Research on An XML-based Metadata for Ancient Chinese Architecture

Jing Wan, **Ze Yuan**, **Huiming Luo**, **Junkai Yi**
Department of Computer Science and Technology, Beijing University of Chemical Technology, Beijing, 100029, China
wanj@mail.buct.edu.cn

Abstract - Ancient Chinese architecture was digitalized preserved in the projects funded by the government in the past few years. Many application systems have been built in order to use these digital resources effectively. In this paper, we investigate a practical framework for data exchange and sharing among heterogeneous databases of these systems. An XML-based metadata is proposed as a standard for data representation and exchange. And a data exchange model is presented. An application system using the proposed model has been developed. It has been proved that the model works well.

Index Terms - Ancient Chinese architecture, Data exchange, XML, Heterogeneous database, Metadata

I. Introduction

Ancient Chinese architecture is an important part of Chinese civilization, having an indigenous and unique system of construction. In order to preserve this architectural heritage, a large number of application systems have been built through projected supported by the government. Most of these systems are heterogeneous, which means that their system platforms, databases and development tools are inconsistent and incompatible. Due to the different data formats and different database structures, it is difficult to achieve data exchange and sharing among these systems. There is an urgent need to develop the approaches that can facilitate heterogeneous database access in this domain.

Data exchange methodologies among heterogeneous database include middle database technology, two-way middleware technology, data warehouse technology, electronic data interchange technology, and common manipulation language technology [1-4]. The problems in applying these technologies to specific domains are implementation difficulties, low security, and poor interoperability. XML (Extensible Markup Language) has been widely used for the representation of arbitrary data structures since it was put forward by W3C (World Wide Web Consortium) in February 1998. It has many advantages, such as extensibility, openness, self description and cross platform, which make it an effective method of information representation to achieve data exchange among different systems [5-8]. In this paper, we design an XML schema for ancient Chinese architecture, which is later utilized as the unifying data model for data exchange. We also propose a data exchange model based on the metadata.

II. XML-based metadata

A. Data structure

Data structure of ancient Chinese architecture is organized as a tree, as shown in Fig. 1, where the rectangles denote data elements. The root is "Ancient Chinese architecture" node. The data element of an upper level node is composed of that of its child nodes. For example, the root node has 6 child nodes, which means the root data's composition is as follows: "basic data", "single building data", "restoration data", "affiliated relic data", "disease data", and "research data". The symbol located in the upper left corner of the rectangle indicates the repetitions of the data element. An asterisk denotes data element repeated 0 to n times, and number "1" denotes the repetition of 1 time. If there is no symbol, it denotes the repetition of 0 or 1 time.

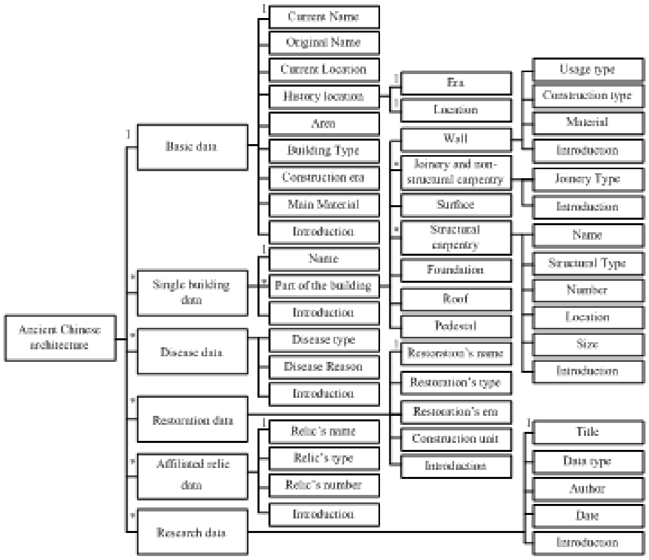


Fig. 1 Data structure of ancient Chinese architecture

B. XML schema

XML Schema specifies XSD (XML Schema definition language), which offers facilities for describing the structure and constraining the contents of XML documents. It has many advantages over DTD (Document Type Definition), including consistency, scalability, interoperability, and standardization [9]. XSD is far more powerful than DTD in describing XML languages. Thus, we use XSD to describe the metadata of ancient Chinese architecture information.

