Research on Data Expression in J2EE Architecture system

Zhang Xiaoshuan\textsuperscript{1,2}, Chen Peijun\textsuperscript{1}, Zhao Ming\textsuperscript{1}
\textsuperscript{1}College of Information and Electrical Engineering
China Agricultural University, Beijing, P. R China, 100083
\textsuperscript{2}Jiangsu Provincial Key Laboratory of Computer Information Processing Technology, Suzhou

Abstract: J2EE architecture has many complicated layers, including client, presentation, business logic and data persistence layer. There is respective data presentation in relevant layer of J2EE, with strictly rules to access and transform data objects between these layers. Through the concrete example of engineering application, this paper analyzes the above layers; discuss the data expression of J2EE and the detailed data expressing methods in corresponding layer with some instances. According to the Façade pattern, In J2EE, it should control the access to data persistence layer seriously to protect important enterprise data and avoid showing database pattern to client directly. The data persistence layer hides behind the business logic layer and provide business interfaces via the business logic layer to communicate with the presentation layer to package the data persistence layer, separates the data persistence layer from the presentation layer.

Key-Words: Data Expression; Persistence Layer; Facade

1 Introduction
On the Internet today, with the enterprise level application is developing quickly, and the electronic commerce market is inflating fast, more and more enterprise level projects presents the development trend that they take the Web technology as the central technology. At the same time, the dependence to the server end technology like the middleware is also increasing. So the information and technology department of the Enterprise needs a feasible way to develop the application programs, and make the application programs be related with those middleware which are flexible and can be transplanted.

These application programs should be able to concurrently deal with tens of thousands of users uninterruptedly all day long. One of the problems to build such a complex application program is how to design it, develop it and maintain it. Using J2EE can not only simplify the foundation of the enterprise level application program, but also cause the designers and the programmers to distribute the function in each discreteness of the server end when using J2EE to establish the application programs. As the J2EE adopts the layered thought, and subdivides the process of the corresponding response or the resources that obtains from the user into multi-layer to deal with, and each layer is realized by the component technology, therefore regarding a large-scale enterprise level application, it needs to coordinate the relations of each layer well, and increases the cohesion of each layer, but maintains the incompact coupling. The key of the managing to each layer in J2EE is to control the interior business data and the customer data. And this is the primary coverage of this article--the data expression problem in the J2EE architecture system.

2 The multi-layered architecture of J2EE
The multi-layered architecture of J2EE \cite{1} mainly includes the client layer (the behave layer), the presentation layer (the Web layer), the business logic layer (the application layer) and the data persistence layer (the EIS layer). Each layer has its specific function. Through providing components to the application program, J2EE realizes the multi-layered architecture function \cite{2}.

In the J2EE architecture, each level has its respective data expression method:
(1) The data expression of the Web layer is FormBean \cite{3}, and the data root in the HTML Form POST;
(2) The data expression of the business logic layer is the VO (Value Object);
(3) The data expression of the persistence layer is the PO (Persistence Object), and the data root in the database.

The following is explaining the function of each layer.

2.1 The client layer
J2EE supports many kinds of client types \cite{4}. Because of the B/S structure \cite{5}, the main body of the
program runs at the server end, and the client server does not need to undertake the complex calculate duty (this is the so-called "the thin client server"). The major function of the client layer is to enable the users to carry on the communication smoothly with the server through the user interface.

There are three kinds of technologies on the client layer: HTML (a general electronic commerce network, and is also the most universal kind), the radio equipment (can get information anytime and anywhere), and the Applet or the Java application program (can be used to realize the complex and fast user interface) and so on.

2.2 The presentation layer
The presentation layer is also called the Web layer [6], it runs in the J2EE Web vessel. Its major function is to deal with the HTTP request, and produce the response of HTTP dynamically according to the Servlet and JSP in the Web server.

2.3 The business logic layer
The different technologies the business logic layer [7] use run in the corresponding vessels. They are mainly used to realize the business logic in the enterprise application program. The so-called business logic refers to the processing demand to the data carried by the specific enterprise (client). For example, in the telecommunication enterprise, there is a series of operations such as inquiring the phone bill, filling the account, calculating the cost and so on. Each enterprise has its unique business logic because of there own distinctive quality, so it needs to be realized in the business logic layer.

In the actual project development, the business logic lay is very important; it provides concurrency, flexibility, life cycle management, and fault tolerance and so on for the entire project.

2.4 The persistence layer
The persistence layer [8] is a core problem of data expression, the data expression of other layers is all working for the persistence layer in fact. The persistence layer describes the business data of the application program, and simulates the real entity and persistence layer, as the Figure 1 shows.

In the enterprise development, it always needs to persistent the BO and make the data keep long haul. This is the function of the persistence layer.

3 The research on the data expression in J2EE architecture system
In the J2EE multi-layered architecture, it should strictly control the visit to the persistence layer. One of the reasons that J2EE adopts the layered thought is to protect the important business data, and avoid exposing the database pattern directly to the client server. Generally, the presentation layer interacts with the database by the business layer and the persistence layer, as the Figure 1 shows.

