The aggregation of Digital Libraries resources is concerned with two important issues of Information System technologies: provide mechanisms to simultaneously search several repositories of metadata; and provide the technology to access records of different Digital Libraries or systems, ignoring the operating systems and details of implementation of the information systems that support them.

From a Computer Science point of view, resources can be aggregated either with static or dynamic methodologies. We consider static aggregation protocols as methodologies that gather content from

Almeida P., Fernandes M., Arnaldo Martins J. and Sousa Pinto J. (2008).
RESOURCE AGGREGATION IN DIGITAL LIBRARIES - Static vs Dynamic Protocols.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - DISI*, pages 405-412
DOI: 10.5220/0001675604050412

different Digital Libraries or information repositories, format it into a single and uniform metadata language, and provide search facilities over collected metadata/information. On the other hand, we consider dynamic aggregation protocols as methodologies that gather content from different Digital Libraries or information repositories in real-time. In the last methodology, the collected metadata/information is obtained at the moment the query is submitted to the search interfaces provided by the systems that contain the information.

It is important to refer the role of service providers and data providers according to each of these methodologies. In the static model approach, the data provider's only responsibility is to provide metadata. The service provider must collect the metadata, index it and provide search facilities.

In the dynamic model approach, data providers must implement extra services, namely search facilities over the metadata they contain. The role of the service provider is less demanding, as, according to this model, it does not need to store the metadata, or index and provide search facilities over the indexed content. Service providers act as pure information aggregators.

Each of these aggregation methods has advantages and drawbacks, but these concepts will be discussed in more detail further ahead. At this stage, we intend to clarify some of the concepts associated with each methodology and, in the next section, we will present some related work.

3 RELATED WORK

During the last years, several researchers from all over the world have been studying the problematic of aggregating information from different repositories. The problems that arise are related with the usage of different metadata to describe multi-format digital object content and the different technologies used to implement the information systems that store and retrieve metadata from repositories.

3.1 Z39.50

One of the first available protocols to search simultaneously different databases is the Z39.50, an American national standard for information retrieval. It is formally known as ANSI NSO Z39.50-1995 – Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (National Information Standards Organization, 2003). The main purpose of this standard is to define a communication protocol to access and query databases stored in different computers with different software, facilitating the process of interconnecting computer systems.

The standard specifies the formats and procedures involved in the exchange of messages between a client and server, enabling a Z39.50 client to request the server to search a database, identify records which meet specified criteria, and to retrieve some or all of the identified records.

Z39.50 protocol also defines different record syntaxes (Library of Congress, 2005), being most of them variants of MARC records (Library of Congress, 2007a) and resource format types, such as mime-types and other file formats.

The Z39.50 Information retrieval protocol is composed by a group of facilities to access database information, namely: Initialization, Search, Retrieval, Result-set-Delete, Access Control, Accounting / Resource Control, Sort, Browse, Explain and Termination. These operations specify the interaction between the Z39.50 client and the Z39.50 server, defining the services and functions that can be invoked by the client.

In order to obtain the facilities that a Z39-50 server supports, the Z39.50 clients can invoke the Explain facility, and the server will answer with details of the implementation, a list of databases available for searching and the schema, record syntax and element specification definitions supported for record content retrieval.

Another operation defined in the Z39.50 is the Browse facility. It is composed by a single service, Scan. The Scan service is used to scan database content, as long as the client provides an ordered term list to scan (subject, names, titles, etc.), a starting term and a number of entries to be returned.

Using these two facilities, Ray R. Larson developed a Cross-Domain Information Server, using Z39.50 as the protocol to implement Distributed Resource Discovery (Larson, 2001). As stated in the article, the author used the Z39.50 functionality Explain Database to determine the databases and indexes of a given server. Then, using the SCAN facility, the author extracted the contents of the indexes and used that information to build “collection documents”. The records were retrieved using probabilistic retrieval algorithms. Z39.50 also defines a network protocol to transfer information between the client and server. Usually, the port number 210 is defined as the default port for Z39.50 message transfers. Using port 210 in modern Information Systems causes a problem in large networks. As network security is always an important factor to guaranty in corporation networks, accessing a Z39.50 server on the Internet imposes a reconfiguration of the firewall in order to open port 210 to communications. Sometimes this might be hard to negotiate with the network administrator.

Other drawback associated to the network protocol defined in Z39.50 is the fact that the messages are transferred using rigid syntaxes. XML would facilitate the interpretation of these messages by third party tools.

