Integration of Heterogeneous Databases based on XML

Venciya.A
Student, Department Of Computer Science And Engineering, SRM University, Kattankulathur, Venciya.a@gmail.com

Abstract

As companies grow bigger and bigger the need to integrate data stored across the various departments becomes crucial for the proper functioning of the company as a whole. In this paper we start with a brief summary of the existing approaches commonly used in heterogeneous database integration followed by a new and unique approach of integration of heterogeneous databases based on XML. Compared to conventional models, this model is simple, efficient and cost effective and hence can be widely used in businesses spread across multiple locations.

Keywords: XML, Heterogeneous database, Template Mapping, Model Mapping, Data Type Mapping.

1. Introduction.

With the rapid growth of technology and the constant need to derive information through processing of data, many enterprises are currently facing the problem of sharing data across its various departments. Most industries start as a small-scale project with a single department but as the necessity to extend the functionality of the organization and establish various other inter-related departments arise; they face various challenges in integrating databases across multiple sources. The isolated database that was once sufficient to run the company is no longer effective in compiling data across the different departments or geographical location of the organization. For example, initially the company spends a lot of money developing a well operated financial, production and management system but these systems are developed by the company at different times, the platforms used are also different. This leads to increasingly isolated chunks of data that requires a common platform in order to be converted into a processed form. Thus there is a co-existence of various heterogeneous databases. As the heterogeneous data grows with each passing day, the interoperability of the various departments faces a serious setback. This so-called information island causes barriers in communication across the various departments. Cross-industry communication is also affected. In today’s society it is mandatory that companies must be able to respond quickly to the market pressure and make right decisions by analyzing large volumes of data, else the company is bound to suffer a humungous loss. Isolated data stored in different formats does not allow a company to perform to its full capacity. Thus the company must strive to integrate this heterogeneous data. The current heterogeneous database information sharing methods lays too much emphasis on the integration of various applications, but often overlooks another fundamental and important issue closely related to it, namely the data integration. In this paper we present a method of integration of heterogeneous data based on XML that aims to provide an efficient and highly effective integration of data across various departments in an organization.

2. Heterogeneous Database Integration Strategy And Methods.

Heterogeneous databases can be defined as a collection of databases that are fundamentally dissimilar to each other, each with its own processing techniques and data format. The objective of this paper is to describe the construction of a transparent global database required by enterprises for integrating the distributed multiple heterogeneous databases with little
impact on the local autonomy and to facilitate information exchange and sharing between them.

2.1. Heterogeneous Database Integration Strategy.

Heterogeneous database integration is generally accomplished by conversion and standardization. This generally solves the platform and network transparency through data model conversion, model transformation and integration, distributed transaction management, and other problems. The current heterogeneous database system integration mainly uses three strategies:

1) Public programming interface,
2) The public database gateway and
3) Public agreement, as shown in figure 1.

These strategies are all based on a client/server architecture, so that all computers can be integrated to work together, thus accomplishing computer application system optimization

Figure 1. Main strategies used in heterogeneous database system integration.

2.2. Heterogeneous Database Integration Methods.

Before the heterogeneous database can be integrated two requirements must be met

1) All the information that needs to be shared in the source database mode should be converted into the destination database.

2) Data transformation process shouldn’t contain redundant association Information.

The major conversion methods involve the use of

1) Software tools
2) Third-Party Database
3) Database Components

We now go ahead to compare the various transformation methods:

Fig 2. Comparison of various conversion methods.

3. Xml-Based Heterogeneous Database Integration.

XML- Extensible Markup Language is a widely used portable and operating system independent information description language that is used to express a wide range of processed data. It provides a very efficient and effective means of data representation and processing of content present in the databases. XML is also used to manage multimedia data and complex
synthetic data; one can also reasonably decompose the complex data to meet the data model being used.

The data description along with specification of relationship between data can be achieved using XML Schema.

The heterogeneous database integration model (based on XML) is shown as in figure 3.

Figure 3: Proposed model for heterogeneous database integration using XML.

