The Design of Mixed Database in the 4PL Infrastructure

Xiaoqiang Liu and Ning Jiang

***Abstract*—In this paper, we state firstly the data classification, design principle of the fourth Part Logistics Service Bus (4PLSB), and then introduce the architecture of 4PL database, a mixed database. Finally, we describe the design of Relational Data Base(RDB) and eXtensible Markup Language Data Base (XMLDB) in 4PLSB respectively.**

Index Terms—**Data, Mapping, RDB, XMLDB, 4PLSB.**

I. Introduction

Combining RDB and XMLDB, mixed database has become a new trend [1],[2]. The first mixed database is IBM DB2 v9 including RDB and XMLDB[3],[4]. Compared with traditional RDB, mixed database can process semi-structured data by means of XML. Furthermore, mixed database can be used various fields because semi-structured data widely exist everywhere. In this paper, authors manage to design a mixed database used in logistics. The 4PLSB is able to provide small–medium enterprises with a collection of logistic services based on Service Oriented Architecture (SOA)[5], but one most important problem in 4PLSB is the representation of non-structured data and semi-structured data. Based on previous research[6], we give an available solution of mixed database in 4PLSB.

II. DATA CLASSIFICATION AND DESIGN PRINCIPLE OF 4PL DATABASE

A. Data Classification

The 4PL infrastructure covers the whole logistic process containing store, transportation, distribution, cost assessment, decision supporting, service validation. Therefore, there is different data type from various data resource. Generally these data can be classified three sorts shown as follow:

Structured data: these data are some entity information such as account, order, product. It is stored in RDB.

Non-structured data: for example, the image of product, the audio introduction of company, these data can’t be represented as two dimensional logic.

Semi-structured: between structured data and non-structured data, these data can be transformed into structured data by indexing on some attributes of things.

From the perspective of architecture, the 4PLSB’s data from three layers, the kernel layer, the support layer of application service and the application service layer. The data described by XML is mainly from the first two layers.

B. The Architecture of 4PL Database

The architecture of the 4PL database is fig. 1 as follow:

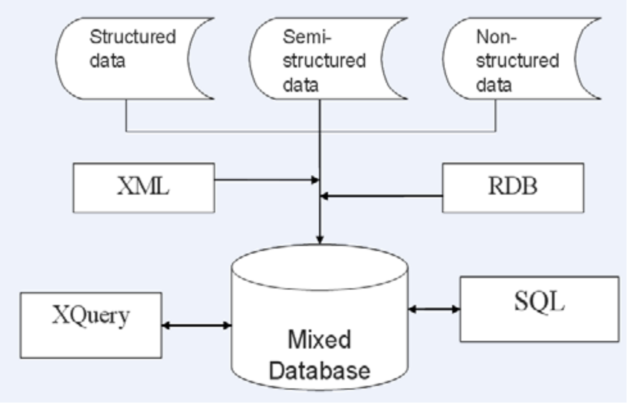

Fig.1 The architecture of the 4PL database

In 4PLSB, structured data are stored in RDB, and semi-structured data stored in XMLDB. Database operations do by SQL and XQuery respectively.

C. The Design Principle

According to the design principle that is “putting data into corresponding database, transforming in time”, we try to design a mixed database combining RDB and XMLDB. “Putting data into corresponding database” means putting structured data into RDB, and putting semi-structured data into XMLDB. “Transforming in time” means structured data can be transformed into semi-structured data when required, or vice versa.

III. The design of RDB in 4PL

A. Design Accordance

Following service standards can be recommended in the design of 4PL database[7]:

GB/T 8567–1988, the specification of computer software product development.

GB/T 8566-2001, the process of software lifecycle

Xiaoqiang Liu is with the School of Software, Nanchang University, Nanchang, China, 330047. (phone: 086-791-8305676; e-mail: lxqiang@ncu.edu.cn)

Ning Jiang is with the School of Software, Nanchang University, Nanchang, China, 330047. (e-mail: kaihaijihua---li@163.com.)

GB/T 11457-1995, software engineering terminology.

GB/T 16680-1996, the management guide of software document.

GB/T 18354-2001, logistic terminology.

GB/T 16472, the classification code of goods, package and package material.

GB/T 17295, measuring unit code of international trade.

B. Name Specification

Table Name Specification

The prefix of table：

a) Basic Description Table BD_[desc]
b) Order Information Table CO_[desc]
c) Do-task Order Table DO_[desc]
d) Stock Information Table IC_[desc]
e) Vehicle Information Table TK_[desc]
f) Transportation Task Table TR_[desc]
g) System Permissions Table TS_[desc]
h) Enterprise Information Table TI_[desc]
i) Client Information Table TC_[desc]
……

The postfix of table: the combination of English word in upper case, the first two characters replaces that word if the word’s length is long.

C. Mapping Principles

Following principles were used in the mapping from object to relation in the RDB design of 4PL:

1. Mapping a class on a table and setting a primary key for each object class, and mapping primary key on the key of the table and mapping other attributes on column.

2. Mapping basic data that need to keep permanently on a class for a high store efficiency.

3. Defining corresponding external key between object classes for describing the relation among objects, such as one to one, one to multiple, multiple to multiple. Generally, other primary key is defined as external key if there are many relevant roles. The external key is set non-null and uniqueness as constraints according to the relevant level.

4. Mapping superclass and subclass on different tables separately if there is generalization relation containing multiple inheritance, and setting the primary key of parent class table as the external key of subclass table.

