Framework for a Federated Digital Library

Hardik Joshi Manoj Vyas

Abstract

The Integration of bibliographical data today is considered one of the most important tasks in the area of Digital Library. Various available sources of bibliographical info vary widely in terms of data representation. In this paper we have tried to propose a framework for the development of the middleware based digital library using the XML.

Keywords : Federated Digital Libraries, Middleware, XML

0. Introduction

Digital Library have gained acceptance in many scientific and technical disciplines. However most of the Digital Library are implemented in systems and protocol specific to the discipline they support. Interoperability between Digital Library has yet to be achieved on a large scale. The challenges to interoperability are :

a) Integration should be flexible enough to allow individual Digital Library to add/modify features at the same time give the user an impression of a single library.

b) Relocation of individual, Digital Library should be transparent to the users.

A federated library is an integration of digital libraries, which may consist different database systems and file systems. Federation can be achieved in 3 ways:

1) Modifying the existing Digital Library to interoperate.

2) Extracting metadata from each Digital Library and indexing it as a separate Digital Library

3) Treating each Digital Library as a separate entity and performing distributed searches.

The first approach to interoperability requires digital libraries to use some kind of protocol or Digital Library software suite. The second method has certain advantages but it assumes that metadata can be extracted and re indexed with no technical or legal barriers. The third method creates more work for the provider of the federated Digital Library but allows for the inclusion of a greater no of Digital Library.

In this paper, a framework is provided that facilitates the development of web based digital libraries. It is based on the XML standards. The framework is presented through digital library architecture of University Data. One of the goals of this work is to identify the benefits that derive from the employment of XML to the field of digital libraries. More specifically, XML allows the Web client to present different views of the same data to different users since it separates content from presentation. Interoperability between different components of a digital library can be easily established since XML defines a common format for computer-to-computer interaction. In the implementation of the architecture, XML data structures that are both human readable and computer understandable reduce the complexity.

1. Related Work

The Networked digital library of Theses and Dissertation (NDLTD) [4] in the US is a inter-University digital library of Theses based on SGML-XML and Z39.50 standards. In this work, an argument in favor of XML is presented, claiming that it offers a great deal of flexibility compared to other alternatives such as storing document descriptions in a relational database.

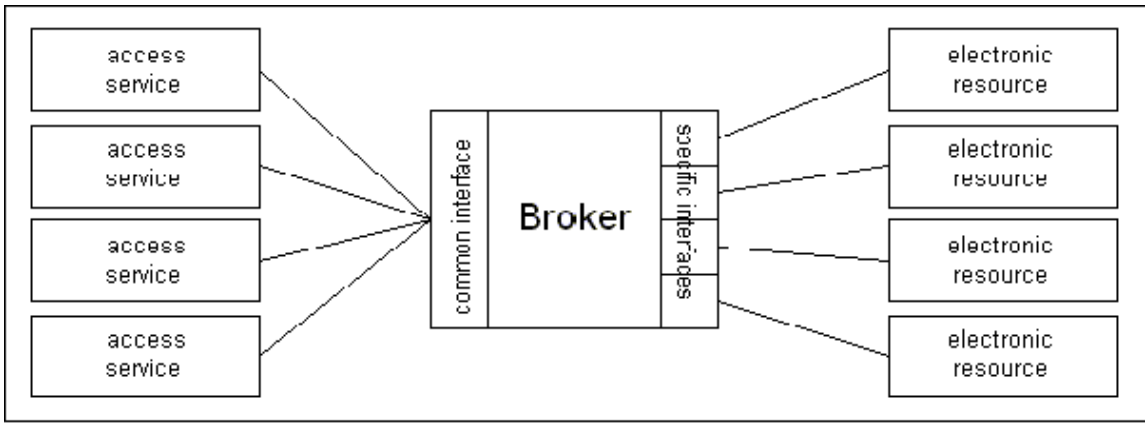
THETIS is a web-based distributed environment consisting of one or more underlying repositories. Each server node contains a search engine and a retrieval engine module. In this architecture, distributed queries against repositories whose objects are described by different metadata sets, are supported in the intersection of the sets.

