A Middleware for Metadata Management Oriented Distributed Data Sharing

Jiufang An, Faming Lu, Hua Duan, Qingtian Zeng
College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China, *Corresponding author, E-mail: qtzeng@163.com

Abstract—Metadata based data sharing technology has been applied in many fields. The traditional metadata based data sharing technology rarely shares data according to users’ personalized demand. Meanwhile the portability and security of data sharing hasn’t been considered very well. To address these problems, we propose a new metadata based middleware for distributed data sharing in this paper. Compared to some existed data-sharing middleware, the middleware presented in this paper has advantages in portability, personalization and security.

Keywords: Metadata; distributed data; data sharing; metadata management

I. INTRODUCTION

Nowadays, the information silos phenomenon has become a very serious problem. There is an urgent need to share data between distributed heterogeneous data source. Metadata technology plays an important role in promoting data integration, sharing and exchange [1]. How to combine metadata technology with distributed data sharing has become a research hotspot.

Metadata based data sharing technology has been applied in many fields. For example, in neuroimaging area, experts pointed out that neuroimaging data sharing with advanced metadata technology can improve data quality, ensure data security and improve the return of current research investment [2][3]. In the field of geographic information system, metadata architecture and metadata schema specification has been used very well in data sharing project such as the geosciences data clearinghouse (www.geodata.cn) [4][5], which is one of the experiment units of National Science Data-Sharing Program in China. The successful application can also be found in spatial data, scientific data, and ecological field and so on [6]. Although there are lots of successful application, the traditional metadata based data sharing technology rarely share data according to users’ personalized demand. Meanwhile the portability and security of data sharing hasn’t been considered very well. To address these problems, we propose a new metadata based middleware for distributed data sharing in this paper.

II. SYSTEM ARCHITECTURE

The metadata based middleware for distributed data sharing is designed using Model-View-Controller model. The architecture of the middleware is shown in Figure 1. It is composed of three layers, i.e., data-persistence layer, business-logic layer and user layer.

III. IMPLEMENTATION OF KEY TECHNOLOGIES

A. Data Model

Definition 2.1 A 9-tuple D= <dbId, orgId, dbName, displayName, username, psw, drive, ip, port> is called a database metadata model, where:

① dbId: ID of the database used for identification.
② orgId: ID of the department the database belongs to.
③ dbName: the name of the database.
④ displayName: the identity name of the database.
The goal of the algorithm is to get the metadata information of the database, data table and data item, the user can realize data sharing through metadata management.[8][9].

The algorithm consists of two steps. The first step is to extract data table information, and the second step is to get data item information.

Step 1: Extraction Algorithm of Shared Data Table

Input: orgID: ID of the current user’s department.

Output: String of the data table metadata information, formatted as:

\[
\{…\{\text{dbId:id}, \text{nameId:name}, \text{dbItems:[]}\}…\}\}
\]

Begin:

Step 1: Construct the set of all departments Olist \(\{o_1, o_2, …, o_n\}\), where \(o_i\) is an instance of O. And O = \{orgId, orgName\}, where orgId is the ID of a department, orgName is the name of a department.

Step 2: Construct the set of database metadata Dlist for each department in Olist, Dlist = \{d_1, d_2, …, d_n\}, d_i is an instance of D.

Step 3: For each item d_i in Dlist, establish a database connection. Generate the set of data table metadata information Tlist which are shared to the current user, where Tlist = \{t_1, t_2, …, t_i\}, t_i is an instance of T.

Step 4: Construct string S with Olist, Dlist and Tlist as formatted above, output S.

End

Step 2: Extraction Algorithm of Shared Data Item

Input: orgID: ID of the current user’s department.

dbId: ID of the selected data table’s database ;

tableId:ID of the selected data table.

Output: String of the data item metadata information, formatted as:

\[
\{…\{\text{dbId:id}, \text{orgId:name}, \text{dbItems:[]}\}…\}\}
\]

Begin:

Step 1: Construct the set of all departments Olist except for the department, Olist = \{o_1, o_2, …, o_n\}, where \(o_i\) is an instance of O. O = \{orgId, orgName\}.

Step 2: Get the set of database metadata Dlist for each department in Olist, Dlist = \{d_1, d_2, …, d_n\}, d_i is an instance of D.

Step 3: For each item d_i in Dlist, start a connection, get the set of database table metadata information Tlist which have already been opened and shared to the current department, 

Tlist = \{t_1, t_2, …, t_i\}, t_i is an instance of T.

Step 4: Get string S with Olist, Dlist and Tlist as formatted above, output S.

