

# A Middleware for Metadata Management Oriented Distributed Data Sharing

Jiufang An, Faming Lu, Hua Duan, Qingtian Zeng<sup>+</sup>

College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China,

<sup>+</sup>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<sup>[1]</sup>. 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<sup>[2][3]</sup>. 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]<sup>[5]</sup>, 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<sup>[6]</sup>. 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.

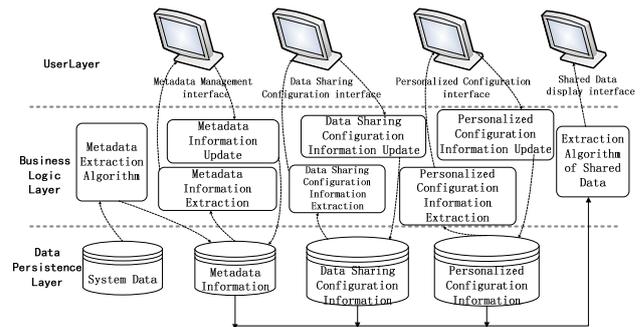


Figure 1. System architecture

Data-persistence layer is used to manage the system data which may be stored in distributed and heterogeneous databases. In addition, it manages the pre-treatment metadata information, the configuration information of distributed data sharing, and the configuration information about the display of shared data.

Business-logic layer realizes the following functions related to metadata management and distributed data sharing: ① Pretreatment function. It is used to extract the metadata information from the system using the middleware, including database metadata information, data table metadata information and data item metadata information. ② Metadata management function. Users can maintain the metadata information of their own data with the function. ③ Data sharing configuration function. Users can configure the share property of their shared data. ④ personalized configuration of shared data. Users can have personalized adjustment for the display of their own data or the shared data of other departments. ⑤ Shared data extraction algorithm. Users can extract shared data and display it according to their personalized configuration.

User view layer is composed of four functional interfaces, which are metadata management interface, data sharing configuration interface, data display configuration interface and shared data display interface.

## III. IMPLEMENTATION OF KEY TECHNOLOGIES

### A. Data Model

Definition 2.1 A 9-tuple  $D = \langle dbId, orgId, databaseName, displayName, username, psw, drive, ip, port \rangle$  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 databaseName: the name of the database.
- ④ displayName: the identity name of the database.

- ⑤ username: the user name access to the database.
- ⑥ psw: the password access to the database.
- ⑦ drive: the drive type of the database.
- ⑧ ip: the address access to the database.
- ⑨ port: the port of the database.

Definition 2.2 A 7-tuples  $T = \langle \text{tableId, tableName, displayName, nameForOthers, keywords, description, isopen} \rangle$  is called an data table metadata model, where:

- ① tableId: ID of the table used for identification.
- ② tableName: the name of the data table.
- ③ displayName: the identity name of the data table.
- ④ nameForOthers: the name shown to other departments
- ⑤ keywords: the keywords to describe the data table.
- ⑥ description: the description of the data table.
- ⑦ isopen: whether or not to open to other departments.

Definition 2.3 An 8-tuples  $I = \langle \text{itemId, tableName, itemName, displayName, nameForOthers, description, isopen, isquery} \rangle$  is called a Data item metadata model, where:

- ① itemId: ID of the item used for identification.
- ② tableName: name of the table the database belongs to.
- ③ itemName: the name of the data item.
- ④ displayName: the identity name of the data table.
- ⑤ nameForOthers: the name shown to other departments.
- ⑥ description: the description of the data item.
- ⑦ isopen: whether or not to open to other departments.
- ⑧ isquery: whether to let the item be a query condition.

### B. Metadata Extraction Algorithm

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<sup>[8][9]</sup>.

Input: A database metadata information  $d = \langle \text{dbId, orgID, name, display, user, password, driver, Ip, port} \rangle$

Output: TRUE or FALSE

Begin:

Step1: Try to connect database d. If it fails, return FALSE. Else jump to Step2.

Step2: Construct the set of all tables Tlist in d by a sql command,  $Tlist = \{t_1, t_2, \dots, t_n\}$ , where  $t_i$  is an instance of T. Let  $t_i.displayName = t_i.tableName$ ,  $t_i.nameForOthers = t_i.tableName$ ,  $t_i.description = ""$  and  $t_i.isopen = false$ . If the operation fails, return FALSE. Else jump to Step3.

