

On Heterogeneous Database Integration Under the Scope of Web Service and Java

Fanxing Meng, Xiumei Wen, Jianhua Li
Hebei Institute of Architecture Civil Engineering
Zhang Jia Kou, China
xiumeiwen@163.com

Abstract— More and more accessible databases are available in the Web. In order to provide people a unified access to these Web heterogeneous databases and achieve information from them automatically. This paper proposes a comprehensive solution for heterogeneous database integration, explains the heterogeneity of the data source, analyzes the method of integration mod of heterogeneous data source by using interconnect technology of database and considers mainly the integration about the heterogeneous data source based on Web Service and Java. This solution is extensible and portable.

Keywords- *Heterogeneous Database Integration;WebService;XML; JDBC; Middle ware*

I. INTRODUCTION

Today's business applications rarely live in isolation. Users expect instant access to all business functions an enterprise can offer, regardless of which system the functionality may reside in. An enterprise has multiple applications that they are developed independently, with different languages and platforms. The enterprise needs information shared rapidly and consistently. This requires the disparate applications to connect into a larger, integrated solution. Usually this integration achieves through using some form of "middleware". Middleware provides the "plumbing" such as data transport, data transformation, and routing.

II. HETEROGENEOUS DATABASE

A Heterogeneous Database System is an automated (or semi-automated) system for the integration of heterogeneous, disparate database management systems to present a user with a single, unified query interface.

Heterogeneous database systems are computational models and software implementations that provide heterogeneous database integration.

Data integration provides a union presentation, storage and management for every heterogeneous data. It shields the difference among each heterogeneous data, makes the heterogeneous data after integration unified and the same for users. For the current data integration system, the data of the most data sources belong to the heterogeneous data. So data integration means heterogeneous data integration.

Heterogeneous data is a concept of rich meaning, not only the means of heterogeneous database system (such as the oracle and SQL Server database system) but also including heterogeneous data among the different structures

(such as data of structural database and semi-structure XML data).

Along with the development of the informatization construction, data processing modes have differences in time and space, the data description has many methods, and data environment difference is bigger and bigger, these display mainly in the following respects.

(1) System heterogeneity: the difference among service application system on which the data source depends, database management systems and operating system constitutes the system heterogeneous.

(2) Model heterogeneity: namely the storage model difference of data sources. The storage model includes relation model, object model, object-relation model and document embedding model etc. Thereinto the relation model is the mainstream storage model. However even the same kind of storage model may have differences in model structure.

(3) Data heterogeneity: different data source has different data type, such as structure data (database), non-structure data (text, picture) and semi-structure data (HTML, XML).

(4) Semantic heterogeneity: the same data form can express the different semantics or the identical semantics is represented by the data of different forms.

III. WEB SERVICE

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

A. Agents and Services

A Web service is an abstract notion that must be implemented by a concrete agent. (As shown in Fig. 1) The agent is the concrete piece of software or hardware that sends and receives messages, while the service is the resource characterized by the abstract set of functionality that is provided. To illustrate this distinction, you might implement a particular Web service using one agent one day (perhaps written in one programming language), and a different agent the next day (perhaps written in a different programming language) with the same functionality. Although the agent may have changed, the Web service remains the same.

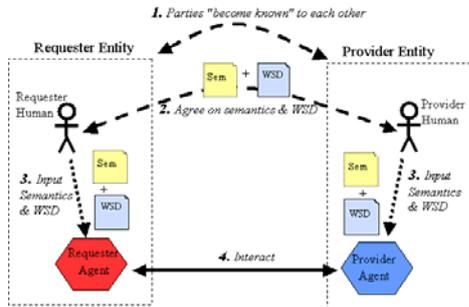


Figure 1. The General Process of Engaging a Web Service

B. Requesters and Providers

The purpose of a Web service is to provide some functionality on behalf of its owner -- a person or organization, such as a business or an individual. The provider entity is the person or organization that provides an appropriate agent to implement a particular service. A requester entity is a person or organization that wishes to make use of a provider entity's Web service. It will use a requester agent to exchange messages with the provider entity's provider agent.

C. Service Description

The mechanics of the message exchange are documented in a Web service description (WSD). The WSD is a machine-processable specification of the Web service's interface, written in WSDL. It defines the message formats, data types, transport protocols, and transport serialization formats that should be used between the requester agent and the provider agent. It also specifies one or more network locations at which a provider agent can be invoked, and may provide some information about the message exchange pattern that is expected. In essence, the service description represents an agreement governing the mechanics of interacting with that service.

D. Semantics

The semantics of a Web service is the shared expectation about the behavior of the service, in particular in response to messages that are sent to it. In effect, this is the "contract" between the requester entity and the provider entity regarding the purpose and consequences of the interaction. Although this contract represents the overall agreement between the requester entity and the provider entity on how and why their respective agents will interact, it is not necessarily written or explicitly negotiated. It may be explicit or implicit, oral or written, machine-processable or human oriented, and it may be a legal agreement or an informal (non-legal) agreement.

While the service description represents a contract governing the mechanics of interacting with a particular service, the semantics represents a contract governing the meaning and purpose of that interaction. The dividing line between these two is not necessarily rigid. As more semantically rich languages are used to describe the

mechanics of the interaction, more of the essential information may migrate from the informal semantics to the service description. As this migration occurs, more of the work required to achieve successful interaction can be automated.

E. Engaging a Web Service

In general, the following broad steps are required, as illustrated in Fig. 1.

- (1) The requester and provider entities become known to each other (or at least one becomes know to the other).
- (2) The requester and provider entities somehow agree on the service description and semantics that will govern the interaction between the requester and provider agents.
- (3) The service description and semantics are realized by the requester and provider agents.
- (4) The requester and provider agents exchange messages, thus performing some task on behalf of the requester and provider entities.

IV. WEB SERVICES TECHNOLOGIES

Web service architecture involves many layered and interrelated technologies. There are many ways to visualize these technologies, just as there are many ways to build and use Web services. Fig. 2 below provides one illustration of some of these technology families.

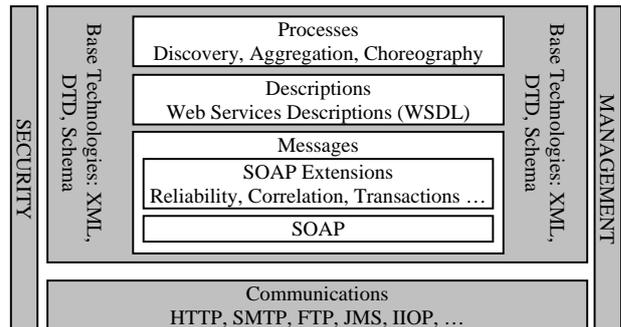


Figure 2. Web Services Architecture Stack

A. XML (eXtensible Markup Language)

XML is used to organize documents and business data. XML files can be stored or transmitted between two applications on a network. Basically, they are just plain text documents that contain special tags that label different parts of a document or fields of data.

B. SOAP (Simple Object Access Protocol)

SOAP defines a standard packaging format for transmitting XML data between applications on a network. A SOAP message is just an XML document. SOAP is specially designed, however, to contain and transmit other XML documents as well as information related to routing, processing, security, transactions, and other qualities of service.

C. WSDL (Web Services Description Language)

WSDL is a language for describing Web services and it is a standard for describing the structure of the XML data exchanged between two systems using SOAP. WSDL describes Web services starting with the messages that are exchanged between the requester and provider agents. The messages themselves are described abstractly and then bound to a concrete network protocol and message format.

D. UDDI (Universal Description, Discovery, and Integration)

UDDI defines a standard set of Web service operations (methods) that are used to store and look up information about other Web service applications. In other words, UDDI defines a standard SOAP-based interface for a Web services registry. You can use a UDDI registry to find a particular type of Web service, or to find out about the Web services hosted by a specific organization. A UDDI registry is often referred to as a "Yellow Pages" for Web services. The UDDI specification is now maintained by the Organization for the Advancement of Structured Information Standards (OASIS).

V. THE DATABASE INTEGRATION

A. The Combination of Java and XML in Web Service

Java is an Internet programming language. Java is simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, multithreaded, high-performance, and dynamic. Now it has been the standard programming language for network. XML can create the open data that do not be dependent on platform, language or the restricted format. XML has been widely used, and it has become the standard language of the younger generation for the data exchange on Web. Both Java language and XML are independent of the platform, they can accomplish the most complex and the best elastic distributed applications. XML technology is expected to bring the revolutionary influence, especially in the data exchange field. Java with XML makes the new generation Web application such as in e-commerce and the enterprise application fields possible. Almost all Internet technology participants support the XML technology. Some companies such as IBM, Oracle are developing products and technologies that will use XML and Java technology. Many web developers draw the conclusion that XML and Java is the perfect couple, they supplement each other. XML contributes to independent platform and easily extracting information and data. Java conduces to independent platform and is fit for the object-oriented application software solutions. So you can get the following formula: Java + XML = portable data and behavior.

B. Integrate database with the Web

The key reason for using database for data storage is that the database can be integrated easily with the Web. This is a real revolution in terms of information sharing and exchanging. It brings us enormous opportunity and convenience for sharing data. The Web is a client-server type of network, using a series of protocols collectively called TCP/IP (Transmission Control Protocol/Internet Protocol),

for communication. Client is any computer or program requesting a service or file. Server is the computer or program accepting the request and returning the requested file. We can use client-server architecture to link our database with any users in any place. This makes it possible that we can sit in office querying the data in database, or conducting search using the computer instead of searching the index page of a journal.