2. Prerequisites

The proposed framework assumes that the digital libraries, which are supposed to be, integrate, deliver the web-based data. These libraries have some form of back-end in the form of relational database or which can be converted into the XML form. Client browsers support XML parsing. Java is best suited for the web based programming with XML, hence the middleware is implemented using Java technologies.

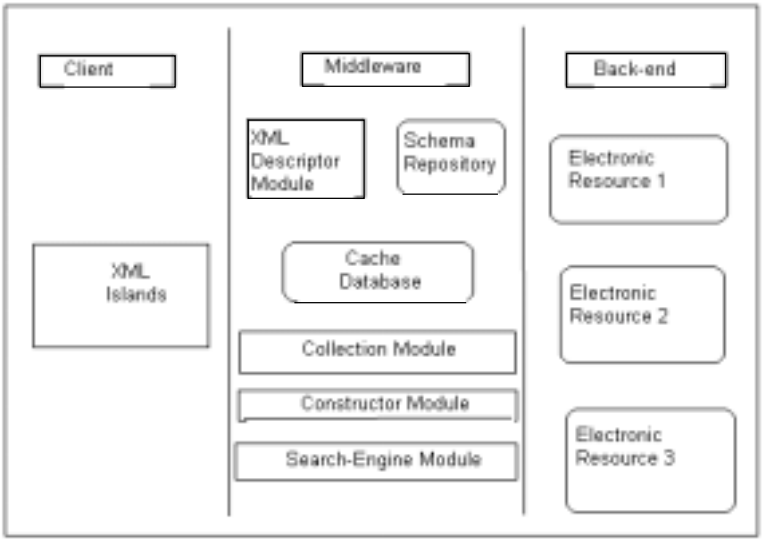
3. Proposed framework of Middleware for Core Services

It provides uniform interfaces to the distributed components of a Digital Library. It helps tie together the storage, delivery, searching, browsing of electronic resources. By wrapping the Digital Library core services inside a middleware, existing and new resources can be more easily integrated into the Digital Library. This scalability is achieved through a middleware architecture that “brokers” communication between components. Components providing access to the users who might be authorized for various resources can communicate via one interface with a broker that handles the complexity of multiple resource interfaces when a resource is added or changed, only the broker is affected rather than every component. Hence by changing the middleware slightly a new Digital Library can be easily integrated into the system of federated digital library.



The Broker Model for Integrating Federated Digital Resources

The proposed framework supports web-based, three-tier architectures based on XML. It is going to be presented through deriving digital library architecture of University data. The conceptual design of the digital library is based on the organizational structure of a University. Thus, the back-end consists of number resources that correspond to the departments within the University. Each department is planned to maintain a repository with department-specific documents. The database may or may not be under a unified schema. But the user can view a homogenous and highly organized repository structure. This is due to the fact that XML facilitates management of documents with diverse structure/content.



Schematic Representation of various modules of the proposed framework

The middleware consists of 4 modules that support the core functionality of the digital library. The modules are:

1) The XML Descriptor module
2) The Collection module
3) The Constructor module
4) The Search-Engine module.

The XML descriptor module contains the various schemas that define the structure of each document in the database. According the XML specification, each document has certain rules(constraints) referring to its structure and content. The Collection module accepts documents form individual repositories. The Constructor module merges the results into a unified result-list document according to the broker's schema. This module also interacts with the end-users and provides the functionality similar to other search engines. The Search-Engine module is responsible for locating documents that satisfy users requests. It is an XML oriented program that accepts queries and return results in the form of XML trees.

A cache database is also maintained to improve the overall performance of the digital library. Another component that resides I the middleware is the schema repository. It stores the necessary schemas that are used from the authors as templates and definition files for their documents

The client layer is XML capable web browser, It is supposed to handle the XML trees and present a formatted output to the end-users.

3.1 The XML Descriptor module

The XML descriptor module facilitates Resource Discovery for the digital library users. The structure of XML descriptor is defined in its referring schema, which is named XML_Descriptor.dtd. This schema migrates to all nodes of the digital library in the form of a file name XML_Descriptor.XML as shown in the below figure.

The elements of the XML descriptor module are actually metadata. The benefit of metadata is they make large portions of information available both for human and computer understanding.

3.2 The Collection module

On initialization of interaction with a user, the aforementioned XML Descriptor is invoked at each repository and is requested to transmit its corresponding XML-tree of the Collection module. The Pull Model is applied for querying the database because it guaranties the consistency of the underlying document collection. If some nodes of the digital library are unavailable at the time of a transaction, they simple won't be fetched.