C. Metadata design

According to the data structure given above, we use XSD to design an XML schema as ancient Chinese architecture metadata (ACAM). It is used as a standard for data exchange. The information transferred between systems must be in the form of XML document in accordance with the standard.

The standard defines the legal building elements of an XML document. There is a one-to-one correspondence between the legal elements and the nodes in the data structure tree. In the schema file, each defined element corresponds to a tree node, and element definitions are listed from top to bottom and left to right. First, the element “Ancient Chinese Architecture” corresponding to the root node of the tree is defined at the beginning of the file. And then the nodes at the second level are defined, and so forth. The metadata design includes complex element design, and simple element design.

In an XML schema file, the complex elements are often used to control the high-level abstract data structures. They contain other elements and/or attributes. In our design, the complex elements correspond to non-leaf nodes of the tree. The <xs:complexType> tag is used to define complex element. The <xs:sequence> tag specifies that the child elements must appear in a specific order. The <minOcurrs> and <maxOccurs> indicators are used to define how often an element can occur, corresponding to the symbol in the tree structure. The <ref> indicator is used to define the nested complex elements.

In an XML schema file, the simple elements are often used to control the low-level data structures. A simple element contains only text, and not any other elements or attributes. In our design, the simple elements correspond to leaf nodes of the tree. The text of a simple element can be one of the built-in data types included in the XML Schema definition. The most commonly used built-in types are: string, decimal, integer, boolean, date, and time.

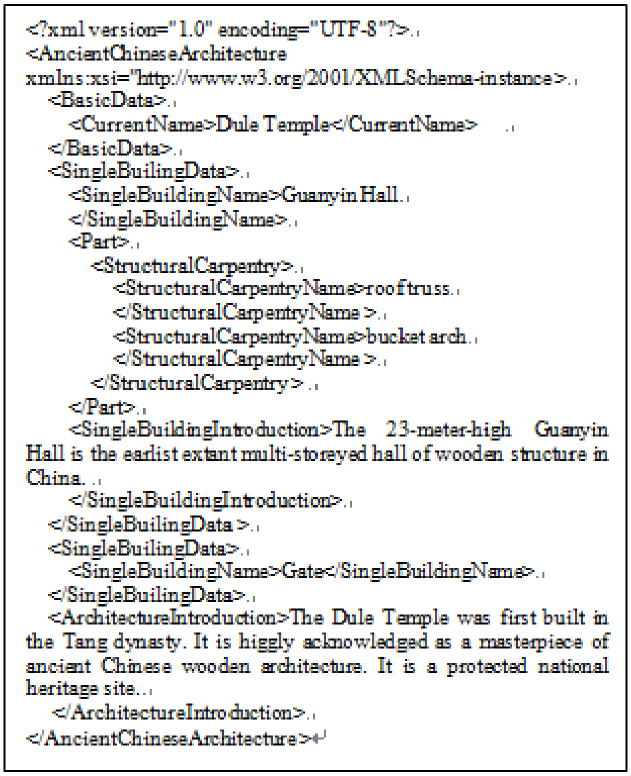
The example using built-in data type is shown below.

```
<xs:element name="Name"  type="xs:string"/>
```

The built-in data types cannot meet all of our requirements. For example, the element “RestorationType” has only four acceptable values. We cannot use the built-in data type “string” to define it. The Schema provides a mechanism to define custom simple data types by applying constraint to basic types. It provides 15 constraints, such as pattern constraint, enumeration constraint, whiteSpace constraint, and length constraint. For example, the enumeration constraint can be used to limit the content of an XML element to a set of acceptable values.

