The design and implementation of data exchange based on XML

Yiqiu Fang [a], Xiaolong Gu [b] and Junwei Ge [c]

College of Computer Science and Technology Chongqing University of Posts and Telecommunications 400065 Chongqing, China

[a]fangyq@cqupt.edu.cn, [b]darlon@live.cn, [c]gejw@cqupt.edu.cn

Keywords: Digital Campus, Heterogeneous database, Data exchange, XML.

Abstract. The construction of digital campus is always involves the integration of various data resources, integration and exchange. Discusses digital campus under the "information isolated island" phenomenon, currently used by the XML, ODI technology is widely used to solve data exchange between heterogeneous databases. On the basis of the research on pattern mapping between database and XML based and data type mapping matrix, and puts forward the solving logic problem of heterogeneous exchange rules of use, which solves the digital campus construction in the "information isolated island" problem.

Introduction

With the rapid development of network technology, the construction of campus network in Colleges and universities has entered a new stage of digital campus. The so-called digital campus [2], which is based on the traditional campus, the use of advanced information technology and tools, will be the original digital campus resources. Based on the network, the school resources for information. In short, the digital campus is the mature and advanced information technology as a tool, the activities of the traditional campus re integration, efficient new forms to realize modern institutions of higher education teaching, learning, management and other functions [5].

Summary of heterogeneous data exchange

From the application ways, digital campus construction inevitably involves two core issues, application integration and data integration. The data integration is the most important problems in the digital campus construction. The application of campus network on the surface is very rich, from the school management, administrative office to all aspects of the teaching management, etc. These applications, however, because of the development time and development department, the use of different database [6] differences, led to the "information island"[2]. In order to avoid this kind of phenomenon, realize data sharing and data exchange, it is necessary to set up a perfect platform for the exchange of data integration based on campus network. Through this platform can effectively maintain data consistency and integrity between various departments, in order to improve work efficiency.

The design of the data exchange platform

Platform using the MVC pattern to manage the whole project [4], realize the background and the separation of interface, make the project more structured. Application framework based on MVC pattern, the n-tier architecture, each layer defined communication interface, the function of the upper subsystem using the lower subsystem.

Based on MVC pattern, according to the hierarchy, the system can be divided into five layers. From bottom to top respectively are business entities (Model), data access layer (DAO), business logic layer (BO), control layer (Action) and the Web. Business entities, data access layer, business logic layer corresponding to the model part of the MVC pattern, control layer corresponding to the

© 2015. The authors - Published by Atlantis Press

controller, the Web page corresponding to view part. As shown in figure 1 shows the overall structure of the system [1].

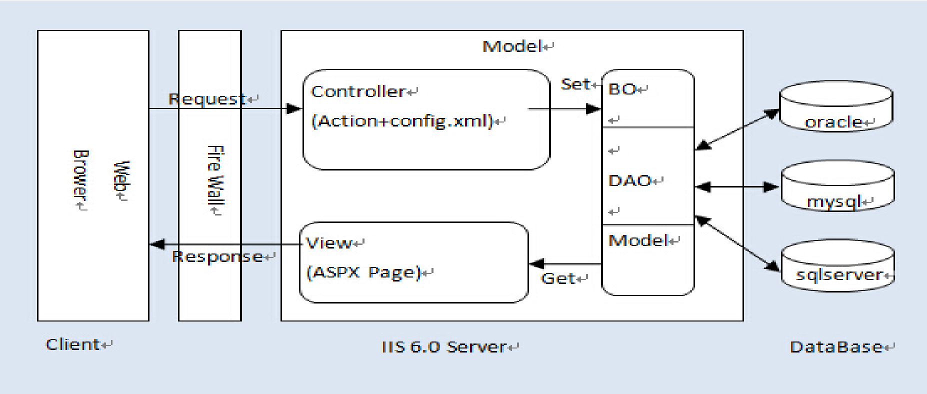

Fig. 1 System overall design architecture

The login module instance design. Login user management module is an important part of the data exchange platform, is also one of the most basic function. Above this module as an example, this paper introduces the design scheme in the concrete implementation steps and technical details. Work processes are: user login interface, allow the user to enter the user name and password, and carry on simple input control. Verify the legitimacy of the user name and password. Verify the login is successful, then draw the basic information of the user in the Session variable, given the login success page; If the login fails, the prompt login failed, asked to enter your user name and password. According to the previous design, figure 2 is the login module chart.