After the initialization, the user is requested to decide how to classify the collection. There are 3 ways available (Supported by the XML Descriptor module) :

Through a list of lectures being taught, through a list of the individual departments and through a list of the predefined thematic areas. By selecting one or more individual items of a classification, the user actually narrows the search to the resources that are associated with the above departments. This way the searching process is faster and with more accurate results.

3.3 The Constructor module

This module is responsible to merge the results into a unified schema and cache the XML database. It maps each result query from the repository into a single XML-tree according to the XML data specification. This is the key module for scalability. Simply by modifying this module we can scale the digital library.

The programming language that is potentially best suited to manage XML data and consequently implement the functionality that will be provided by the various modules of the digital library is Java. Its platform-independent nature as well as its specialization t web applications, renders Java as the most appropriate language for such a task.

3.4 The Search Engine module

After focusing in a certain subset of the entire document collection, a user submits his/hers search criteria to the Search Engine module. The retrieved search criteria will be translated to a query that is formatted according to the XML query specification. The resulting query will be addressed to every local resource that has been selected to participate in the search during the previously described first stage of searching/browsing process. The results are again sent to the Constructor module where there are merged into a single XML tree. There is no need t expand the searching process beyond the XML files since every file is associated directly or indirectly with an XML file.

3.5 Back end structure

The Back end may be any database or distributed databases. The queries to the back end are in returned in the form of record sets, which are converted into XML. This includes mapping between various database models to XML. The unification of each individual results is handled by the constructor module.

A web server is maintained at every resource node of the digital library. It registers the node to the middleware through the previously described XML Descriptor module. Considering the digital library as three broad classes of elements (data, metadata and processes), each instance of every class should be referenced and identified consistently, through permanent names and identifiers.

Conflicts between the component names (element names, attribute names, user-defined entities etc) of the various XML files are prohibited through the employment of the XML namespaces. Each XML file is associated with a namespace that includes the definition of the various XML component names that it discusses.

The proposed framework supports various formats. Nevertheless, the fact that every file in the back-end is associated with the XML file, dictates that the modules residing at the middleware need to know the location of only these XML files. A common structure should therefore be followed at every node of the digital library for proper and transparent information retrieval.

3.6 Document structure

Each document in the collection may have multiple files of various formats that constitute it. Since these files are created in formats that don't describe their content, an additional meta-file accompanies each such file. Its elements are essentially metadata that provide a description to the content of the other file they refer to. Using Xpath [5] technology, the various sub-components of the document are referenced from links that belong to a single file that is name "COMPOSITE.XML". Starting from the COMPOSITE.XML, the user is able to navigate through the rest of the associated files and annotate, evaluate and make comments on them without having to modify the original read-only files.

4. Conclusion

We presented our approach for the integration of digital libraries into a single federated digital library. We have proposed a framework of 3-tier architecture where Clients are supposed to have XML based browsers (Client can interpret XML documents). The middleware framework consists of various modules which involve inter communication using XML. The back-end is assumed to have various resources in the form of databases or distributed databases. We are trying to prototype this model for our college libraries.

5. References

1. http://www.w3c.org
2. http://www.drtc.isibang.ac.in
3. http://www.ndltd.org
4. http://www.dlib.org
5. XML Path Language (Xpath) : http://www.w3.org/TR/xpath
6. McGath R, Integrating Scientific Datasets and Digital Libraries.
7. Powel J, Multilingual Federated Searching Across Heterogeneous Collections
8. Iannella R, Metadata : Enabling the Internet
9. Ioannis Papadakis, A digital library framework
10. Baldonado M, Metadeta for Digital Libraries: Architecture and Design Rationale
11. Rauber A, Management of Distributed Information Repositories.

About Authors

Mr. Hardik Joshi is from Department of Computer Science, G.K. & C.K. Bosamia College, Jetpur – 360370, Gujarat, India.
E-mail: hardikjjoshi@yahoo.com

Mr. Manoj Vyas is from Department of Computer Science, G.K. & C.K. Bosamia College, Jetpur – 360370, Gujarat, India.
E-mail: manojvyas66@yahoo.com