End

IV. TOOL SUPPORT

To verify the previous methods and algorithms, we have developed a metadata middleware tool oriented data sharing on Java platform. The tool supports the sharing of data in SQL Server data base, Oracle data base, MySQL database and so on. The framework of the middleware tool is shown in Figure 2. It can be divided into 3 modules, i.e., user-department configuration module, metadata management module, and resource sharing module.
A. User-department Configuration Module

The module contains functions such as database configuration, department configuration, and user information configuration. It is used to login those database needed to be shared by different departments.

B. Metadata Management Module

The module contains functions such as database metadata management, data table metadata management, and data item metadata management. The module can perform operations on database metadata, data table metadata, and data item metadata, including addition, deletion, and maintenance.

The user can set the name displayed to other organizations and whether or not to open it and so on.

C. Resource Sharing Module

Resource sharing module mainly contains functions such as data table sharing configuration, data item sharing configuration, shared data display.

Data table sharing configuration and data item sharing configuration are used to share specific data tables or data items to specific departments. There are two methods, including configuration based on data tables or data items and configuration based on organizations.

With shared data display, users can not only decide whether or not to hide specific data items shared to them, but also can set the display name and the order of those shared data.

V. APPLICATION CASE

We have adopted the data sharing middleware in one of our software platform “Coalmine Accident Emergency Response System”. The system needs to integrate databases of multiple departments, there are data sharing and exchange between databases. We should follow the following steps for this purpose:

A. User-department Configuration

Login the system with administrator account, and one can configure the information of the system database and the relationship of the user and department. The configuration interface is shown in Figure 3.

B. Metadata Management

Login the system with “user_zongheke” (an account of general department), one can manage the metadata of database, data table and data item, such as configure whether or not to open the data table to other departments and the display name and so on. The data item metadata management interface is shown in Figure 4.

C. Sharing Settings

First, “user_zongheke” shared data table “task Table” to Production Division and Electrical and Mechanical Services Division. And secondly, “user_zongheke” selected the data items “taskName”, “taskContent”, “DeliveryTime”, “deadLine”, “taskPlace” and check the “Share” box to share them to Production Division as shown in Figure 5. Finally, “user_zongheke” shared data item “taskName”, “taskContent”, “FeedbackInfo”, “StartTaskPerson” and “taskPlace” to Electrical and Mechanical Services Division in the same way.

D. Personalized Configuration of Shared Data

Login the system with “user_shengchanke” (an account of Production Division), one can configure those data shared to Production Division based on individual needs. E.g., one can set whether or not to hide the shared data, rename the shared data, set the display order of shared data etc. One personalized configuration is shown in Figure 6. After the configuration, the shared data item “taskname”, “taskcontent”, “TaskPlace”, “DeliveryTime” will be presented to users belonged to Production Division in this order.
After similar operation, one can configure the data shared to Electrical and Mechanical Services Division so that those shared data items are present in the following order, "name", "content", "TaskPlace", "StartTaskPerson" and "FeedBackInfo".

E. Shared Data Display

After those foregoing operation, the user-interface for the Production Division is shown in Figure 7. The user-interface for the Electrical and Mechanical Services Division is shown in Figure 8. The user can select the data table from the tree in the left side and view the data in the right side.

VI. CONCLUSIONS

We have developed a data-sharing middleware based on metadata technology which works properly on one of our software platforms--“Coalmine Accident Emergency Response System”. Compared to some existed data-sharing middleware, the metadata-based middleware presented in this paper has the following characteristics:

1. Portability. The middleware can work in distributed data system soon after simple configuration.

2. Personalization. Users can configure those data shared to his according to its specific. E.g., renaming those shared data item.

3. Security. Users can set the shared name of data table or data item to avoid exposing private information of the data to others.

In the future, we will work to improve the efficiency of the metadata extraction algorithm and the shared data extraction algorithm. Moreover, we will devote ourselves to making the middleware easier to use.

ACKNOWLEDGMENT

This paper is supported partly by the NSFC (61170079 and 61202152); Specialized Research Fund for the Doctoral Program of Higher Education of China (20103718110007) and Special Fund for Fast Sharing of Science Paper in Net Era by CSTIC(2012107), Sci. & Tech. Development Fund of Qingdao(10-3-3-32-shn and 2011-2-47), Excellent Young Scientist Foundation of Shandong Province (BS2009DX004); China Postdoctoral Science Foundation (2011MS501155); and Natural Science Foundation for Distinguished Young Scholars of Shandong and SDUST (2010KYJQ101).

REFERENCES