Research on Heterogeneous Data Integration Technology for Personal Public Utility Bills Management Platform

Jiayun Pan, Yinghua Xie and Zhuan Song
Department of Information Science and Technology, University of Donghua, Shanghai, China
panjyhoho@126.com, yh_xie@dhu.edu.cn

Abstract - At present public utility bills systems are independent of each other and data in these systems are heterogeneous. In view of this issue, this paper combines the middleware and Web Service technology to come up with an integrated management platform to integrate heterogeneous data and query public utility bills within one system. After introducing the transformation method between XML schema and relational data models, it uses an XML Schema model as an integrated system framework, transforms the heterogeneous data into a unified XML format. The transformed data will present a unified view in virtual database and provide a consistent view to the application layer, making the upper applications easily access the service. This integration method can shield difference in data structures, make the heterogeneous data sources become transparent to realize the dynamic integration of heterogeneous data in a web environment. This system implements online query, billing, online payment and other functions to achieve the unification of individual utilities information, standardization and shared services.

Index Terms - Heterogeneous data integration, XML, Mapping, Web Service, Middleware.

I. Introduction

Under the background of smart city, the advancement of informatization makes the utility systems achieve the function of online query and payment. But each system uses different development platforms, and data in these systems are stored in different structures. It causes the “information isolated island” which means formation in different systems cannot be shared and exchanged. The “information isolated island” hinders the informatization process and brings inconvenience to users so the best way to eliminate it is heterogeneous data integration. So it’s necessary to build an integrated management platform to integrate heterogeneous data from different utility systems, so that the users can use a unified interface to query comprehensive information.

This paper combines the middleware and Web Service technology to come up with a personal public utility bills management platform to integrate heterogeneous data and query public utility bills within one system. It not only achieves the unification of individual utilities information, standardization and shared services but also conducive to the government on more specification for utility data management.

II. Heterogeneous Data Integration Technology

Heterogeneous data integration technology shields heterogeneous underlying data source and provides a unified interface for users [1]. Users can use unified query interface to achieve access to heterogeneous data sources, realize the integrated application of data and information without considering the different models, categories and geographic location of underlying data.

At present there are a variety of methods to achieve heterogeneous data integration system. They have different emphasis on solving the problem of data sharing.

A. Data Warehouse

Data warehouse will preprocess and transform data which are copied from different sources according to a centralized and unified view to comply with the data warehouse model, and store these transformed data in it. But when data sources change, it should make corresponding modification of data in data warehouse, this will cause the data update not in time and duplicate storage.

B. Middleware

Middleware is similar to data warehouse, but they have a completely different structure. Data in middleware are still stored in different heterogeneous data source, and the integrated system only provides a virtual integration view and the view of query processing mechanism. It do not need to store large amounts of data repeatedly, the data are still stored in each local data source, It is widely used in data integration because it has the characteristics of good real-time, strong autonomy and suitable for mass data integration.

C. Web Service

Web Service is a modular application can describe, publish, locate and invoke through the web. It’s loosely coupled and can locate other service components on the Internet dynamically and interact with them. Its platform independence and the way of data description based on XML (Extensible Markup Language) schema can use standard methods to expose the data from different platforms or software application without considering their specific data formats and database management system. Putting the data type conversion in Web Service can easily achieve interoperability, greatly reduced the time and cost of data integration [2]. From the perspective of the characteristics of Web Service technology, it provides a good solution to integrate heterogeneous data under the network environment.

Considering the data integration pattern characteristic and the requirements of personal public utility bills management platform, this paper combines the middleware and Web Service technology to come up with an integrated management platform to integrate heterogeneous data. It uses an XML

Schema model as an integrated system framework, a Web Service as a wrapper of heterogeneous data source, transforms the heterogeneous data into a unified XML format, making the heterogeneous data sources become transparent to realize the dynamic integration of heterogeneous data in a web environment.

III. Mapping between XML Schema and Relational Schema

This article first discusses the mapping between XML schema and relational data model. It only discusses data in different relational database as heterogeneous data.