D. *XML document*

The data transferred among systems should be converted to an XML document that conforms to ACAM schema. An example of XML document is illustrated in Fig. 2. Because the example deals with Chinese architecture, some contents are in Chinese.



```
<?xml version="1.0" encoding="UTF-8"?>
<AncientChineseArchitecture
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance>
   <BasicData>
      <CurrentName>Dule Temple</CurrentName>
   </BasicData>
   <SingleBuilingData>
      <SingleBuildingName>Guanyin Hall
      </SingleBuildingName>
      <Part>
         <StructuralCarpentry>
            <StructuralCarpentryName>roof truss
            </StructuralCarpentryName >
            <StructuralCarpentryName>bucket arch
            </StructuralCarpentryName >
         </StructuralCarpentry >
      </Part>
      <SingleBuildingIntroduction>The 23-meter-high Guanyin
Hall is the earlist extant multi-storeyed hall of wooden structure in
China.
      </SingleBuildingIntroduction>
   </SingleBuilingData >
   <SingleBuilingData>
      <SingleBuildingName>Gate</SingleBuildingName>
   </SingleBuilingData>
   <ArchitectureIntroduction>The Dule Temple was first built in
the Tang dynasty. It is higgly acknowledged as a masterpiece of
ancient Chinese wooden architecture. It is a protected national
heritage site.
    </ArchitectureIntroduction>
</AncientChineseArchitecture>
```

Fig. 2 A sample XML document

III. Data exchange model

A. *ACAM-based exchange model*

Data of ancient architecture are located in heterogeneous relational database of the application systems. ACAM is used as a standard language to exchange data. The ACAM-based data exchange model is illustrated in Fig. 3. The exchange procedure involves the following steps: first, database records are extracted from the source database, and are converted to an XML document according to ACCM by Generator; second, the generated xml document is sent to the target system; third, the xml document is parsed and converted to data records of the target database according to ACCM by Parser.

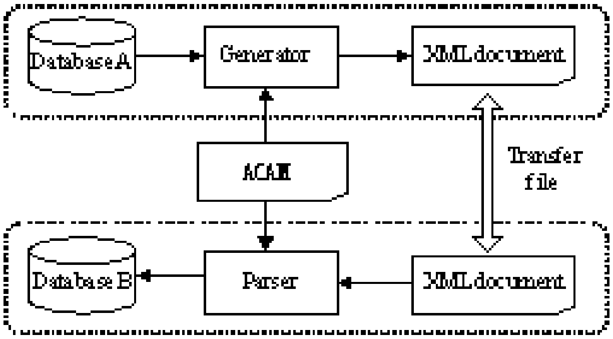


Fig. 3 ACAM-based data exchange model

B. *Model implementation scheme*

To realize the model defined above, the key issue is the implementation of Generator and Parser. JAVA is a widely used web programming language, and is able to develop cross-platform application. By using JDBC, JAVA can access heterogeneous database, such as Oracle, MySql and SQL Server. Therefore, we choose Java as the implementation language.

Generator is used to map relational database records onto an XML document. The process has the following steps: first, the relational-to-XML mapping rules are configured; second, according to the above rules, SQL statements are generated; third, the statements are committed, and data results are organized as an XML document conforming to ACAM.

Parser is used to map an XML document onto relational database records according to the following steps: first, the XML-to-relational mapping rules are configured; second, based on ACAM, the received XML document is analyzed, and data elements are extracted from the file; third, according to the mapping rules, the extracted data elements are converted to data records of the local database.

C. *Mapping rules*

The mapping between XML and relational schema can be divided into two categories: template-based mapping and model-based mapping [10-14]. Compared with the template-based mapping, the model-based mapping is bidirectional mapping, using both sides. We choose model-driven mapping method. Mapping rules are as follows.

1) Database to XML mapping:

Table names are mapped to the names of complex type element; column names are mapped to the names of simple type elements; column values are mapped to type property values; foreign keys association are mapped to the hierarchical relationships between elements..

2) XML to database mapping:

Simple type element: An element is mapped to a column of its parent's table; element type is converted to column type; constraints are converted to the column's attributes.

Complex type element: From bottom to top, an element is mapped to a table, and the primary key of its parent element's table becomes the foreign key of the child element's table. If the element mostly occurs for 1 time, the corresponding table is merged into the parent table.

IV. An application case

We apply the model to an application system named ancient architecture protection platform (AAPP). AAPP includes 5 subsystems, which are the knowledge base subsystem (KBS), the simulation repair subsystem (SRS), the scene animation subsystem (SAS), the three-dimensional modeling subsystem (TMS), and the encyclopedia subsystem (ES). The subsystems are built by different research organizations. The databases involved in the platform are SQL Server2000, SQL Server2005, and MySql. In AAPP, every subsystem has the same Generator and Parser. And each subsystem should configure its mapping file. Thus, Generator and Parser can use custom mapping files to realize different data mappings.

