Complex Data Integration into an Active XML Repository

Rashed SALEM
supervised by Pr. Jérôme Darmont & Pr. Omar Boussaid
Ph.D. Student, Université de Lyon (ERIC Lyon 2)
5 av. P. Mendès-France 69500 Bron, France
http://eric.univ-lyon2.fr/
rashed.salem@univ-lyon2.fr

ABSTRACT

Decision-making processes are becoming complex, particularly with the presence of huge amounts of distributed and heterogeneous data. Therefore, warehousing these complex data into a unified format has become an urgent necessity. In this paper, we focus on the integration phase and propose an architecture for integrating relevant complex data into a repository of Active XML (AXML) documents. AXML documents are XML documents with embedded calls to Web services. They have the advantage of not requiring to store all data explicitly, but AXML frequently updatable data can be loaded on the fly via calling Web services if necessary.

Categories and Subject Descriptors
H.2.7 [Database Management]: Database Administration—Data warehouse and repository; H.3.5 [Online Information Services]: Web-based services

General Terms
Design, Standardization

Keywords
Complex data integration, AXML documents, Web services

1. INTRODUCTION

There are huge amounts of distributed and heterogeneous data available everywhere (e.g., operational data, Web data, XML data, text data, multimedia data, data streams, spatial data, etc.). Such data are needed in decision support. We term these data complex data [5]. Warehousing these complex data into specified storage to be later analyzed for decision support purposes is not a trivial, due to the heterogeneity and non-interoperability of data sources. However, traditional XML data warehousing approaches are not adequate when dealing with complex data. They also lack of active features and supporting excellent data freshness.

Therefore, we aim at designing, developing, implementing and validating a framework for active XML warehousing of complex data online, for intelligent decision support. The proposed framework is able to handle complex data, unify heterogeneous data into XML, integrate distributed data using Web services (WSs), refresh the warehouse with up-to-date data and support near-real-time decisions on-line. Data heterogeneity of complex data motivates us to exploit XML in order to standardize data into a unified format. In order to add more functionality to the warehoused XML documents, we embed them with calls to WSs. WSs can solve the data distribution and interoperability problems by exchanging data between different applications and different platforms. Embedding WSs into XML documents results in so-called AXML documents [1].

In this paper, we propose an active XML integration architecture enriched by active features utilizing WS technology. In this architecture, WSs are employed for performing ETL tasks. Therefore, WSs are responsible for integrating data from their distributed sources upon independent platforms. WSs of AXML documents can also be invoked to get information via the Web, to invoke specific ETL services, and/or to execute specific XQueries over other AXML documents. The rest of this paper is organized as follows; related work is discussed in section 2. Section 3 demonstrates the proposed architecture for integrating complex data. In section 4, we highlight some implementation issues and challenges. Finally, we conclude and discuss future trends in section 5.

2. RELATED WORK

Active XML warehousing is based mainly on XML warehousing and AXML technology.

2.1 XML Warehousing

Several researchers address the problem of designing and building XML data warehouses (XML-DW). They propose to use XML documents to manage, model or represent facts and dimensions. The main purpose of these approaches is to enable a native storage of the warehouse and to query it using XML query languages. XML-DW research is divided into three families [8]: Web data integration for decision-support purposes, adapting approaches based on classical warehouse logical models and document warehousing.

XML is becoming the standard for exchanging semi-structured data over the Web. Integrating these data into warehousing environments is becoming increasingly common. In Xyleme, authors build a dynamic warehouse for massive volumes of XML data from the Web [16]. Golfarelli et al. propose a
2.3 Active XML

Abiteboul et al. propose Active XML as a useful paradigm for distributed data management on the Web. AXML documents are XML documents where some of the data are given explicitly, while other parts are given only intentionally by means of embedding calls to WSs. When one of these calls is invoked, its results is returned to enrich the original documents [1]. The authors also studied issues raised by the distribution and replication of dynamic XML data. Several performance issues for Active XML are addressed, e.g., the problem of guiding the materialization process [9]. Ruberg and Mattoso also deal with that problem when the result of some service calls can be used as input of other calls [13].

However, Abiteboul et al. utilize AXML and WS technologies to present an approach for building and maintaining domain-specific content warehouses [2]. Their active content warehousing approach is aimed at storing data for management purposes, but our active multidimensional warehousing stores data for analysis purposes.

3. AXML-BASED DW ARCHITECTURE

This architecture handles complex data integration. It integrates relevant complex data from different data sources utilizing ETL services, and yields the store of AXML documents. Thus, our architecture is composed of three main parts: sources of complex data, ETL services, and AXML store (Figure 1). The architecture is described and managed via metadata and AXML engine.

There are different functionalities in integration processes:

1. Identification of different heterogeneous data sources:
   (a) Identification of their location, types, interfaces,
   (b) Identification of relevant data at these sources;
2. Data extraction via different interfaces;
3. Data integration and standardization into XML;
4. Identification of the most frequently updatable data to be embedded in XML as calls to WSs;
5. Data propagation into the AXML repository.