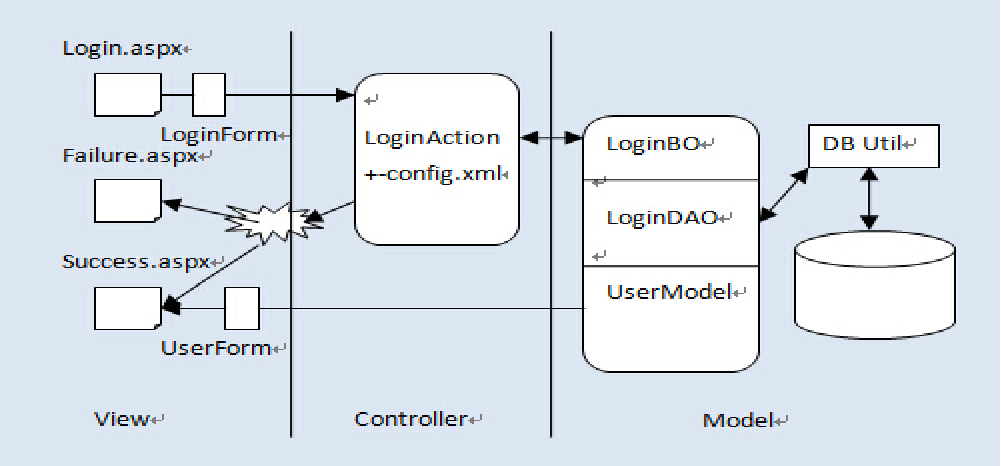

Fig. 2 Login management structure diagram

(1) The view part. Login screen is Login.aspx page, and successful interface is Success.aspx page, and the failure interface is Failure.aspx page, and login interface corresponding Login Form, it is used to pass data page. Page part mainly completed by a web designer, the background control mainly by the application developer is responsible for the division of labor.

(2) The controller part. Login Action with Struts-config. XML to complete the distribution of the controller. Action to accept the login Action, the config in the configuration file. Search an XML mapping, perform Login Action, and call the business logic layer Login BO Validate () method. According to the result of return to the page.

(3) The part model. Business logic layer (BO) mainly includes validating the user legitimacy Validate () method and to achieve the user information Retrieve () method. Validate () method in according to the incoming user name, call the data access layer (DAO) of Login DAO Retrieve () method, obtained from the database user's password, and compares the incoming password, if equal return true, otherwise it returns false.

Login DAO is mainly responsible for the data access layer and database interaction, provide Retrieve () method, etc. User Model save the user's basic information. At the same time, in order to improve the development efficiency and system performance, the DAO database access layer mainly through the provision of components, the use of database connection pool and connection container technology management, enhance the access efficiency.

The platform design of data flow. In data exchange platform, we design the following core modules: authentication center, data management, data maintenance, and data exchange center. The following figure 3 will make a detailed description of platform data flow.

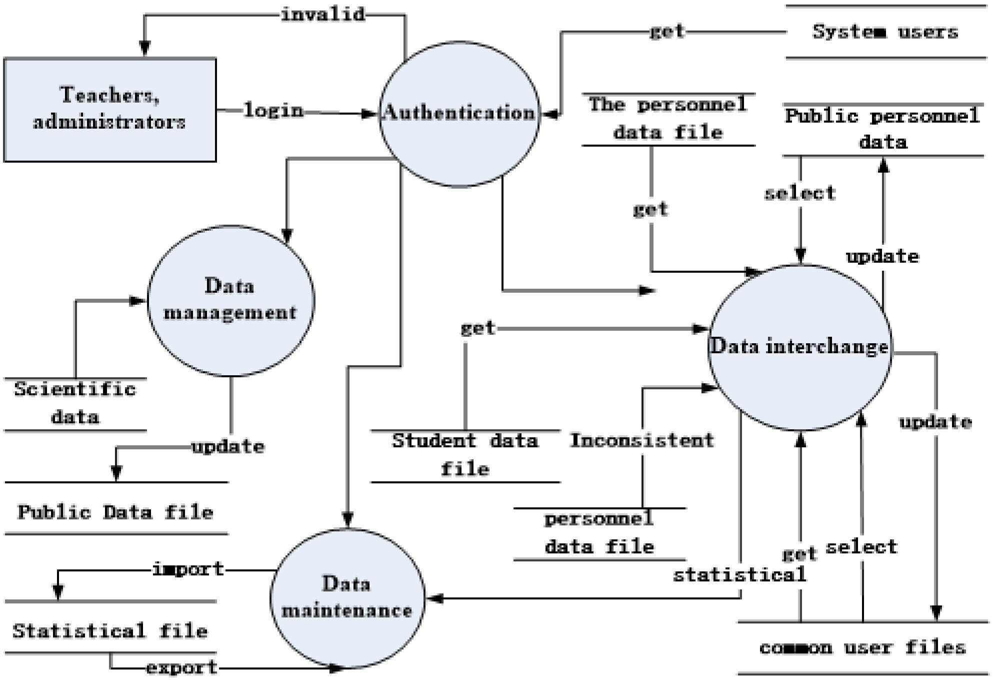

Fig. 3 Platform overall DFD diagram

The realization of data exchange platform

This platform structure of B/S structure, the client program is based on Windows Server 2003 /NT/XP system, with Visual Studio.NET + Adobe Dreamweaver those + Oracle for the development of software applications, the Server side using Windows Server operating system of workstations, is to use Oracle database and used for the development of software service program. Coding adopts the object-oriented program design [3]. Data exchange platform, the main function structure as shown in figure 4.

