2012 International Conference on Applied Physics and Industrial Engineering

Heterogeneous Database Integration Middleware Based on Web Services

Xianmin Wei

Computer and Communication Engineering School, Weifang University, Weifang, China

Abstract

With the company further accelerating the process of information, the heterogeneity problem of each business system database is more and more salient. How to effectively use these historical data, integration of legacy systems, become a hot spot of integration field. In this paper, how to effectively address heterogeneity of data distribution, to achieve customer access, while maintaining the autonomy of the local system for the objectives of heterogeneous database integration middleware based on Web Services solutions. Applications show that middleware solution has certain feasibility and practicality.

© 2011 Published by Elsevier B.V. Selection and/or peer-review under responsibility of ICAPIE Organization Committee. Open access under CC BY-NC-ND license.

Keywords : Web Services; XML; middleware; database integration

1. Introduction

In the information process, enterprises left behind a lot of application systems based on various business processes and heterogeneous data sources. These systems meet the particular business needs of a enterprise, data has its own characteristics, and other subsystems is difficult to use other sub-system data, generated information “islands”; to impede the process of the enterprise information. Enterprise integration of heterogeneous data sources become an effective way to eliminate information “islands”. To solve data integration in the following problem.

- Heterogeneity, the biggest obstacle of multiple data source systems is to solve the heterogeneity of the various data sources. This heterogeneity has some parts of platform heterogeneity, database systems heterogeneity, semantic heterogeneity.

- Transparency, the greater degree heterogeneous data source, the higher transparency requirements. Transparency including platform transparent, data source transparency and semantic transparency.

- Autonomy, each local database has complete control of their own ability on whether to provide and to determine how much to provide for its function and resource serving for other system. Of the number of its functions and resources to serve other members of the system.
This paper proposed solution for heterogeneous database integration middleware based on Web services to address the above three problems.

2. Related Technologies

2.1. Web Services

Web Services is initiated and defined by W3C (WORLD WIDE WEB CONSORTIUM), currently more popular definition: Web service is a software system marked with URI, using the information of XML format to define and describe the external public interface and bindings. Web services can be discovered by other software systems, and by using the message based XML and IntelNet agreement, in accordance with the manner described by definition of Web services to achieve interact. Web Services have the following features:

- **Intact encapsulation:** for external users, only can see list of features available provided by the Web service object.
- **Loosely coupled:** Within Web services the realization of any changes to the external caller is transparent.
- **Applying protocol normative.**
- **Highly integrated capabilities:** Web services completely shielded difference between different software platforms, each platform can interoperate through standard protocols.

2.2. XML Technology

XML come from SGML (Standard Generalized Markup, standard Generalized Markup Language), with open form of self-described to define data structure, while in the description of data content to highlight the description of the structure, to reflect relationship between the data. XML’s main characteristics are as follows:

- **Scalability:** organizations, individuals can establish their markup for a collection, based on the demand to achieve parametric and semantic corresponding data.
- **Structured:** data storage format not restricted by the display format.
- **Platform independence:** XML document is plain text, independent of platforms and applications.

3. Integration middleware program

Middleware using the manner of Web services to provide interactive way outside, Web application client and other applications client from different platforms, different software development environment called the middleware Web service interface, so get data from different data sources to achieve transparent access to data. The following chart combining middleware framework to explain detailed the core module.
Fig. 1 Detailed core module combining middleware framework

3.1. Describing Core Module

**Middleware Configuration Module**

- configure the current middleware and heterogeneous database connection information: including database addresses, login user name, password, database name. After completing the configuration, current middleware can access through the local sub-query module to get configuration data in the database.
- configure current middleware and other middleware connection information: including the user name and password of connecting other middleware. To complete the configuration middleware can interconnect to form data-sharing network.

**Query Routing Module**

Routing module analyzes a query the user submitted to find out the middleware of the query need routing, and in local middleware configuration information to find out routing information of other middleware, that is the connection information, queries sent to the corresponding middleware, to complete routing.

**Local sub-query Execution Module /Query Result Reorganization Module**

Local sub-query execution module will use the OLE DB technology to realize. The traditional database integration way needs the appropriate wrapper for different databases to complete the query operation. Local sub-query module according to different types of the databases, loading the corresponding OLE DB PROVIDER, achieving a number of different database query.

Query result reorganization module collect the local sub-queries and non-local sub-query results, these results exist through the form of DataSet, and then combining into a large DataSet as the final result. Returning the results to the client to generate XML file under the DataSet to the client, completing the inquiry process.

**Safety Control Module**

Middleware configured to form data sharing network by other middleware, through security control module to configure middleware in data sharing network to share their own data, and data sharing to what extent, in network what middleware can access to their own and so on.

**Query Decomposition Module**

To achieve query efficient decomposition, we define new query syntax of middleware is as follows:
• use the keyword DEFINE, defining the source data unit for the data sheet as the unit, and to [ ] as a data element separator.

Example:
DEFINE [server1.db1.table1 = a] [server2.db2.table2 = b]
Explanation:
The data table table1 as a variable in database db1 of middleware server1. The data table table2 as b variable in database db2 of middleware server2.
• use the keyword SELECT pointing to result set element, [ ] as a cell separator.

Example:
SELECT [a. *] [b.id].
• use the keyword FROM to specify the data unit from the result set, and [ ] as a data element separator.

Example:
FROM [a] [b].
• use the keyword WHERE AND OR to specify the query condition, and [ ] as a condition of cell separator.

Example:
WHERE [a.id = b.id] AND [a.age > b.age]
• use the keyword ADDITION to specify property/database functions/stored procedures of data unit, and [ ] as a cell separator.

Example:
ADDITION [a.id = GROUP BY] [b.age = ORDER BY DESC] [a.age = DISTINCT] [A.age = MAX]
Meaning:
Query process grouped by a.id; by b.age for the results in descending order; in results a.age to sum only for a.age.
After the user submitting a query, the query decomposition module in accordance with the above rules to check the validity of the request and the grammar.
If check is successful, then begun to decompose. Query decomposition in accordance with the following principles:
• independent inquiry split: the middleware necessary as the unit, the query is decomposed into a set of sub-queries, each group of sub-queries corresponding to a single data source.
• in accordance with relevant multi-database query split: two middleware data needs to match the query, first decomposed into a single middleware independent sub-query, check out the results of the independent sub-data to generate subquery matching another middleware data, in order to achieve the connection of two middleware data need to match the query.

Decomposition examples:
DEFINE[server1.db1.table1=a][server2.db2.table2=b][server3.db3.table3=c]
SELECT [a. *][b. *]
FROM [a][b]
WHERE[a.beginTime='2004-9-8'][a.id=c.id][a.id=b.id] ADDITION[a.di=DISTINCT]
Based on decomposition principle, decomposition process is as follows:
• Decomposition into independent sub-queries, and recorded as a new query module variables.
• On the basis of independent sub-query multiple database queries division, each query record the new query variable unit.
DEFINE [SELECT [d. *] FROM [d] WHERE [d.id = c.id] = e]
DEFINE [SELECT [e. *] [b. *] FROM [e][b] WHERE [e.id = b.id] = f]
At query end, f recorded query results.

3.2 Application Description

Domestic each airlines have developed their own aviation information systems, based on Web services for heterogeneous database integration middleware integrated major airlines to provide users with data query services. Described as follows:

- Capital Flight, China Eastern, China Southern Airlines, Sichuan Airlines to install integration middleware respectively, for local configuration, connected to the self aviation information database, and configuration and connection to other integrated middleware, so as to constitute an air information sharing network.
- Client through Web service interface provided by any middleware to submit queries, query through decomposition to route to other middleware on the network, implementation, the final results returned to the client.

Query example: user queries on 2004-9-7 sent from Sichuan and Hainan flights to Beijing, and the arrival time of departure from SichuanFlights earlier than departing from Hainan flights.

The query submitted by user as follows:

Define [server1.db1.info=a][server2.db2.information=b]
Select [a.*][b.*] from [a][b]
And [b.destination=Beijing] And [b.beginTime=2004-9-7]
And [a.endTime<b.endTime]

Explanation:
server1.db1.info=a Sichuan Airlines middleware connected data
server2.db2.information=b China Southern middleware connected data
The query is decomposed into:

Description:
server1.db1.info = a Sichuan Airlines middleware, the data connected
server2.db2.information = b China Southern middleware connected data
Southern middleware connected to the data. The query is decomposed into the following three patrs.

- Take data from 1 and 2 to match data sub-query.

Select [c.] [d.*] from [c] [d] where [c.endTime<d.endTime] the end of the entire query process.

Data sharing network constituted by the middleware has high flexibility, other airlines only need to install the integration middleware, configuration, and share network connection can easily expand the data-sharing network. Aeronautical information and data sharing network diagram is as follows:
4. Conclusion

Data Integration for the realization of EAI, for enterprise internal integration is great importance, this paper proposed a based on Web services heterogeneous database integration middleware solutions. The middleware solution is certainly feasible, practical and capable of heterogeneous database integration.

References