On its essence, Z39.50 is a dynamic aggregation system, since using this protocol Z39.50 clients can search different databases and integrate the results from distinct data sources. But in the case of the system develop by Ray R. Larson, Z39.50 is used just to collect metadata and build a database with information from different providers. This way, we consider that this system is a static aggregation system, since in order to retrieve information from different repositories it is necessary to first collect data from the associated databases.

3.2 OAI-PMH

Another research group that studies the issue of digital library metadata aggregation, created one important initiative that has been gathering a significant amount of supporters from the Library and Archive community: the Open Archives Initiative (OAI) (Open Archives Initiative, 2007).

This research group has defined a protocol to gather information from different data sources, the Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH) (Open Archives Initiative, 2002).

This protocol specifies a standard mode to harvest and retrieve records from repositories that implement OAI-PMH.

OAI-PMH specifies two types of participants: the data providers and the service providers. Data providers are digital repositories that expose the metadata about their objects with specific methods defined in the OAI-PMH protocol. Metadata is then gathered by harvesters, or aggregators. In the OAI-PMH definition, these agents that gather the information are defined as service providers.

Some important difference between OAI-PMH and Z39.50 are the network protocol used for message transfer and the syntax used for message transfer: OAI-PMH uses HTTP, port 80, as a network protocol, and uses XML to transfer messages between the client and server. As port 80 is used by Web Browser to access the Internet, usually this port is always configured in the firewalls as an open port, eliminating the need to open special ports in the firewall.

By using XML, OAI-PMH messages can be easily interpreted and processed by third party tools.

Data providers expose their metadata as sets: a collection of metadata available for harvesting. They are responsible to maintain a service available, usually as a web server, which supports the OAI-PMH protocol as a means of exposing metadata from repositories. At least, data providers must be able to expose metadata expressed in the Dublin Core (DCMI, 2007) format, but more complex metadata may also be disseminated.

The Dublin Core metadata is composed by fifteen elements which are: contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title and type. Not all the elements are mandatory in the metadata that is sent to the service provider, but it is important that data providers expose all the metadata that is available for each record.

Service providers gather metadata from a group of data providers and provide search mechanisms over the information gathered. To collect data from different repositories, the service providers issue OAI-PMH requests to data providers.

Data providers answer to these requests with the metadata collection they contain. Service providers use the metadata to fill an inter-repository database that stores information from all the data providers that are associated with him. This collection of metadata from different repositories is used as a basis for building value-added services, namely providing search facilities over large collections of metadata from different repositories.

An example of a service provider is OAIster (University of Michigan, 2007), a search engine developed by the University of Michigan Digital Library Production Service.

Their goal is to create a collection of freely available, academically-oriented digital resources that are easily searchable by anyone.

OAI-PMH is also an example of a static aggregation system for Digital Libraries. In order to search the content of several repositories simultaneously, service providers must first harvest the repositories content. In small repositories this could be a simple task, but when we consider huge repositories with millions of records, harvesting such a database is necessarily a time consuming task.

3.3 SRU

Library of Congress serves as the maintenance agency for one of the new standards that perform search queries on Internet Databases: Search/Retrieve via URL (SRU) (Library of Congress, 2007b).

SRU acts as a search protocol for Internet search queries, and uses a Common Query Language (CQL) (Library of Congress, 2007c) to define query syntax. It is composed by a SRU Request, and a SRU Response. A SRU Request is a URI as described in RFC 3986 (Berners-Lee, 2005). A SRU is composed by a base URL and a search part, separated by a question mark ("?"). One example of a SRU Request would be the following URL: http://z3950.loc.gov:7090/voyager?version=1.1&operation=searchRetrieve&query=dinosaur. As it can be easily understood, the expected result of this query would be resources that contain the term "dinosaur" in the associated metadata.

All records retrieved by an SRU Request are transferred in XML. It is not necessary that the records themselves are stored in XML, but they must be transformed to XML before the transfer from the server to the client. Another important characteristic of SRU is that records in the response may be expressed as a single string, or as embedded XML.

This protocol an example of dynamic aggregation since the search results are obtained on real-time. As we will show further ahead, the SRU protocol is similar to the methodology that we have adopted and that we will present in this chapter. The problem with SRU is mainly associated with the limitations imposed by specification, and the lack of a mechanism to retrieve the records themselves. Using SRU it is only possible to retrieve metadata associated to a record, not the information that was classified and indexed.

4 STATIC AGGREGATION VS. DYNAMIC AGGREGATION

In this section we pretend to identify the implications in using either static or dynamic aggregation. Both of the methodologies have advantages and drawbacks, and our intention is to provide a list of features that will be easy (or difficult) to achieve depending on the adopted methodology.