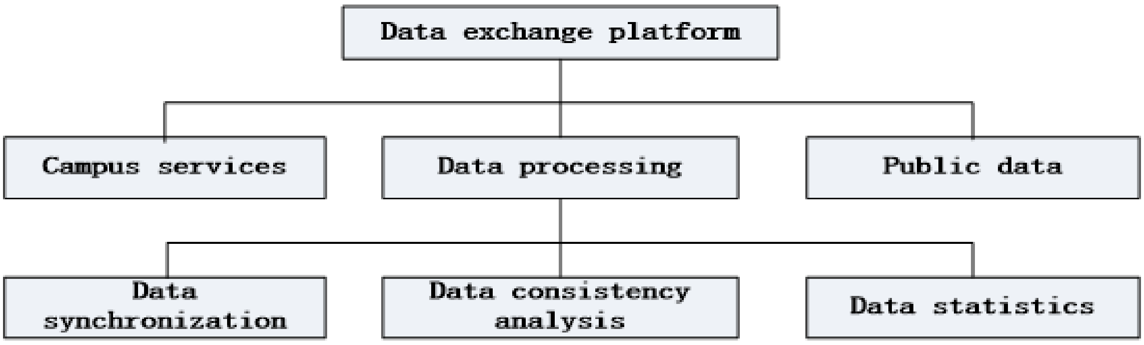

Fig. 4 Main module block diagram

The realization of the database connection. Because the system is closely connected with the database, therefore, before and the database for the household should first establish the connection with database. When connecting to the database, need to load drivers, creating a connection, etc. Here will connect to the database of these operations in a class ADBC, when need to replace, transplant database, you just modify the class, so can reduce the programmer transplant database workload, In Microsoft Visual Studio. NET in 2010, is to realize the connection in the following way: First of all, the introduction of the three basic namespace: using System. Data; using System. Data. SqlClient;

using System. Data. Oracle Client; Then, set up and the function of the database connection, such as the following function, so that at the time of using the database directly and call, such is the Model portion of the MVC.

```
private SqlConnection con;
public static string reStr(String condatabase) { else if (condatabase. Equals("oraconstr")){
return " server=121.83.226.101;  Data Source =gps20; uid=DB; pwd=rs69945";}}
public static DataSet ora_select(string sql){
OracleConnection oracon = new racleConnection(ADBC.reStr("oraconstr"));
DataSet orads = new DataSet();
OracleDataAdapter orada = new OracleDataAdapter(sql, oracon);
try{oracon. Open();orada. Fill(orads);}catch (OracleException e){} finally { oracon. Close(); }
return orads;                              }
```

As you can see, creating a connection is mainly using the Oracle Connection method of System. Data. SqlClient. After creating a connection, can be operated in the need to connect to the database control code, the first to use the Open () method to Open the database connection, and after completion of the call, you can use the Close () method will Open the connection is closed. Such a design, can realize to call the database only when the need to access the database, can release a large portion of memory space, for the security of the database and code running speed are greatly improved.

Effect and achieve the goal

The functional requirements, After running the test data exchange platform, multiple business system with the center of Shared database data timing synchronization, can satisfy the application system basic data complement each other, share and reuse requirements.

Non-functional requirements, Data exchange platform can satisfy the safe, stable and reliable, easy to maintain and better scalability and other non-functional requirements.

To establish a convenient, flexible, accurate and secure data exchange platform, starting from the actual demand of Chongqing Technology and Business University, of all the basic data sharing, the independent scattered data centralized management, reasonable use, is to achieve the overall goal of paper orders.

Acknowledgements

Supported by the Commission of Science and Technology Research Projects of Chongqing, No.KJ1 30533.

References

[1] Kruchten P, Obbink H, Stafford J. The past, present, and future of software architecture. IEEE Software. 2006.

[2] XIA Ke-jian, ZHANG Ying, CHAO Qun, ZHANG Fa-ming, ZHANG Huan-sheng. Application research of XML in data synchronization platform for E-campus [J]. Computer Engineering and Design, 2008, (01): pp483-486

[3] Steffen Zschaler; Birgit Demuth; Lothar Schmitz. Salespoint: A Java framework for teaching object-oriented software development [J].2014, P189-203

[4] ZENG Yan-yang; LIU Shu-fen; GAO Wu-xing. Research of Applying MVC Framework in Command and Control Simulation System [J].Control& Automation, 2010,(31), pp79-80

[5] AVI SILBERSCHATZ, HENRY F.KORTH, S.SUDARSHAN. Database System Concepts Fourth Edition [M].New York: McGraw-Hill Companies, Inc, 2001.

[6] WANG Ning, WANG Neng bin. Query Decomposition and Optimization in Heterogeneous Data Integration System [J]. Journal of software, 2000, (02):pp222-228.