ABSTRACT
Digital libraries are more and more available on the web. However, retrieving information in these libraries is not easy because of sources heterogeneity and distribution. Thus, we propose the use of virtual integration with mediator-wrapper architecture. This architecture allows access to relational sources, XML documents and text files. The mediator schema is based on XML and Xquery as query language.

KEYWORDS
Information retrieval, digital libraries, virtual integration, mediation, Xquery

1. INTRODUCTION
Digital libraries can be defined as collections of electronic documents. The documents contents are generally stored in text files, pdf files but more and more we find them in XML format. Information on documents (metadata) such as title, editor, ISBN,… are structured, for the most, in relational databases and sometimes in XML format. Digital libraries are now available on the web. However retrieving information in these libraries is very difficult. Indeed, users have to connect to each system’s interface, using a specific query language. When receiving results, he has to integrate them in order to get the final results. Different approaches exist for retrieving information on the web such as search engines, physical integration and virtual integration using mediators. We propose using XML-based mediation for information retrieval on the web or in private databases. We propose an architecture for an XML mediator that enables access to relational databases, XML documents and text files. Xquery [CHA02] is adopted as query language. We also present a relational wrapper architecture.

This paper is organized as follows. The second section details the different approaches for information retrieval. Section 3 motivates the use of XML-based mediation in digital libraries. In section 4, we present the mediator and wrapper architectures and we conclude by presenting our work in progress.

2. INFORMATION RETRIEVAL
To retrieve information on the web, different approaches exist: the first one is using search engines such as YAHOO!, GOOGLE or ALTAVISTA. Another approach consists of adopting physical integration. Virtual integration is another solution. In the next, we briefly discuss these approaches.
Search engines build indexes in order to speed up retrieval. Some of these engines such as ALTAVISTA use robot for updating indexes at regular interval or when accessing to server pages. By this way, they cover large number of sites. However, returned results are unreliable, with too many references sometimes or too few. Others search engines such as YAHOO! rely on human indexing. This has the advantage to give relevant results but web coverage is weak.

Retrieving information using search engines needs access to each interface and query this interface using specific language. To solve this problem, METACRAWLER, a meta engine, presents a unique interface. To treat the user’s query, this engine throws several search engines. This solution doesn’t solve the integration problem since the user has to treat results in order to eliminate redundancy. Furthermore it doesn’t allow access to databases.

Physical integration consists on creating datawarehouses from different sources. This has the advantage to speed up access to information but it requires importing, storing and refreshing data. This approach is used by INQUERY, DIALOG or LEXIS/NEXIS. These are system for information retrieval in private databases (repositories).

Virtual integration let the data where they are but provide unified access to them and integrate dynamically the results. This is realized by the mean of the mediator-wrapper architecture. The concept of mediation has already been proposed for retrieving information in digital libraries by MEDIATOR [GRO02] and MIX [BAR98]. However MEDIATOR applies the mediation to physical integration. In addition, it is limited to intranets whereas we propose a solution for the web. MIX has proposed the virtual integration with XML and XMAS, a proprietary query language. MIX allows access to relational, object and html sources. What we propose is the use of Xquery as query language and an XML wrapper to access to XML documents in addition to a relational wrapper and a text wrapper.

3. MEDIATION FOR DIGITAL LIBRARIES

We consider a virtual library composed of two sites S1 and S2. In the first one we have both relational and XML sources and in the second one we have XML sources.

The relational source contains references on the electronic documents. It is described in the next.

Table 1. Relational table for documents references.

<table>
<thead>
<tr>
<th>TITLE</th>
<th>PUBLISHER</th>
<th>AUTHORS</th>
<th>ISBN</th>
<th>SOURCENAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION TO DATABASE</td>
<td>EYROLLES</td>
<td>ADIBA S.</td>
<td>124-465</td>
<td>Db1.xml</td>
</tr>
<tr>
<td>DISTRIBUTED SYSTEMS</td>
<td>PRESS</td>
<td>FEE G.</td>
<td>321-548</td>
<td>Db2.xml</td>
</tr>
<tr>
<td>XML AND DATABASES</td>
<td>OREILLY</td>
<td>GEOR F.</td>
<td>321-328</td>
<td>Xbd.xml</td>
</tr>
</tbody>
</table>

Xbd.xml in S1 contains the contents of the book titled “XML AND DATABASE”.

```
    <book>
        <Title>XML AND DATABASE</Title>
        <Chapter>
            <Title>Data Model</Title>
            <section>
                <Title>Syntax for data model</Title>
                <section>
                    <Title>Basic Syntax</Title>
                    ....</section>
                ....</section>
        ....</Chapter>
    </book>
```

At S2, ref.xml contains information such as title, authors, editor, sourcename,... and docx.xml contains book contents. A user’s query that looks for documents about “XML” is as follows :

```
  <results>
    { for $b in view("reference") let $doc :=$b/sourcename
      where contains($b/Title/text(),"XML")
      return $doc };
```

858
4. MEDIATION ARCHITECTURE

The suggested architecture is based on virtual integration using mediator and wrappers [WIE92], one by resource. Today XML is a leading standard language for storage, exchange and data representation on the Web as well as in the enterprises. It is expected to have in the future most of documents in XML format. On the other hand, lot of information are already stored in relational database systems. In this context, our architecture allows access to relational resources as well as to XML documents. The service will offer Xquery, a uniform query language for both relational and XML data. Xquery is the W3C emerging standard. The mediator global schema consists of XML views of the resources. This allows hiding information which we don’t want to be published. These views are not materialized in order to avoid storing data when only part of them will be requested.