Drawbacks of static aggregation:

1. Time to harvest the entire repository content.
2. Searches are made in possibly outdated metadata.
3. Difficult integration with other software tools.
4. Harvested data must be stored in the server that provides the search mechanisms.

Advantages of static integration:

1. Offline repositories can be queried.
2. Possibility to detect and eliminate duplicate records in different repositories.
3. Search performance depends only of the server.

Searching simultaneously several repositories from a static set of information can originate some problems. Retrieving the right information from large collections of metadata is not an easy task. For example, in this article (Hochstenbach, 2003) authors defined some future guidelines for OAIster, in order to improve the quality of information retrieval. These guidelines include: searching within institutions; browsing capability; eliminate duplicate records (records that are the same among repositories). These limitations belong to a specific system, and this is the reason why they were not considered in the list of drawbacks.

The time required to harvest data from a repository might also be a problem. It is common to think in a Digital Library as a repository that stores millions of records. Harvesting all these records might be a time consuming task.

In order to obtain updated results it is necessary to collect new data from repositories in a regular basis. If dealing with three or four distinct repositories is a perfect acceptable number of repositories to keep track of, considering the extensibility of these systems, and the possibility to interact with a higher number of information repositories, this type of approach might lead to another time consuming task.

Regarding to the integration of harvesters by other information systems we could not identify other advantages in addition to reuse harvested information. Since the information has to be harvested and stored in a centralized mode, querying directly data providers should be a better option than integrate the search results in another software tool. Another problem that does not encourage the integration of harvesters is the lack of control in the harvesting process by the system integrator. It is difficult to determine if the results are updated or synchronized with the original repository.

Yet another issue in static aggregation is the need to store all the metadata collected in the harvester. Since the data is already present in the original repository, harvesting the metadata represents creating redundant copies of the same information, and as a consequence, the harvester will necessarily need more resources, namely disk space, to store metadata copies.

Despite the drawbacks already identified, static aggregation has some advantages over real-time dynamic aggregation. One advantage is that online repositories can be queried, even if the service providers, harvester, has no connectivity with the data provider. Since all the metadata is stored in the server, queries may retrieve result sets from o²ine repositories. Yet, it is necessary to be aware that this might lead to another problem: the harvester might provide the link but the document might be inaccessible.

Identifying duplicate records is, most of the times, a time consuming task. Since a harvester usually takes some time to gather information from a repository, it is acceptable to spend small amounts of time to find duplicate records. As a benefit, a user could get results from different repositories ignoring duplicate results.

Regarding to the drawbacks and advantages of dynamic integration we have identified the following items:

Drawbacks of dynamic aggregation:

1. Offline repositories won't be able to show results.
2. Difficult to eliminate duplicate records.

Advantages of dynamic integration:

1. Guaranty of updated results.
2. Direct integration in software tools.
3. Web Services and XML oriented, providing high interoperability and low time cost in application development.

The drawbacks associated with dynamic aggregation are related with connectivity and processing time issues. The first problem in dynamic aggregation is that if a metadata provider has no connectivity with the system that queries the repository, the aggregator system will not be able to obtain search results. The second problem is related with eliminating duplicate records. In order to detect duplicate records it is necessary to spend some processing time. This might constitute a problem because it would mean spend extra time to obtain the results. In our opinion, finding duplicate records in an aggregator system is not a critical problem, but despite our point of view, it still is a limitation in real-time aggregation systems.

One important advantage is the guaranty that the obtained results are always updated. As in dynamic aggregation data providers implement search interfaces, service providers always have access to the most actual information. This is an important issue in repositories that have high rates of document insertion, like the case of the repository of the University of Aveiro, where every day new items are being added or updated in the different collections.

Regarding to advantages 2 and 3, they are only possible if the dynamic aggregation system is implemented using Web Services and XML. These two technologies provide important an important feature to the information systems: interoperability. It also allows different computers, with different operating systems and software, to exchange data via a common set of business procedures. Integrating a search interface of a data provider that exposes the search functions with Web Services is a very rapid, simple and efficient mode to put two (or more) applications communicating with each other.

5 CASE STUDY

The case study that we present is based on the information systems strategy adopted by the University of Aveiro. The policy being implemented inside the institution promotes the development of software and information systems that expose data, metadata and search interfaces via Web Services. The actual philosophy is to maintain the information centralized in specialized service providers, but accessible to the rest of the information systems of the university, avoiding data replication between the different databases that exist over the University.