5. Specifying the field of primary key of “ overall class” in the table corresponding to the “part class” of polymerization relation or combination relation. “Part class” must have the same lifecycle as “overall class”.

6. Enabling table structure with reasonable relation paradigm by adjusting appropriately and controlling redundancy on the tables after mapping.

IV. THE DESIGN OF XMLDB IN 4PL

The semi-structured data in 4PL can be stored in format of XML and then can be queried, exported and serialized into the desired format. We mainly state data exchange and heterogeneous data integration in 4PL.

Taking a simple logistic process as example, we state briefly some design by using XML in the process. The flowchart is shown following:

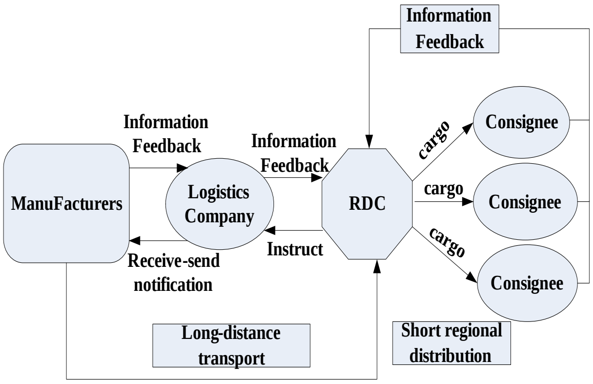

Fig. 2 The flowchart of a simple logistic process

The logistic process can be divided two parts, one is distant transportation, another is short regional. In the first situation, logistic company receives the notification of delivery or pickup from manufacturers, and ships cargos from enterprises to regional distribution center (RDC). There is feedback between logistic company and manufacturers and RDC. In the second situation, RDC ships goods to consignees, and gets the feedback from consignees.

A. Data Exchange

The main task on this aspect is to transform both semi-structured data and non-structured data into structured data, which is the most important contribution of XML from the perspective of business. The issue of data exchange is information standardization including information understandability of both computer and human. It is very important for computer to recognize information, and automatically process data. XML is a powerful tool on information standardization.

Documents is main certificate of information exchange during the each link of logistics. Following cargo receipt is written by XML:

```
< ? xml version = “ 1. 0” encoding = “GB2313”>
<receipt>
   <name>Dealer receipt</name>
   <receiptId>GT0289879-87</receiptId >
   <goodsName>automotive parts</goodsName>
   <quantity>10000</quantity>
   <date>2009-08-28</date>
   <receiver>Mr. Zhang</receiver>
   <place>Honda automotive parts shop</place>
</receipt>
```

Using XML enables rapid data processing between both of deal even they use different management information system.

B. Heterogeneous Data Integration

There are various databases and multiple data formats in the 4PL infrastructure, so called heterogeneous data. One feasible method is to define a common XML format followed by other different document format.

The mapping from RDB to XMLDB means to transform data of RDB into XML format. It is necessary to create data type mapping from various database to XML. It contains three steps, firstly reading the names of fields and their data type from RDB, secondly creating corresponding schema, and finally defining the structure and data type of XML document. Generally, the mapping principle is from table to element, from column to attributes. The content of XML is identifiable by using element tag and attribute tag. These tags are customized.

The following example is to create a corresponding XML schema according to the table named as logistics Company:

```
  logisticsCompany.xsd
  <?xml version="1.0" encoding="gb2312"?>
  <schema xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
     <element name="logisticsCompanys">
       <complexType>
         <sequence>
           <element name="logisticsCompany" minOccurs="1"
maxOccurs="unbounded">
             <complexType>
               <sequence>
                 <element name="id" type="int"/>
                 <element name="name" type="string"/>
                 <element name="contact" type="string"/>
                 <element name="line" type="string"/>
               </sequence>
             </complexType>
           </element>
         </sequence>
       </complexType>
     </element>
  </schema>
```

Specific XML document corresponding to RDB record of a logistics company can be created by using above XML schema, thus the mapping from RDB record to XML document is achieved.

V. CONCLUSION

There are different data types in 4PL, so mixed database is a reasonable choice for 4PLSB. According to the design principle that is “putting data into corresponding database, transforming in time”, we make full use of the advantages of both RDB and XMLDB, and try to address some issues such as data exchange, heterogeneous data integration, and so on. The task we’ll do in 4PL is to transform data automatically from RDB to XMLDB, or vice versa.

REFERENCES

[1] Create a DB2 database for XML. Available: http://www.ibm.com/developerworks/offers/lp/demos/summary/db2xml.html

[2] Shred XML documents using DB2 pure XML Available: http://www.ibm.com/developerworks/data/library/techarticle/dm-0801ledezma/

[3] Query DB2 XML data with XQuery. Available: http://www.ibm.com/developerworks/data/library/techarticle/dm-0604saracco/

[4] Query DB2 XML Data with SQL .Available: http://www.ibm.com/developerworks/data/library/techarticle/dm-0603saracco2/

[5] Xiaoqiang Liu and Tingting Shu, The Fourth Party Logistics Service Bus Based on SOA.*IMECS* 2007

[6] Xiaoqiang Liu and Junhua Wu, the XML Database Application in the 4PL Service Bus. *Journal of Computational Information Systems*. Vol4, No 1, 2008

[7] ICS-GB Applied Guide of Logistics Commonality Information Flat Operation Datasheet Directory