Step3: Construct the set of table items Ilist in each table of Tlist,  $Ilist = \{I_1, I_2, \dots, I_n\}$ , where  $I_i$  is an instance of I. Let  $I_i.displayName = I_i.itemName$ ,  $I_i.nameForOthers = I_i.itemName$ ,  $I_i.description = ""$ ,  $I_i.isopen = false$  and  $I_i.isquery = false$ . If the operation fails, output FALSE, else jump to Step4.

Step4: Write and store the information of d, Tlist and Ilist to database. If the operation fails, return FALSE. Else return TRUE.

End

### C. Extraction Algorithm of Shared Data

The goal of the algorithm is to get the shared data and display on the shared data display interface.

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.

#### 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:

```
{...{orgId:idi, orgName:namei, dbItems:{...
{dbId:dbIdi, dbName:dbNamei, tableItems:{...
{tableId:tableIdi, tableName:tableNamei}...}}...}}
```

Begin:

Step1: Construct the set of all departments Olist  $Olist = \{o_1, o_2, \dots, o_i, \dots\}$ ,  $o_i$  is an instance of O. And  $O = \{\text{orgId, orgName}\}$ , where orgId is the ID of a department, orgName is the name of a department.

Step2: Construct the set of database metadata Dlist for each department in Olist,  $Dlist = \{d_1, d_2, \dots, d_i, \dots\}$ ,  $d_i$  is an instance of D.

Step3: 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  $Tlist = \{t_1, t_2, \dots, t_i, \dots\}$ ,  $t_i$  is an instance of T.

Step4: Construct string S with Olist, Dlist and Tlist as formatted above, output S.

End

#### ② Extraction Algorithm of Shared Data Item

Input: orgID: ID of the current users' 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:

```
{...{orgId:idi, orgName:namei, dbItems:{...
{dbId:dbIdi, dbName:dbNamei, tableItems:{...
{tableId:tableIdi, tableName:tableNamei}...}}...}}
```

Begin:

Step1: Construct the set of all departments Olist except for the department,  $Olist = \{o_1, o_2, \dots, o_i, \dots\}$ ,  $o_i$  is an instance of O,  $O = \{\text{orgId, orgName}\}$ .

Step2: Get the set of database metadata Dlist for each department in Olist,  $Dlist = \{d_1, d_2, \dots, d_i, \dots\}$ ,  $d_i$  is an instance of D.

Step3: For each item  $d_i$  in Dlist, start a connection, get the set of data table metadata information Tlist which have already been opened and shared to the current department,  $Tlist = \{t_1, t_2, \dots, t_i, \dots\}$ ,  $t_i$  is an instance of T.

Step4: 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.

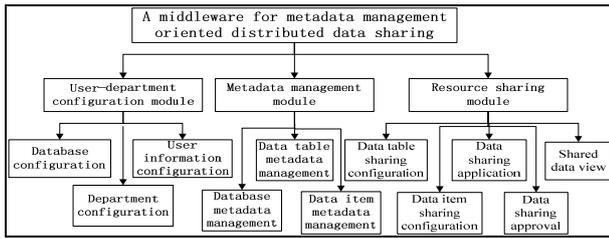


Figure 2. Function modules of the middleware

**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 data base 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**



Figure 3. Database 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.

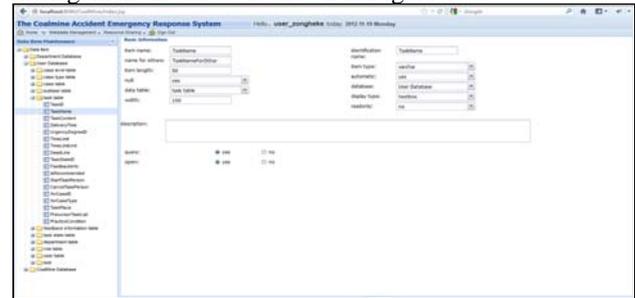


Figure 4. Data item metadata management

**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 items “taskName”, “taskContent”, “FeedbackInfo”, “StartTaskPerson” and “taskPlace” to Electrical and Mechanical Services Division in the same way.