Three systems compose the actual Digital Library: SInBAD, Curriculum and eABC. Each of the systems contains specialized information that, by itself, is very useful for the university. But the increased value obtained with the integration of these three systems is one interesting example of the potential of growth of functionalities and services just by using dynamic aggregation.

5.1 SInBAD

SInBAD (Almeida, 2006) is an integrated system for digital libraries and archives. This project is being developed at the University of Aveiro and its major goal is to collect and store information about institutional records.

At the moment, this system contains a collection of posters, a photographic archive, an audiovisual archive, a thesis collection and a vast collection of bibliographic information of jazz books, magazines and records.

One interesting feature of SInBAD is that it contains four subsystems, which can be accessed individually or simultaneously. It implements a real time dynamic aggregation mechanism internally, providing a search portal that can simultaneously access different types of information.

5.2 eABC: Bibliographic Archive for Scientific Production

eABC (Santos, 2005) is an information system that contains bibliographic records of the scientific publications of the University of Aveiro. Initially, the main purpose of the system was to provide a tool where researchers could manage their scientific publications. Later, the Research Institute began to use this system to manage the research production of the University of Aveiro.

Every year, the Research Institute produces a report with all the publications of the research units that belong to the University and sends it to the Portuguese Foundation of Science and Technology. This system provided an easy mode to accomplish the difficult task of gathering publication information of all the researchers inside the university.

5.3 Curriculum

Curriculum (Teixeira, 2005) is an information system that stores the curriculum vitae (CV) of researchers, professors, students and other elements of the University of Aveiro. This system has one important objective as an element of the University of Aveiro information sources: maintain an updated repository with the researcher's CVs. Despite the simple goal of this system, the quality and coverage of the services that it can provide make of it one of the key information systems inside the university. Over the last two years, the candidates of the PhD scholarship applications of the University used Curriculum to store their CV. Another example of system usage is that when researchers of the University need their CV to be attached to a project or career progress contest they can access this system and obtain an updated version of their CV.

5.4 Real-time Resources Aggregation

In order to build increased value services implemented with the existing systems, we have defined an interaction schema between SInBAD, eABC and Curriculum. Each one can perform its tasks without interacting with other external systems, but the interaction between the three systems allows users to access transparently a vast group of related information.

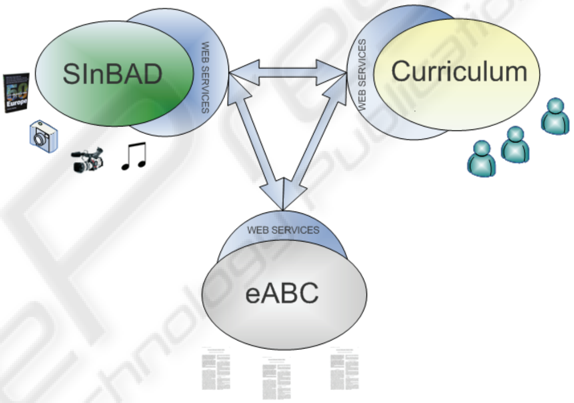

Figure 1: Dynamic aggregation model.

Figure 1 depicts the interaction diagram between the three systems.

SInBAD provides digital content objects, eABC provides bibliographic information and Curriculum provides author information. The aggregation of resources is possible due to the Web Services interfaces that all systems implement. Each of these interfaces has specific methods to insert, delete, search and retrieve data and metadata from any system.

As it becomes obvious, the increased value services come from the real-time dynamic integration of the data and metadata from the three information systems.

Consider, for example, a situation where a researcher inserts a new publication in eABC. Before the existence of SInBAD, the user could not upload a PDF file of his scientific work. The only option that was available was a bibliographic field where the user could register the URL of a digital version of the article. Now, eABC allows the user to select a PDF file with the work content and submit it along with the bibliographic information. Actually, the PDF file is not stored in eABC, but in SInBAD. Each of the systems maintains its primary goal, and there is no overlapping of functionalities. As an increased value, all the scientific publication will be stored in a centralized repository, avoiding the dispersion of publications for several user machines. Another interesting point is that, when SInBAD is queried for scientific publications, the search results are obtained from eABC. SInBAD just identifies articles that are stored in its database and presents the users a list of scientific publication that meet the required query parameters. As the query is actually resolved by eABC, it is guaranteed that the most actual metadata is always accessed. This eliminates the time consuming task of synchronizing metadata values from different information systems.