C. The Heterogeneous Database Integration Based on Webservice and Java

If every data source and destination data source have no unified middle pattern in data integration, the cost will be high. Because there is no unified middle pattern, the data analysis handling model is used to convert the data of some data source into the different destination data source. Therefore, the key problem of the data integration system is to build a good common data model that is a global data model for more data sources. A good global data model should meet the following demands:

- (1) It can describe all data format whether it is the structural or semi-structural, whether it supports all the query languages or it supports simple text query. So, the data model that is self-described is more suitable.
- (2) It is published easily, it can make data exchange, the integrated data may be published conveniently in more formats and be used to exchange data.

XML is a meta-language standard whose data format is self-described. Using the XML as the global data model can enable the middle ware system of the data integration to be suitable for every data integration environment.

Java provides strong support for XML, provides many API and technologies to support developing XML. Therefore, Java has been considered the most suitable language for XML programming. Java independent of platform and data XML independent of platform can finish the most complex and best elastic distributed applications. So, the data integration solution based on XML using Java technology is put forward, it is shown in Fig. 3.

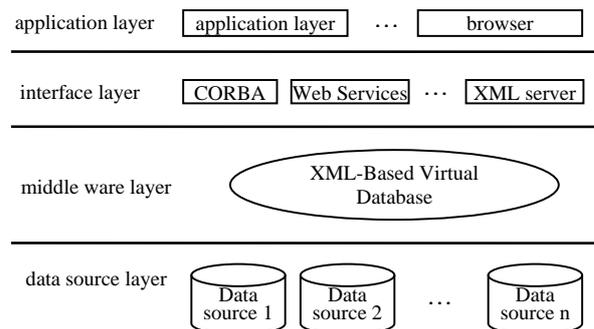


Figure 3. the frame of heterogenous database integration system

In this model, according to the user's query middle ware layer brings the multiple data source data into the same XML file and transmits it to the client, the generated XML files can be edited locally or be manipulated.

The user's access and operation to information do not act directly on every data source. The users access the data through the XML-based virtual Database. The users do not need to understand the specific location and platform type of the querying data, thus the user's transparent access to data is realized.

This system is divided into data source layer, middle ware layer, interface layer and application layer from bottom to top. The function of each layer is as follows:

(1) Data source layer: it is located in the most bottom layer, it is the data provider of the system, it includes the various kinds of information such as database, file and multimedia etc.

(2) Middle ware layer: it maintains the map relation between each data source and XML data, provides the necessary data transmission functions and tools, converts the data into XML format, maintains the map between XML data space and every heterogeneous data source. The client sends request to Web server by browser, the requests is received by Servlet. According to the user's action Servlet calls JSP and JavaBean. JavaBean is responsible for database inter-working. The dynamic XML created by JSP is fed back to the client. On client XSL style sheets handle the XML documents and the handling results are emerged to users in HTML FORM.

(3) Interface layer: it is responsible for the information issue of different application component according to specific protocol or protocol model. The different components may be represented in this layer. The different application components need access XML data space in application level.

(4) Application layer: namely the user interface layer. According to the specific application and user computing environment, adopting appropriate information access technology or application software to complete the corresponding functions.

VI. CONCLUSIONS

With the rapid increasing of Web databases, it is impending to integrate these Web databases and to provide people a unified access to them and achieve information automatically. In this paper, a comprehensive solution for Web database integration is proposed. There are a number of components in the solution, each of them is also a research issue in this area. How to face the future, using rational techniques to realize the heterogeneous database integrated will be a lasting problem. The solution proposed in this paper will bring far-reaching influence on the informatization progress.

REFERENCES

- [1] XIAO Jian-guo. Study on the Integration of Library Heterogeneous Databases Bases on Jva and Hibernate. *Journal of Library and Information Sciences in Agriculture*, 2007,7(7):8-11.
- [2] NIE Li-ping, ZHANG Wei-yong, XU Jin-song. Research on integrating of distributed heterogeneous database systems and its application. *JOURNAL OF HEFEI UNIVERSITY OF TECHNOLOGY*, 2006,3(3):309-312.
- [3] JING Ya-ping. Web-based integration of heterogeneous datasource. *JOURNAL OF GUIZHOU EDUCATIONAL COLLEGE (Natural Sciences)*, 2004,8(4):72-79.
- [4] Chen Feng-yan, Tang Zhen-yu, Bu Zhao-jun. Heterogeneous Database Integration Research on Java and XML. *Journal of Information*, 2006,7:16-21.
- [5] Zhou Ying, Liu Xin Sheng. Resarch of Heterogeneous Database Based on JDBC-Driver. *Computer and Digital Engineer*, 2006,10:65-75
- [6] JIANG Fan. Simply analyze the Inegration of Heterogeneous Databases. *Journal of Chongqing College of Electronic Engineering*, 2009,7(4): 106-108
- [7] HaoFan, HaoGui. Study on Heterogeneous Data Integration Issues in Web Environments. 2007 Intemational Conference on wireless Communications, Networking and Mobile Computing, 2007:3755-3758.
- [8] ZheZhang, MinSong, DaxinLiu etc. Data-Structure-Model for Data Integrationin Distributed Systems. 3rd Intemational Multi-Symposiums on Computer and Computational Sciences, 2008:194-199.