The user’s query is expressed in term of the mediator schema (global schema). In the Global As View (GAV) [LEV00], the global schema, at the mediator level, is described as view on the sources. To express the user’s query on the sources we only need view unfolding. The drawbacks are that modifying one source needs to reconsider the global schema. The other approach is known as the Local As View (LAV). In this case, the sources are described as views on the mediator schema. In this approach, it is easy to add or to delete sources but rewriting the user’s query from the global schema to the local (sources) schema is complex. Nevertheless algorithms to solve this problem exist for relational schema [HAL01].

In addition to the mediator and wrappers, our architecture provides a QBE interface for data interrogation. It can be requested to translate the user’s query to an Xquery query.

4.1 The Mediator

The Xquery is transformed at the mediator into an XML algebra tree [ZHA02] in order to be treated. The mediator is composed of one catalogue and five modules.

The catalogue is a metadata associated with the mediator. It describes the available resources by using DTD (Data Type Definition). It also indicates their physical localization.

The modules are the analysis and normalisation, the rewriting, the optimisation, the execution and integration and the formatting. In the next, we briefly describe each one.

The analysis and normalisation module has to analyze the queries. It rejects the syntactically incorrect queries. It eliminates also the queries that refer to elements that can’t be accessible. This is achieved by using the catalogue. Normalisation consists of transformation of complex queries (nested queries) into simple ones.

The rewriting module decomposes each simple query into sub-queries, one by resource. As we have adopted the GAV, then the mediator schema considers the data resources as XML views. The query expressed in terms of the mediator schema (global schema) is translated into a query according to the resources schema (local schemas) by unfolding the definitions of the mediated schema views.

The optimization module allows to determine if queries can be executed in parallel or series. So it generates an execution plan [MAN01].

The execution and integration module sends the query towards the adequate wrapper using the localization information produced by the catalogue. This module uses a cache in order to store the intermediate results.

The formatting module recovers the final result before presenting it in XML format.

4.2 Wrappers

Recall that the proposed architecture manipulates XML and relational data. So, the wrapper layer contains two kinds of wrappers : XML Wrapper and Relational Wrapper.
XML Wrapper: XML data need an engine to execute Xquery. The associated wrapper is a simple Xquery processor. It is thus a light wrapper.

Relational Wrapper: It transforms the Xquery query into SQL query. The relational wrapper is composed by the following modules: XML Algebra Tree generator, Decorrelator, SQL-pushdown, SQL generator, SQL executor and XML tagging.

The XML Algebra Tree generator module transforms the Xquery into an XML Algebra Tree. The decorrelator module eliminates the FOR operators and replaces them by the related sub-tree nodes. The resulted tree is then reorganized by the SQL pushdown module to pushdown the operators that can be translated to SQL. This translation is realized by the SQL generator module. The resulted query is sent by the SQL executor module to the RDBMS. The received tuples are formatted by the XML tagging which sends the final results to the mediator.

If an Xquery query cannot be transformed into SQL because of the semantic mismatch between the two models (relational and XML) [MAN01], the wrapper creates, temporarily, a materialized XML view on the requested tables and executes the query in Xquery engine.

One mediator can’t cover large amount of sources. But, using hierarchy of mediators enables larger web coverage. Indeed, an Xquery could be sent to a mediator of lower level or to a wrapper. The process executes recursively until the query arrives to a wrapper level.

5. CONCLUSION AND FUTURE WORK

In this paper, we have applied the concepts of virtual integration based on XML mediation. Then, we have presented services architecture of access to data. Thus, retrieving information on the web won’t be limited to text files but also to databases sources and XML documents.

We have developed, using Java components, the relational wrapper including the module that allows materializing XML views. Currently, we are implementing the rewriting module and we are about adding the normalisation functionality to the analysis module. This latter is realized using the QWEELT open source. The XML wrapper is implemented using an Xquery execution engine that is the IPSI open source.

Our future work is organised according to five axis. The first one is to develop the rest of the modules. The second one aims to extending the system in order to integrate data resulting from text files. This extension requires only the development of a suitable wrapper. This has not any impact on our architecture. The third research axis takes into account the semantic aspect by our mediation system. Because of the limitation of the GAV approach we are working to adopt the LAV approach for the global schema. This will affect only the rewriting module. Finally we project to consider the Arabic data both at query and at presentation levels.

REFERENCES