Data in Relational database are structured, whole in XML document are semi-structured. So to realize conversion between relational data and XML data, the key problem is how to establish a mapping between the two structures, and keep the relevant constraint information. There are two main methods of mapping between relational schema and XML schema: template-based mapping and model-based mapping [3].

Template-based mapping only need to embed the appropriate SQL statement in XML documents, replaces the result to the corresponding location to get the target XML document. Its advantage is simple and flexible, the disadvantage is that it only allow transform relational database to an XML document, and can't reverse transform.

Model-based mapping build a concrete model to realize the bidirectional conversion between relational schema and XML schema without SQL statement. Table-based model is one of the model-based mapping methods. It takes an XML document as a collection of one or more tables in a relational database. The corresponding XML document structure is as follows:

```
 <RDBHS>
   <Database>
     <Table>
       <Row>
         <Column1>……</Column1>
         <Column2>……</Column2>
         ……
       </Row>
       ……
     </Table>
     ……
   </Database>
   ……
</RDBMS>
```

A. Relational Schema to XML Schema

Transforming RDBMS data into an XML document has three steps: relational schema information extraction, model transformation, XML documents generation [4]. First of all, the applications use the mechanism provided in RDBMS to extract information from database in table structure, data type and semantic constraints aspects. Secondly, in the schema transformation stage, the relational schema information transform into the corresponding XML schema information according to the mapping rule. Finally, using the XML DOM or SAX programming interface to make the XML schema written to XML documents.

The following is an example of converting relational table data into an XML document.

Table I is a two-dimensional table records persona water bill information. As space is limited, it reduces the table to four columns, which are customer id (Cus_id), date of bill (Date), usage of water (usage) and total water fee (Total_fee), and it lists only two records.

Following is the pivotal code for transforming Table I to an XML document (waterbills.xml) with ASP.NET [5].

```
//Create and add the root element waterbills
XmlElement
xe_waterbills=xdoc.CreateElement(“waterbills”);
xdoc.AppendChild(xe_waterbills);
// Read each row in the database and add to the XML
document
for (int i=0;i<dtblWaterbill.Rows.Count;i++)
{
XmlElement
xe_waterbill=xdoc.CreateElement(“waterbill”);
//add “Cus_id”, the child element of “waterbill”
XmlElement xe_Cus_id=xdoc.CreateElement(“Cus_id”);
xe_Cus_id.InnerText=dtblWaterbill.Rows[i][“Cus_id”].T
oString();
xe_waterbill.AppendChild(xe_Cus_id);
// add other child elements of “waterbill”
……
//add “waterbill” as the child element of “waterbills”
xdoc.DocumnetElement.AppendChild(xe_waterbill);
}
```

Following is the XML document based on the code.

```
< waterbills>
   < waterbill>
      < Cus_id >1000001< /Cus_id>
      < Date >2013-01< /Date >
      < Usage >28< /Usage>
      < Total_fee >51.8< /Total_fee>
   </ waterbill>
   < waterbill>
      < Cus_id >1000002< /Cus_id>
      < Date >2013-02< /Date >
      < Usage >32< /Usage>
      < Total_fee >59.2< /Total_fee>
   </ waterbill>
< waterbills>
```

TABLE I Waterbills

Cus_id	Date	Usage	Total_fee
1000001	2013-01	28	51.8
1000002	2013-02	32	59.2

B. XML Schema to Relational Schema

Transforming an XML document into RDBMS data has two steps: XML document parsing, create tables and constraints [4]. DOM and SAX are programming interface of XML, they are used for parsing. If use a DOM parser to parse an XML document, program will first parsing the XML document into a DOM tree. Then according to certain rules to determine which node of the tree should be changed into RDBMS tables, which nodes should be converted to RDBMS table columns, and depending on the type of node's properties to determine some constraints in the RDBMS, and finally create the information to the RDBMS.

Following is the DOM tree of the XML document “waterbills.xml”, as shown in Fig.1.

In a DOM tree, the logical structure of the document likes a tree. The document root, elements, element content, attributes and attribute values are in the form of object model. The model not only describes the structure of the document, but also defines the formation properties and behaviour of the model.