The procedure in brief is as follows, an intrinsic mapping relationship is established between the heterogeneous data format and XML Format, and then the heterogeneous data forms an XML document. XML technologies are applied iteratively in order to obtain an XML integrated module. A package body is added with XML as the main format in all data that is obtained from the various departmental sources. Thus no change in data source should take place. Extensible markup language is then used to pack all information such as data source definition, description etc.

3.1. Mapping Between XML And Relational Database.

Mapping between XML and relational database plays a vital role in the integration process but due to a huge difference in the underlying data structure that is used between the XML and a relational database, one cannot directly store XML data in a relational database. The table structure must be reflected as a DID or XML schema document followed by further processing. This mapping stage plays a very crucial role in the integration of data across various sources. As XML has become a standard intermediate format the cost of conversion has decreased drastically. Two types of mapping methods frequently used are template-driven and model-driven mapping. In this paper a new approach called Data type mapping is introduced.

3.1.1. Template mapping

In this method a SQL command is embedded in a predefined template with certain parameters. Template mapping is a shallow map.

Template mapping proceeds as follows

SQL

<XML Version 1.0="1.0">
<BankInfo>
<SelectEmployee>
 select EmployeeId, EmployeeName
from BankInfo
where EmployeeName='Mary'
and EmployeeAge='30'
</SelectEmployee>
</BankInfo>
</SelectEmployee>

XML

<XML Version 1.0="1.0">
<BankInfo>
</BankInfo>
</SelectEmployee>
Data results are embedded in XML, it does not take into consideration relation mode or the object mode. The constraints of the database mode are not always kept into consideration. Thus it is used only in data transmission between XML documents and the relational database. It does not support object oriented databases. It is a shallow mapping technique, as it does not support two way mapping necessary in object-relational databases etc.

3.1.2. Model mapping.

Instead of relying solely on embedding SQL Commands, the model mapping technique uses a specific underlying model. The XML model is based on XML schemas or DTD. The data base model might be a relational model or an object-oriented model. This model depends on establishing a mapping between these two models. Thus the XML document is generated based on the underlying relational database structure. The success of this model depends on the ability to map the database model and the XML schema and generate an XML document.

3.2. Xml Storage Strategies.

XML is generally used as a tool in data management. The expansion of XML has brought about the proliferation of XML Data. The widely used management methods include:

1) The file system
2) XML-Enabled database, and
3) Native XML database.

It is commonly seen that the document-centric XML data is accessed as a large object and the data-centric XML document is stored in relational tables.

3.3. Data Type Mapping Between Xml And Database.

Two questions that commonly arise when converting data between XML documents and databases is as follows:

1) What way is to be used to generate DTD of XML from the given database structure and
2) What way is to be used to generate database structure from the DTD of XML.

In data type mapping the steps involved in generating XML Document from relational database structure are as follows:

Create a new element for each table present.

1) Create a sub-element for each column present, which contains only the PCDATA.
2) Create a new sub-element for each column displaying a foreign key relationship.
3) Create a new sub-element for each primary key.

The steps involved in generating a relational structure is as follows:

1) For each element containing a sub-element generate a table and a primary key field.
2) For each element type having multi-valued attributes and occurring only once, generate a separate table, and through the primary key element form a link to the parent table.
3) For each single valued attribute, create a field in the table. If the attribute has a null value this field can be empty.
4) For sub-elements with multi-valued attributes generate a separate table and through primary key field create a link to the parent table.

In data mapping the following steps are used to generate a DTD from a relational structure:

1) Generate a new element, for each table,
IV. Conclusion.

With the advent of different Database Management Software, there is a need to integrate data generated from various sources in order to obtain information that is pivotal for the success of any company. In order to obtain the highest possible efficiency in processing the data stored in different formats across the various departments of an organization, it is necessary to integrate the heterogeneous database using XML. A built in template and system format seriously affects the information exchange between the various systems. Thus for a cost efficient solution a database integration technique based on XML should be adopted.

V. References.


