XML-based Heterogeneous Database Integration System Design and Implementation

Wu Xiaoli, Yao Yuan
Department of Computer Science and Engineering, Henan University of Urban Construction
wxl-pds@sohu.com

Abstract—In this paper, based on the research of the existing heterogeneous database integration systems, according to the data exchange and sharing needs of enterprise heterogeneous databases, a framework for heterogeneous database integration system is proposed and designed, and the key technologies of the system implementation process are also described in detail. The system provides a heterogeneous data sharing and integration middle platform to achieve transparent operation and seamless integration of the heterogeneous data.

Keywords—heterogeneous database; XML; data integration

I. INTRODUCTION

In the new environment of the information age and global economic competition, especially the rapid development of enterprise e-commerce, data resource sharing has become a hot topic. Many businesses need to integrate DBMS, MIS, and OA data to form a network information platform for enterprise management decision-making. The integrated information of the past information integration systems can not be expressed by a standardized data form, but a system-defined data format, which seriously impacts on the information exchange between various systems, and the achieving process is complex, the cost is higher, difficult to get a wide range of applications, there is an urgent need for a new data integration system.

The emergence of XML technology makes the standardization description of all kinds of irregular information and rule information possible, and gradually becomes the standard to describe the data in Internet, and sets up an enterprise information integration platform in the XML technology, is an inevitable trend of information technology development[1]. XML as an extensible markup language, its self-description makes XML itself suitable for data exchange between heterogeneous applications, and this exchange does not make the pre-specified data structure definition as the premise, so it has strong open and broad application prospects. And almost all of the existing large-scale applications are related to the database, so the data exchange and information sharing with XML as neutral carriers must be linked to databases; at the same time, XML-based data exchange and the realization of the database XML data bidirectional access can integrate the XML data with specific applications, thereby enabling the combination with the existing business rules, and finally truly achieving XML-based distributed data exchange and information integration. Therefore, the combination of XML and database technology has become an inevitable trend, and the achievement of data two-way transmission between the XML document and database become one of the hot in XML technology, XML-based database applications also have developed rapidly.

II. HETEROGENEOUS DATABASE INTEGRATION AND XML LANGUAGES

Data integration is to provide a unified representation, storage and data management for various heterogeneous data through the network. XML as the basic function, the heterogeneous data integration system must implement. Data integration shields the heterogeneity of the various heterogeneous data sources, and carries out unified operation to different data sources through heterogeneous data integration system. Therefore, the integrated heterogeneous data is unified for users.

A. Heterogeneous database integration

The data forms involved in heterogeneous database are mainly structured data, semi-structured data and unstructured data three types. Structured data widely exists in a variety of information system database, the most common relational database. Semi-structured data commonly has Web pages as the chief representative, and XML can effectively manage and process such data. Unstructured data has common files, email and various documents. A practical information integration system should have intelligence, openness and initiative. Intelligence is to carry out unified processing, filtering, reduction, abstraction, integration and induction works for the structured, semi-structured and unstructured data from different databases[2]. Openness is a heterogeneous and distributed database, which must solve the mismatching problem of the information expression with the structure. Initiative is to regulate the existing Internet data representation, exchange and service mechanism to provide proactive service mechanism.

B. XML languages

XML is developed and established by the Internet organization W3C, the purpose is not only to meet the ever-growing network application needs, but also to ensure it have good reliability and interoperability in the alternation through the network. XML is a structured and semi-structured data markup language, to define a set of common format for the structured documents and data on Web pages and provide a way for the structured data to write a text file. XML is a markup language independent of the system to express data information, has become a common
data exchange format in network system. XML has been widely used in computer and network-related aspects. The nodes of the XML document can be read through DOM, which is the most fundamental and underlying XML access technology. DOM is a W3C technical standard, in fact, is to provide a set of API to access XML data, which can be achieved through JavaScript, Jscript, VBScript and other scripting programs, also through C++, Java and other high-level languages \[6\] . The system structure of using XML as intermediate data to achieve data exchange between heterogeneous databases is shown as figure 1.

![Figure 1. Architecture of XML-based heterogeneous database data exchange](image1)

III. DATABASE INTEGRATION SYSTEM DESIGN AND IMPLEMENTATION

A. Database integration system design

XML-based heterogeneous database integration middleware system is mainly in view of the fact that the XML has become the important standard medium of information sharing and data exchange in the Internet, and the database is the main data carrier of enterprises. The system is established in reference the existing research results and combining the current mainstream technology, the main propose of which is to provide a middle platform of heterogeneous data sharing and integration to achieve the data transparent operation and seamless integration, the system architecture as shown in figure 2.

![Figure 2. Database integration system architecture](image2)

B. Database integration system implementation

1) Query manager

Query manager is to provide a unified interface for the client to query the data source, responsible for receiving the global query requirements from browser, and then according to the corresponding integration information to decompose them into several local query requests to be transmitted to the data packager, finally to process the results returned from it to return to the browser, while to maintain the integrated information to ensure the accuracy and consistency of the global transaction execution. From the data conversion process, the query manager integrated the existing shared data model and data. The integration is mainly embodied to the integration between the existing different databases, the need to differentiate, despite the current database middleware also support data integration, but their data integration refers to the access to different databases through them, that is, the access integration, rather than the establishment of links between the data of different databases. The integration in this paper not only includes the data access integration, but also the simple integration between the data of heterogeneous databases. The reason for integration is that for a specific task between the heterogeneous databases their may exist a general logical link, and the data stored in them may be different stages of the same thing description. Query manager is responsible for submitting the standard query the client submitted to the query parser, the query analysis unit is to realize the query parsing and use meta data management module to check the legitimacy of the query request to judge whether the query data exists, and convert the query request into standardized form, while recording the result set target format to submit it to the appropriate data packager \[4\] ; Query integrator is responsible for collecting the local query results about a certain query some data packager returned, combining them into the mode users needed, and then sending back to the client browser.
2) Data packager

Data conversion is the core of the heterogeneous database integration middleware system and the key of ultimate implementation, which is achieved by the data packager. In terms of the heterogeneous data sources, data packager is the translator of the relationship mode. Usually, the data packager is a program of data extraction and transformation, which is responsible for receiving the query command the query distribution unit sent, and then through standard interfaces JDBC to send to the data source for query execution to obtain query data and organize them according to the query structure as the result document the XML structure expressed, and finally integrate them to the query result integrator.