Following is the pivotal code for transforming a XML document to a table with ASP.NET [5].

```
// Extract each node information of XML document,
inserted them into the database
// “waterbill” node
XmlNode xn_waterbill=xe_waterbills.ChildNodes[i];
// “Cus_id” node
String strCus_id=xn_waterbill.ChildNodes[0].InnerText
// “Date”,”Usage”,”Total_fee” node
……
//Create a new row in table”Waterbills” DateRow
drowWaterbill=dtblWaterbill.NewRow();
drowWaterbill[“Cus_id”]=strCus_id;
// “Date”,”Usage”,”Total_fee” node
……
dtblWaterbill.Rows.Add(drowWaterbill)
```

This code can transforming the XML document (waterbills.xml) to Table I.

IV . System Design

A. Design Objective and Design Concept

Each public utility system has different regions branches and they may use different development platforms and data storage format, so the design objective of heterogeneous data integration system is as follows [6].

1) It can integrate heterogeneous data from different physical locations and different database and output in a unified interface.

2) It can access heterogeneous data resources transparently and efficiently, eliminate the “information isolated island” and share information resources.

3) It can keep the autonomy of each system without changing the way of original data storage and management.

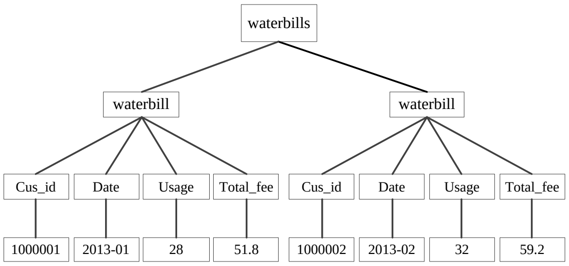

Fig. 1 DOM tree of waterbills.xml

The design concept is transforming the heterogeneous data sources into unified XML format with the help of Web Service package. The transformed data will present a unified view in virtual database and provide a consistent view to the application layer. Users in application layer can easily access the service and query the information according to their actual needs.

B. Function Designs

According to actual user requirements, the system will implement the following four functions, as shown in Fig.2.

1) Query: Users can query utility bills including water, electricity, gas, telephone, broadband network, cable TV bills. And users can do a detailed query of each kind of utility bills.

2) Bill: Users can statistics all the utility bills situation of a certain time period. It can help the users to overview the different kinds of utility bills and is conducive to personal finance.

3) Payment: The users can pay the bills online according to the current utility fees.

4) Print: Print all kinds of utility bills when needed.

C. Query Process

According to the situation of heterogeneous data and user requirements, the flow chart of Query Process is shown in Fig.3.

1) It Receive user requests from application.

2) Request processing module transform the request format into XML format.

3) Virtual database receives XML format of the request, divide the global query into sever sub queries to find the corresponding target data according to the mapping rule.

4) Target Data are XML format data which are transformed from heterogeneous data sources associated with the queries.

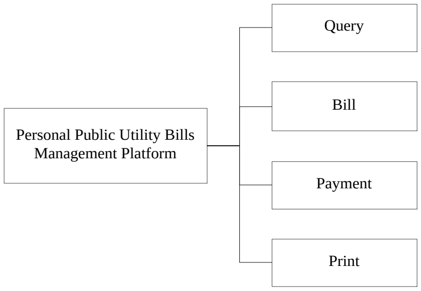

Fig. 2 Function of personal public utility bills management platform

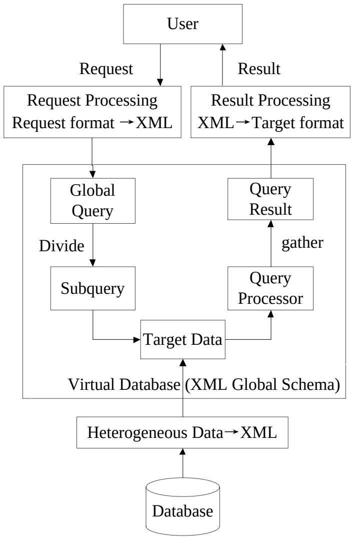

Fig. 3 flow chart of query

5) Gather the query results and send to results processing module.