3.1 Sources of Complex Data

Most traditional DWs integrate data from operational data sources only. In our architecture, we deal with complex data [5], which come from distributed and heterogeneous data sources. Data sources can be listed as input schema via administrator dashboard. The input schema aims at profiling data sources, such as:

- In operational and structured sources, the input schema defines interesting tables, fields, and/or records, etc.
- In Web data sources, the input schema defines relevant texts, tables, links, images, etc.
- In XML and semi-structured data sources, the input schema defines relevant elements, attributes, etc.
- In text and flat-file data sources, the input schema defines relevant keywords, summaries, concepts, etc.
- In multimedia data sources, the input schema defines media type, color, shape, content, description, etc.

3.2 ETL Services

ETL services are responsible for extracting relevant data from data sources, transforming them into XML, and loading them into the AXML repository. These services deal with heterogeneous data sources via different interfaces. They are not only qualified for integrating low-level characteristics (e.g., image color, file size, file location, creation date, update date, etc.), but also go further to integrate the semantic characteristics (e.g., image content, relationship between objects, etc.). However, integrating semantic features requires semantic data retrieval and mining techniques. We exploit WSs to develop ETL tasks, for several reasons. They can solve the data distribution and interoperability problems, where WSs interfaces can be applied to evaluate specific queries in remote, independent and heterogeneous data sources. WSs can also be embedded in XML documents and then can be invoked to refresh the documents on the fly.

3.3 Active XML Repository

The AXML repository is the target where data are loaded. The output of integration tasks are AXML documents. AXML documents are XML documents with embedded WSs, which are invoked when querying AXML documents to refresh their contents with up-to-date data. The evaluation of WSs is managed by the AXML engine to call and execute specific
ETL services or WWW publishing service (e.g., Google, Yahoo, Amazon WSs, etc.). AXML documents are stored in a native XML database (NXD). NXD systems support managing, storing, querying and updating XML data. However, the main drawback of AXML is that there is no browser to parse and read AXML documents. Therefore, AXML finds difficulty to be supported by end-user applications.

3.4 AXML Engine

AXML engine manages the evaluation of AXML WSs and return results to refresh the documents with up-to-date data. For example, when querying a specific AXML document, it receives the embedded calls to WSs and then execute these WSs via specific interfaces. Some issues such as, what data should be given explicitly? and what data should be given implicitly? should be tackled by the administrator according to update rate of data. Another important issue is timing of invoking the WSs, which can be either explicitly (e.g., daily, weekly, etc.), or implicitly when data are requested.

3.5 Metadata

Metadata are stored in a repository concerning different modules of the architecture. Thus, metadata may contain:

- Description of data sources, interfaces, descriptive information such as ownership, source, format, structure, update frequencies, access methods, etc.
- Description of ETL services, including sources - to extracting services interfaces, transformation and cleaning rules, data elements-mapping names, targeted AXML documents, updating and refreshing policies, etc.
- Description of AXML repository’s structure, history of extracting data, physical location of AXML documents, refreshing plan, etc.

4. IMPLEMENTATION ISSUES

We are implementing our architecture using mostly standard open-source and free available software such as: Java JDK 6, Tomcat 6 Server engine, Axis2, JSP and Applets. Thanks to different APIs that facilitate integration tasks either for reading from data sources or writing into the target repository (i.e., JDBC, J/ O, JMF, JAXP, XQJ, StAX, etc.). ETL services are deployed on a Tomcat Web application server via AXIS2, which generates WSDL files automatically. AXIS2, the core engine for WSs, allows the interaction among distant machines over a networked environment. Then, WSs can be invoked from anywhere using their URL. The complete implementation is not yet finished. Now, we face some difficulties such as handling unstructured data, dynamic Web sources and integrating semantic data. To this end, we may embed some external modules to our implementation to handle these issues. Finally, we aim at developing the AXML warehouse as a Web-based application, where users can run it entirely through a browser.

5. CONCLUSIONS AND FUTURE TRENDS

In this paper, we presented an architecture for integrating complex data based on AXML technology. The integration tasks are expressed as WSs. We are motivated to exploit AXML to integrate complex data. AXML has an extensibility feature to represent, model and store complex data, as well as to cope with the problem of distribution and interaction over different networks, due to XML and WSs. Moreover, AXML can support excellent data freshness via evaluating their embedded WSs on-line. In near future, we plan to apply more intelligent features to the architecture. It will be enriched with active rules to realize the integration tasks in autonomous and intelligent behavior.

6. ACKNOWLEDGMENTS

I gratefully acknowledge the support of my Ph.D. supervisors Pr. Jérôme Darmont and Pr. Omar Boussaid.

7. REFERENCES