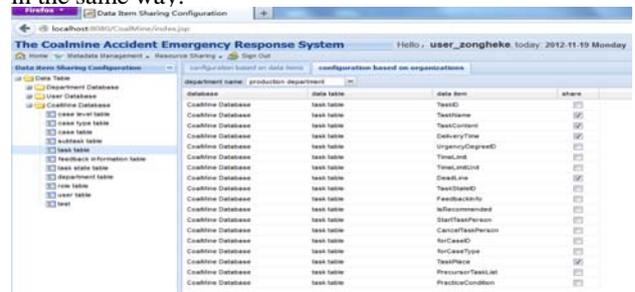


Figure 5. Data item sharing configuration

**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.

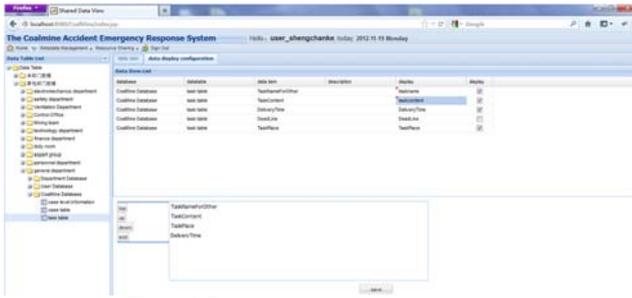


Figure 6. Personalized configuration of shared data

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.

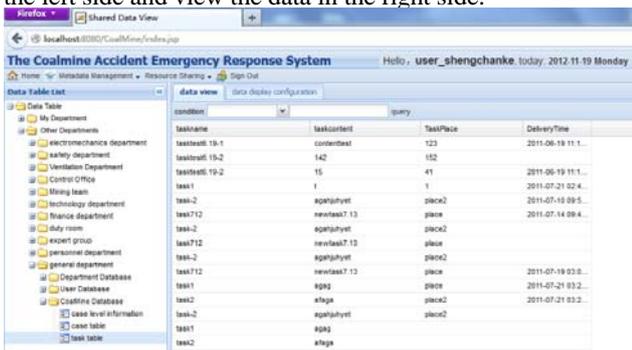


Figure 7. The user-interface for the Production Division

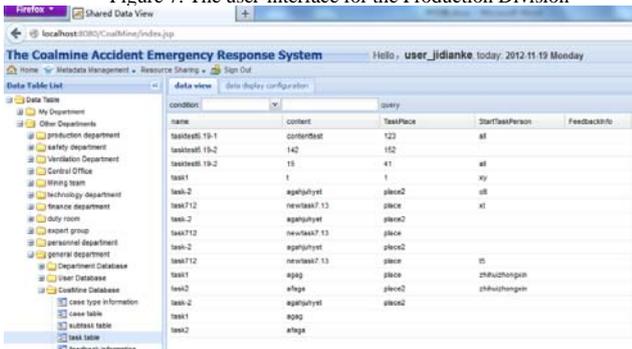


Figure 8. The user-interface for the Electrical and Mechanical Services Division

## 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 CSTD(2012107), Sci. & Tech. Development Fund of Qingdao(10-3-3-32-nsh and 2011-2-47), Excellent Young Scientist Foundation of Shandong Province (BS2009DX004); China Postdoctoral Science Foundation (2011M501155); and Natural Science Foundation for Distinguished Young Scholars of Shandong and SDUST (2010KYJQ101).

## REFERENCES

- [1] Erik Duval “Metadata standards: What, who & why,” Journal of Universal Computer Science, vol.7, pp.591-601, 2001.
- [2] Jean-Baptiste Poline, Janis L. Breeze, Satrajit Ghosh, Krzysztof Gorgolewski, Yaroslav O. Halchenko, Michael Hanke, Christian Haselgrove, Karl G. Helmer, David B. Keator, Daniel S. Marcus, Russell A. Poldrack, Yannick Schwartz, John Ashburner, David N. Kennedy “Data sharing in neuroimaging research,” Front Neuroinform, vol.6, 2012.
- [3] Mennes, Maarten, Biswal, Bharat, Castellanos, F. Xavier, Milham, Michael P. “Making data sharing work: The FCP/INDi experience,” NeuroImage, 2012.
- [4] Wang Juan-le, YOU Song-cai, XIE chuan-jie “Analysis and Design of Metadata Standard Structure for Geosciences Data Sharing,” Geography and Geo-Information Science, 2005.
- [5] Jane Kaye, Catherine Heeney, Naomi Hawkins, Jantina de Vries, Paula “Data sharing in genomics-resaping scientific practice,” Nature Reviews Genetics 10, pp.331-335, 2009.
- [6] Tschangho John Kim “Metadata for geo-spatial data sharing: A comparative analysis,” The Annals of Regional Science, vol.33, pp.171-181, 1999.
- [7] Erik Duval, Wayne Hodgins, Stuart Sutton, Stuart L. “Metadata Principles and Practicalities,” D-Lib Magazine, vol. 8, 2002.
- [8] Tingting Chen, Qingtian Zeng, Faming Lu, Jian Sun, Jie Zou “A Visual Management and Monitoring Tool for Cross-organization Emergency Response Workflows” Information Technology Journal, 2012.