Based on the author ID stored in eABC, it is also possible to link to the CV of the researchers that produced the scientific publication. Once again, the complete CV is stored in two different systems, because the scientific publications associated to the researcher are stored in eABC. The Curriculum system just stores personal and professional information. This characteristic is very important in large research units. Sometimes the number of authors associated to a publication is considerably large, and easily one or two authors could forget to associate the scientific publication with his CV. Therefore, once the record is created in eABC, the association with the authors CV is automatic, which alleviates researchers from the tedious task of updating their CVs.

As all the services are interconnected and related, users accessing one of this systems can easily access a vast group of data that is related. It is also possible to build new services that dynamically import data from the three systems and present subgroups of information. As the systems began to operate, several web masters responsible for the web pages of the departments of the university began to request us information about the scientific publications associated to the department units, links to the researchers CVs, etc. For example, some departments in the university wanted their web pages to contain a list of the master and PhD thesis and scientific articles related with the department. Before the development of SInBAD and eABC, this list was written inside the department web page, or was retrieved from a small database. Now a complete list with this information is retrieved directly from SInBAD an eABC, allowing the web masters to integrate updated data in the department web pages.

6 CONCLUSIONS

We consider that the new XML dynamic aggregation based technologies that allow different systems to communicate in an integrated format might take an important role in metadata and resources aggregation. Web Services provide all the necessary means to build interoperable repositories that share interfaces to their search mechanisms. Instead of building system providers, according to the OAI-PMH definition, that must deal with the harvesting, search and aggregation of the information, a new generation of system providers might only deal with the aggregation of the results. Data providers assume higher responsibility in this new proposal, being responsible not only to manage the metadata, but also to provide query interfaces to system providers.

The new strategy for the information systems architecture of the University of Aveiro Digital Library is actually giving its first results. As the aggregation of different content from different repositories is becoming a rapid and warrantable process, all the web developers inside the university are exploring the new possibilities of accessing department specific information.

Resource aggregation is obviously an important step to increase the potential of Digital Libraries and Institutional repositories.

REFERENCES

National Information Standards Organization, 2003. ANSI/NISO z39.50-2003: Information retrieval (z39.50): Application service definition and protocol specification, NISO Press.

Library of Congress, Network Development & MARC Standards Office, 2005. Registry of Z39.50 object identifiers. Last update: October 2005, Available at www.loc.gov/z3950/agency/defns/oids.html. Accessed in November 2007

Library of Congress, Network Development & MARC Standards Office, 2007a. Marc standards. Last update: July 2007, Available at http://www.loc.gov/marc/. Accessed in November 2007

Larson, R. Ray, 2001, Distributed resource discovery: Using Z39.50 to build cross-domain information servers, *In JCDL 01*. ACM Press.

Open Archives Initiative, 2007. Available at http://www.openarchives.org, Accessed in November 2007

Open Archives Initiative, 2002. The Open Archives Initiative Protocol for Metadata Harvesting, Last update: October 2004, Available at http://www.openarchives.org/OAI/openarchivesprotocol.html, Accessed in November 2007

DCMI, 2007. The Dublin Core Metadata Initiative, Last update: November 2007, Available at http://dublincore.org/, Accessed in November 2007

University of Michigan, 2007. OAIster, Available at http://www.oaister.org/, Accessed in November 2007

Library of Congress, 2007b. SRU: Search/Retrieve via Url, Last update: September 2007, Available at http://www.loc.gov/standards/sru/, Accessed in November 2007

Library of Congress, 2007c. CQL: Contextual Query Language (SRU Version 1.2 Specifications), Last update: August 2007, Available at http://www.loc.gov/standards/sru/specs/cql.html, Accessed in November 2007

Berners-Lee, T., 2005. Uniform Resource Identifier (URI): Generic syntax, http://www.ietf.org/rfc/rfc3986.txt.

Hochstenbach, P., Jerez, H., Sompel, H., 2003. The OAI-PMH static repository and static repository gateway, *In Proceedings of the Joint Conference on Digital Libraries (JCDL03),* IEEE Press.

Almeida, P., Fernandes, M., Alho, M., Martins, J., Pinto, J., 2006. SInBAD - A digital library to aggregate multimedia documents, *In AICT-ICIW '06*, IEEE Press.

Santos, J., Teixeira, C., Pinto, J., 2005. eABC - Um Repositório Institucional Virtual*, In XATA 2005 : XML Applications and Associated Technologies*, pp. 40-51, University of Minho Press.

Teixeira, C., Pinto, J., Santos, J., 2005. Curriculum@ua - Online XML based personal curriculum, *In XATA 2005 : XML Applications and Associated Technologies,* University of Minho Press.