Data packager should be able to access certain types of database systems and can easily upgrade the scope it supported, and it should be not be an interface specifically designed for a certain database, which should be a functional body with more flexible structure, allowing a range of configurations; the data conversion of it should be constrained integrity based on pattern, that is, as far as possible reflect the original source data, including the relationship between these data; it should provide the appropriate structure, need to communicate with the query manager, and good interface is the prerequisite to achieve the interaction of them; data packager should support transaction processing, to enhance fault tolerance. XML is used as the data exchange format of the data packager to achieve the view transformation and management of different data sources from the local to the global model.

C. Key technologies of database integration system

1) JavaBean technology

JavaBean is a software component model to describe Java, somewhat similar to Microsoft COM component concept. In the Java model, the functions of the Java program can be infinitely expanded by JavaBean, and new applications can be rapidly generated through the JavaBean combination. JavaBean also can achieve code reuse, while has very great significance for the program maintenance. Through the Java virtual machine JavaBean can be run correctly. JavaBean provides for the Java component-based development system. And the query manager and data packager in this system are all the JavaBean components based on the Java language.

2) Connection pool

Connection pool is a kind of entity which manages the connection as a resource, and a typical example of such resource is the database connection. The basic idea of the connection pool is to pre-establish some connections to store in the memory for use. To establish a database connection will consume considerable system resources, but once established, the query can be sent to obtain results through it. The number of queries a connection in its life cycle can process is not limit, so a database connection from a certain way is a resource.

Using connection pool, when the program needs to establish a database connection, it only needs to take one from the memory to use instead of new. Similarly, after use, simply to replace to the memory and the connection establishment and disconnection are both managed by the connection pool itself. At the same time, we can also through setting connection pool parameters to control the number of connections and the maximum use frequency of each connection. The use of connection pool will greatly enhance the process efficiency, and we can through its own management mechanism to monitor the quantity, use of the database connection. The connection pool technology allows the data packager efficiently, stably and reliably access to the database connection, to minimize the waste of data resources.

Tomcat is the standard of the Java Servlet 2.2 and JavaServer Pages 1.1 technologies, is free software developed based on the Apache license. Tomcat application server itself comes with database connection pool features, so administrators can modify the appropriate values according to needs and the hardware configurations to achieve the best results. The more commonly used parameters such as maximum number of requests received, connection timeout, connection upload timeout, buffer, the maximum number of active connections, the minimum idle connection, and so on. Therefore, this paper directly uses the database connection pool functions of the Tomcat application server itself.

3) LOB data processing

LOB represents large object data, including BLOB and CLOB two types, the former is used to store large binary data, such as image data, video data, and so on, and the latter is used to store long text data, such as the forum post content, product detailed description. What is worth noting is that in different databases the field types correspond to the large object are different, such as DB2 corresponds to BLOB/CLOB, MySql corresponds to BLOB / LONGBLOB, SqlServer corresponds to IMAGE / TEXT. What is needed to be noted that some database large object types can be accessed as simple type, such as MySql's LONGBLOB operation and VARCHAR type are the same way. Under normal circumstances, the access method of the LOB data is different from the other simple types of data, we often uses stream way to operate the data with LOB types. Furthermore, the LOB data access is not thread-safe, needs to be separately allocated the appropriate database resources and release the resources after the completion of the operation. Finally, Oracle 9i uses a non-JDBC standard API operation LOB data. All of these cases bring challenges to the LOB data program compiling, and spring provides the corresponding help class in the org.springframework.jdbc.support.lob package.

Spring greatly reduces the difficulty of LOB data processing. First of all, Spring provides a NativejdbcExtractor interface, you can select the corresponding implementation class from the different environments to obtain the local JDBC object from the data source; second, through LobCreator interfaces Spring cancels the difference of different data vendors operating LOB data, and provides the LobHandler interface to create LobCreator, so we should only select the right LobHandler to configure according to the underlying database.
IV. CONCLUSION

Database application has a large number of data stored with different forms and rely on different database management systems, so how to share these data is the problem required to be solved. How to effectively carry out heterogeneous database integration is an important research topic. In past information integration systems, the integrated information could not displayed in a standardized form, but a system-defined format, which seriously affected the information exchange between the various systems, the achieving process was complex, and the cost are higher, difficult to be widely used. Therefore, a new data integration system is urgently needed. In this paper, based on the research of the existing heterogeneous database integration systems, according to the data exchange and sharing needs of enterprise heterogeneous databases, a framework for heterogeneous database integration system is proposed and designed, and the key technologies of the system implementation process are also described in detail. The system provides a heterogeneous data sharing and integration middle platform to achieve transparent operation and seamless integration of the heterogeneous data, allowing users to more easily publish data to the Internet/Intranet, to provide a technical basis for users’ heterogeneous data sources at a higher level.

REFERENCES