6) Results processing module transform the XML format into target format.

7) Application will show the results to the user in a particular interface.

D. Logical Level

Based on the above process, the whole system design for three logical levels, they are data layer, middleware layer and application layer (from bottom to top), as shown in Fig.4.

1) Data Layer: Data layer are provide data for system and in the bottom of the system. Data sources are utility systems in different data structure, they can be a variety of structured data, such as Oracle, SQL Server, Access and other data in a relational database, also can be a semi-structured or unstructured data.

2) Middleware Layer: Middleware layer has two parts, the data source wrapper and a virtual database [7]. In this paper, the data source wrapper is Web Service. It can transform the heterogeneous data into a unified XML format and provide a consistent view to the application layer. Create a Web Service for each of the data source, and then use the WSDL to the registration Service centre. When used to build a new integration, integration side firstly send search requests to the registry to collect and select the appropriate data source, and then get the data from the data source through the SOAP protocol. The transformed data will present a unified view in virtual database, and the virtual database does not provide specific physical space, its main function is integrating a variety of heterogeneous data sources through the data source wrapper and providing a global data model based on XML which makes the upper applications to access the service easily.

3) Application Layer: Application layer is the user interface, including Web browser, application program and other various customer applications. It can select targeted access mode and provide query interfaces to users according to the actual needs of users.

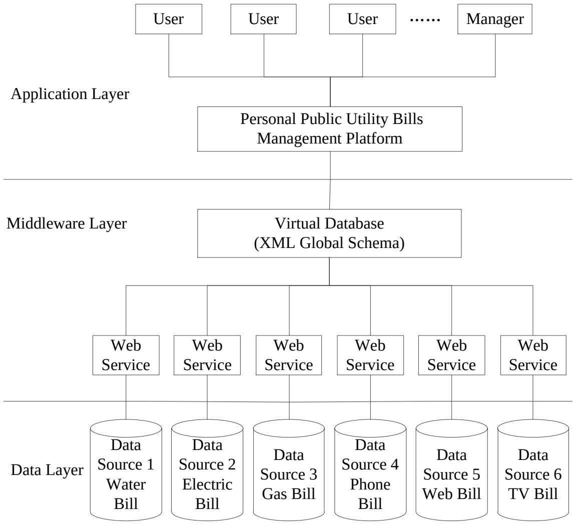

Fig. 4 Structure of personal public utility bills management platform

V. Summary

This paper combines the middleware and Web Service technology to come up with a method to integrate heterogeneous data in Personal Public Utility Bills Management Platform. This method shields difference in data structures, makes the heterogeneous data sources become transparent and realizes the dynamic integration of heterogeneous data in a web environment.

This platform provides multiple functions and helps the users to overview the different kinds of utility bills. It's very convenient for the management and operation of utility bills for both individuals and government.

This system has strong scalability and compatibility because of the unified XML format, and it's convenient to perfect the system and extend the function of platform in the future.

References

[1] Zhu Rongrong , "Research on heterogeneous data integration and information sharing mode," thesis for master degree of Xi'an University of Science and Technology, pp.1-15,June 2010.

[2] Su Guohui, Sun Jihong, "A method reduce sea bottom reverberations based on predictive deconvolution," Marine Geology Frontiers, vol.27, no.11, pp.51-54, November 2011.

[3] Zhang Huijun, "Research on data integration technology for E-commerce in the web environment, " thesis for master degree of North China University of Technology, pp.18-20, June 2012.

[4] Qian Lei, "Research and implement of XML-based data transformation for heterogenous database," thesis for master degree of ShenYang Ligong University, pp.19-25, March 2010.

[5] Wu Jie, "XML application guide," Tsinghua University Press, pp.150-162, May 2007.

[6] Liu Yi, Liu Jun, Mao Chenlin, "Design of database integration of grid online monitoring system based on middleware," Power System and Clean Energy, vol .28, no. 5, p.41, May 2012.

[7] Zhang Lei, "On heterogeneous data integration based on XML," Journal of BeiHua University (Natural Science), vol.6, no.4, pp.383-384, August 2005.