In AAPP, the data exchange steps are as follows:

1) A subsystem submits data exchange request.

2) Controller parses the request, and sends it to subsystems providing data.

3) The subsystems receive the request, use Generator to extract data from local database to an xml document, and send it to the controller.

4) Controller obtains all xml documents from subsystems, validates the files, and sends them to the subsystem that submits the request.

5) The subsystem receives XML documents, uses Parser to analyze files, and converts data to database records.

V. Conclusion

Based on the analysis of the ancient architecture data structure, an XML-based metadata is designed, and the data exchange model based on it is proposed. We also discuss its application in data exchange. The application system built in this study shows that the XML-based data representation standard and the model provide a feasible solution to data exchange among heterogeneous relational databases.

VI . Acknowledgment

This work is supported by the Fundamental Research Funds for the Central Universities.

References

[1] Dey Debabrata, "Entity matching in heterogeneous databases: A logistic regression approach," *Decision Support Systems*, vol. 44, no. 3, pp.740-747, 2008.

[2] C.K.M.Lee, G.T.S.Ho, and H.C.W.Lau, "A generic framework to achieve complex data exchange in an industrial environment," *International Journal of computer Applications in Technology*, vol. 28, no. 2, pp.180-189, 2007.

[3] Gianni Pucciani, Andrea Domenici, Flavia Donno, and Heinz Stockinger , "A performance study on the synchronisation of heterogeneous Grid databases using CONStanza," *Future Generation Computer Systems*, vol. 26, no. 6, pp. 820-834, 2010.

[4] Agustina Buccella, Alejandra Cechich, and Pablo Fillottrani, "Ontology-driven geographic information integration: A survey of current approaches," *Computers & Geosciences*, vol. 35, no. 4, pp. 710-723, 2009.

[5] Fenareti Lampathaki, Spiros Mouzakitis, George Gionis, Yannis Charalabidis, and Dimitris Askounis, "Business to business interoperability: A current review of XML data integration standards," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1045-1055, 2009.

[6] Sridhar M., Mitra M.R.S., Santhi M.N., Kumar, M.V.P., and Sudha P.R, "Normalization in XML and data exchange,". *In: International Conference on Distance Learning and Education,* San Juan, United states, pp. 232-236, 2010.

[7] Lushchenko R.A., Lushchenko O.K., and Tulchinsky V.G, "Improving data integration by means of XML," *In: European Association of Geoscientists and Engineers Conference,* Barcelona, pp. 335-339, 2009.

[8] Anbalagan, Malarvizhi, and Ponmagal, "An XML data representation model for power system reliability assessment," *In: International Conference on Industrial and Information Systems,* Karnataka, pp. 458-462, 2010.

[9] Madria S, Passi K, and Showmick S, "An XML Schema Integration and Query Mechanism System," *Data & Knowledge Engineering*, vol.65, no.2, pp. 266-303, 2008.

[10] Samuel Robert Collins, Shamkant Navathe, and Leo Mark, "XML schema mapping for heterogeneous database access," *Information and software Technology*, vol. 44, no. 4, pp. 251-257, 2002.

[11] Mustafa Atay, Artem Chebotko, PDapeng Liu, Shiyong Lu, and Farshad Fotouhi, "Efficient schema-based XML-to-Relational data mapping," *Information Systems*, vol. 32, no. 3, pp. 458-476, 2007.

[12] Subramaniam Samini, Haw Su Cheng, and Kuan Hoong Poo, "S-XML: An efficient mapping scheme for storing XML data in a relational database," *In: International Conference on Advanced Computer Theory and Engineering,* Chengdu, China, pp. 2149-2153, 2010.

[13] Feng Ye, and Jingsheng Xiao, "Mapping XML DTD to relational schema," *In: International Workshop on Database Technology and Applications,* Wuhan, China, pp. 557-560, 2009.

[14] Dweib Ibrahim, Awadi Ayman, Elrhman Seif Elduola Fath, and Lu Joan, "Schemaless approach of mapping XML document into relational database," *In: International Conference on Computer and Information Technology,* Australia Sydney, pp. 167-172, 2008.