![Fig.2 the interaction between the layers](image)

The word Façade in English means the frontage of the building. But in the design pattern, it means to hide the facility behind the interface (the positive pattern Façade Pattern) by the uniform and simple interface. According to this thought that hides the persistence layer behind the business logic layer, and connects with the presentation layer through the interface that the business layer provided, to realize the separation between the persistence layer and the presentation layer.

In fact, in the actual project development, even if using the Façade Pattern, it is possible to expose the presentation layer to the presentation layer even the client server so long as you do not pay attention slightly. Take some Electrical Power Office as example; it needs to inquire the sale situation of its substation. Obviously, the presentation layer is used to show the sales information, and the business logic layer carries on the corresponding transfer (the Entity Bean or some other ORM frame package object) to the persistence layer according to the request which has been passed by the presentation layer. After obtaining the corresponding business object of the persistence layer, it transmits this BO to the presentation layer, and afterwards the presentation layer transfers the geter method of BO, and shows the data or the information to the users.

It looks nature to do this on the surface. However, when transfer the BO back to the presentation layer, middle, although it has passed through the intermediate layer (the business layer), in fact it makes the presentation layer contact with the
persistence layer directly. Because of the frequent read-write to the database during the persistence layer operation, it must use affair inevitably. But every time the vessel starts an affair, it will read data first from the database and arrive at the presentation layer after the multi-layered transmission. This kind of extra round-trip operation in the database will aggravate the server’s expenses enormously. Therefore generally speaking, it should make the business layer approach to the persistence layer near and be far away from the presentation layer, all the affairs should be put in the business layer to carry on the processing.

4 A case of data expression

4.1 Project background
Some generate electricity management Ltd. has many subordinate power plants. Presently, the local area networks of the group headquarters and the subsidiary company respective have been found, and some of the subsidiary companies have already implemented their management information system (MIS). But because of not carrying on the comprehensive data collection through the software and the network construction, the headquarters is not able to grasp the management information of the subsidiary company by observing the data. As a result, there has formed a information isolated island between the subsidiary company and the headquarters.

The construction of this project can directly obtain the data of the production management, the business management, and the customer relations and so on through the VPN from the MIS and the system-related of its subordinate power plants. This will provide a reliable network foundation for establishing the power plant appraisal system, making development programming, and forecasting the business trade.

4.2 Functional design
The functional module division of the entire application system including evaluation module, data acquisition module, data analysis module, user management module etc. By realizing this application system, it will construct an assistant decision support system which can across network and the system as well as a data analysis platform faced to the management decision.

4.3 The realization
To realize the above functions, it first needs to choose a kind of appropriate architecture [9]. After overall evaluation on each kind of platform and the architecture system based on this platform, it chose J2EE finally.

The application architecture divides into three tiers in the figure: the data tier, the application tier, and the channels tier. And the application tier is subdivided to three parts: the application server, the framework, and the application.

With the application of EAI, the website may realize the application integration with other carry-over systems. With the ETL tool, it could realize the business intelligence (BI) application of the application data.

Regarding to the design of the data persistence layer, although there are many kinds of data memorizer, in the business data aspect, the DBMS on the database server basically uses the relational database in the current major enterprise developments. How to map the object and the relational database, and enhance the system capability, is the content needs to be considerate when design the data persistence layer.

The data expression in the business logic layer is VO, and the presentation layer is Formbean. The former in fact is a JavaBean, and the latter can be realized by inheriting the Formbean which is a class in the Struts. The Formbean class is used to receive the user data, and validate the grammar and others of the data by using the Validate () method. Moreover, it also extracts data from the business layer and gives them back to the users.

5 Conclusion
This article has proposed a blue print to solve the problem of the data expression. Based on the development architecture composed by the Struts and the Hibernate, it passes the data by the value object which the layers corresponding to, and strictly controls the visit to the persistence layer by the users. In this way, it can protect the business data effectively. Besides, it is useful to the upper debug and maintenance by separating the business logic and the data expression, and also by separating the data in different layers.

With the development of the Internet, it is absolutely necessarily for a large scale enterprise to be informational. Thus the application foreground of J2EE is very amplitude. And the data expression question in its architecture system is becoming a hot topic. More and more open source or non-open
source units or organizes are wild about the research on it. Each layer has its architecture package, such as the Struts and Spring Architecture of the business logic layer, the JDO, Hibernate ORM package of the persistence layer and so on. The sealed pattern of the data expression in each layer influents the efficiency of the project development directly. The most popular way is to use both the Struts and Hibernate architecture. The Struts emphasize particularly on the control of the business logic, and the Hibernate is mainly used to seal the business data. Many projects have indicated that this way is effective. In the data expression domain, there are still lots of details waiting people to explore and dig. It believes that with the lapse of time, there will be more outstanding methods to solve the data expression problem smoothly, and optimize the system architecture.

6 Acknowledgements
This research is supported by the open fund from Jiangsu Provincial Key Laboratory of Computer Information Processing Technology, Suzhou University. The authors are grateful for the anonymous reviewers who made constructive comments.

References: