

## Five-layer Web Development Model Based on J2EE Architecture

Wang Zhenghong<sup>1, a</sup>, Li Xiaoping<sup>2, b</sup>

<sup>1</sup>Beijing Institute of Technology Continuing Education and Modern Distance Education College ,  
Beijing 100081

<sup>2</sup>Beijing Institute of Technology Adult Education College , Beijing 100081

<sup>a</sup> wzhwzh@bit.edu.cn, <sup>b</sup> lxpmx@x263.net

**Keywords:** J2EE; MVC; Struts; Hibernate; Multi-source data

**Abstract.** Currently ,there commonly exists some inadequacies such as Low degree of reusability、tedious Maintenance work、weak Program contingency etc. in Multi-tier web architecture. Therefore, it is particularly important to improve the efficiency of the development , selection of the optimal technology for stratified design and framework building, and giving full play to the application of system's flexibility , safety and practicality in the process of developing a Web application system. The authors explore the web development framework of J2EE-based five - layer architecture, analysis the limitations of object persistence which is based on Struts framework in J2EE architecture, and designed a multi-source data access components between the business logic layer and persistence layer, which realized a uniform access to heterogeneous databases of different types , different structures , different environment , different usage.

### Web development framework model of the multi-tier architecture

#### 1.1 Five-layer architecture design of J2EE

As Web applications require more complex performance and logic processing , J2EE has proposed a four-layer architecture , divided into the client layer , web layer, business logic layer and database layer. And with the separation between logic processing and performance of application , the system has a clearer picture of the logical flow , function code reusability and distributed deployment . However , in most of the actual project application , the data layer is a relational database , not explicitly mapped to achieve an object - oriented mechanism, and the interaction process between business logic layer and database needs a further package for database calling interface, which increase a certain degree of complexity and the difficulty of management to the development and maintenance of the project. An effective solution is to split open business information in accordance with the functional modules :business logic separates from database access and user interface separates from the business logic, relatively independent from each other, that is, any change of either will not affect the other , or little effect. Therefore, an data persistence layer between the business logic layer and database layer is increased in the paper, to extend the four-layer system for the five-layer system shown in Figure 1 , the client layer , Web layer, business logic , data persistence layer and database layer . Data persistence layer located on top of the database result in hiding all details of the data access code within data read and operation, isolating the client application from the underlying storage mechanism , and completely abstracting out the the data physical details of the development application .

(Figure1 J2EE-based five-story architecture design)

Increase a data persistence layer between the business logic layer and data layer , realizing the mapping between objects and relational databases. With the use of this mapping framework mechanism, the conversion between the object and relational database can transparently operate without caring about the issues ,such as database connection, concurrency, transaction etc. Business Logic Layer direct access or storage is clear objects, and the conversion during the process is dealt with the mapping framework.

In the hierarchical design, the interfaces of all layers is the basis of communication between them, and the principles followed is to achieve strict interlayer independent 、 separation , and the

implementation details of the layers not disclosed. Communication between the hierarchical design layer is described as follows :

Client presentation logic layer via HTTP protocol communication, that is, through of Http Request and HttpServletResponse receives user's input and returns results to them.

Presentation logic layer uses the Struts framework technology to realize many functions, such as providing the receiving / responding to client requests , to control the entire system workflow and dynamically generating a WEB page by calling the interaction between Action and business logic layer and formatting business data.

The business logic layer and persistence layer's interaction is JavaBean calling of Hibernate , that is, through Data Access Objects DAO to call .

The communication between persistence layer and database is entirely controlled by Hibernate who can make entity mapping to the database , operate the persistent objects and synchronize the results from cache into database.

## **1.2 Integrated Struts framework and Hibernate framework to achieve the MVC design pattern**

As based on the model (Model) - View (View) - Controller ( MVC ) mode of application architecture, Struts has the advantages of modularity, flexibility and reuse of components. However, struts framework is mainly designed for the presentation layer , not a very strong support for the back-end business logic layer. During project development , there are the following limitations : the division of labor is horizontal , work is divided by module ,and software development costs are correspondingly higher; A lot of time should be spent on the packaging of data layer as well as the coordination and communication between the different modules, leading to the time increasing on development; Project transplantation is relatively poor , leading to write different SQL statements for different databases ; The project has relatively poor scalability so that we have to modify the structure of the database table to adapt to new requirements or changes, however, which calls for higher personnel requirements to re-write the SQL statement and backup databases and do other complex operations. As the developers of database operations is uneven and the experience of development is not the same , resulting in system performance may be relatively poor. Hibernate is an open source O / R Mapping ( Object / Relational Mapping ) framework which conducts a lightweight object encapsulates for JDBC to handle database operations by object-oriented mechanisms. Hibernate not only manages the mapping from Java classes to database tables, but also provides data query and retrieval of data, which can significantly reduce the time for manually processing data through using SQL and JDBC within development. The authors points out that Hibernate should be integrated into the J2EE architecture and the object persistence part be separated from the struts framework to the Hibernate to achieve. According to Struts architecture, database operations work is handled by the model layer, so the model layer of Struts can be divided into two parts, one part responsible for business logic and the other using Hibernate to realize object persistence processing. Meanwhile isolate specific business logic to create a new business logic layer ,which is responsible for the interaction between business logic and persistence object with Hibernate .

Figure 2 shows the integration of Struts and Hibernate MVC model . Integration of Struts and Hibernate framework achieves the separation of the control flow , business call and expression so that the system has a good improvement in terms of development efficiency , maintainability , scalability. In this model , the client layer can be viewed as V, it is the DHTML interface displayed to the user by the browser ; Business logic layer and persistence layer are compared to M, which is the core part of the system , by JavaBeans and Hibernate. The Web layer corresponds to the C , which controls the users' operation of the business logic , while controlling operating results are displayed to the users , and is composed of the Servlet and JSP program . Among them PO is the persistent object produced through calling model layer and is displayed to the users by JSP .

(Figure 2 Integrated Struts and Hibernate MVC model diagram)

### 1.3 Integrated Struts and Hibernate to realize J2EE layered architecture

According to the previous analysis , the Struts framework and Hibernate framework for integration , constitutes a new Web application development framework to achieve a multi-tier architecture of J2EE applications. On the one hand the framework inherited the Struts framework in the presentation layer has the advantage of providing a comprehensive tag library and is responsible for receiving and forwarding page request to realize the separation of presentation logic and business logic ;on the other hand it plays the characteristics of Hibernate framework in the data persistence layer and packages persistence layer and transaction by Hibernate framework to realize the separation of business logic and database access .As a result,it forms a clear hierarchy, more concise , more fully featured Web framework ,which can reduce the coupling between the various levels , improve component reusability , reduce the complexity of the code editor, help developers focus in the implementation of the business logic and be conducive to the maintainability of the system .

Integrated Struts and Hibernate to realize J2EE layered architecture as shown in Figure 3 , the system uses the five-story structure design composed of the client tier , Web tier , business logic , data persistence layer and database layer . The client layer is running on the Web browser of user' computer; Web layer is running on the web server which uses the Struts framework technology to realize mang functions, such as providing the receiving / responding to client requests , to control the entire system workflow and dynamically generating a WEB page by calling the interactive between Action and business logic layer and formatting business data; Business logic layer is responsible for implementing the core business logic of the whole system , realized by JavaBeans or EJB; Data persistence layer realized by Hibernate framework technology completes the mapping of objects and relationships and is responsible for database operations.

Figure 3 Integrated Struts and Hibernate to realize J2EE layered architecture

The following simple analysis based on the functions of the layers of the system of Figure 3 .

Client layer: It is the one by which user uses a browser to see and interacts with the system is formed by the HTML language web interface.

Web layer: Customer input obtained from the client layer , passed to the business logic layer components , and then to transport the results, which is obtained by the business logic layer , in the form of HT2ML files to the client forms a web interface. Web layer composed of Servlet and JSP program is packaged in the Web container. The control of business process is generally developed by controller Servlet, which can response to a user's query request and call JavaBean of business logic layer to achieve complex business logic .

Business logic layer: It can deal with the user response from presentation logic layer and return the results to the client layer . Business logic layer encapsulating the interface provided by the system is the center to deal directly with the user's request . This layer is mainly realized by the JavaBean. JavaBean 's main task is to deal with the business logic , to interact with clients and to return the appropriate operating results to it and so on.

Data persistence layer: It completes the mapping from persistent objects to relational database and makes an operation on persistent objects. The business logic module through Hibernate using O / R mapping files to achieve the operation of specific data source (that is, through the persistence layer is mapped to a specific data table ) ,and complete the operation of the database. For small items , this implementation is very efficient and low cost. Because of this way , you do not need to add the application server containing the EJB container , only need a Web server can allow the system to normal operation .

Database layer: The concrete realization of object persistence , can be a relational database management system , file storage and object database storage management system .

### Design of multi-source data integration model

Currently, Enterprise-class application system widespreadly lacks uniformity in the whole and the distributed information resources leads to data sharing being a very difficult problem. These data management system for independently developed for different services , although be able to

meet the business data storage and management requirements, but in many cases , in order to make a decision , it may need to access the data in a plurality of business management system distributing in the different positions of the network. Therefore , it is very important to integrate the scattered information resources, through relatively independent of the original multiple application system integrated, it can eliminate the original / information silos 0 phenomenon to realize data sharing and eliminating data redundancy and enterprise information resources integrated, which can provide integrated , unified , secure , quick information queries, data mining and decision support services to integrated enterprise application systems.

### **2.1 Multi-source data integration strategy**

Hibernate using the database configuration file provides persistence services ( or persistent objects ) to the application . Hibernate converts between Java programs and databases ,when Hibernate accesss the database , the data type used by the underlying database is transparent to the Java application. JDBC driver packages the SQL type used by underlying database , up to provide the standard SQL type interface , so Hibernate can use standard SQL type to generate the DML ( Data Manipulation Language) . When system released, according to different data sources, it can define different Hibernate configuration files to make it possible to access to different databases and realize multi - source data integration.

### **2.2 Multi-source data object mapping**

(1) Definition. According to the business areas and the scope of the need to integrate the application system, to determine the integration of data sources and data objects .

Definition 1: The data source A is a relatively complete collection of data , he is composed by the number of data objects B , that is ,  $A = \{ B_1 , B_2 , \dots B_m \}$  . A data source in practice can be a stand-alone database .

Definition 2: B is set to a data object in the data sources A , B has a number of data elements , that is ,  $B = \{ b_1 , b_2 , \dots b_n \}$ , the data object in the data source can be a database tables and views .

(2) Data mapping rules. From the need in integrating the data source to Hibernate's persistence data collection , it can establish the following mapping rules:

Rule 1: For a data source A, create a corresponding session configuration file and specifies the corresponding from the data source to the databases in the session configuration file, such as : hibernate Acfg . Xml .

Rule 2: For each data object B of data source A, it should create a corresponding persistent class C and a corresponding mapping file D be created for persistent class C and data object B, and each data element of B correspond to a class attribute of C .

## **References**

- [1] Fang Wei , Sun Yong , ZHANG Shu-kui . Integrate Struts and Hibernate Web system application [ J ] Computer and Modernization, 2005, ( 12 ) : 39 - 41
- [2] Song Xiuqin , Houdian Kun , Fang Zhongchun . The building of Struts and Hibernate - based Web application[J]. The Network and Communication , 2005, 21 ( 3 ) : 125 - 127.
- [3] Zhu Qinghong , Wu Yuhong. The analysis and application of an object / relational mapping framework[ J]. Electronic Science and Technology , 2004 , ( 1 ) : 54 - 60 .
- [4] Gao Ang , Wei Wenxue. Based the Hibernate Struts framework data persistence Application Research [ J ]. Journal of Computer Applications , 2005 , 25 ( 12 ) : 31 – 35.
- [5] Xie Yanping Hu Jiabao, Xie Chengwang. Based on Struts and Hibernate MVC design Mode [J]. Communication and Computer, 2005 , 23 ( 4 ) : 62 – 63.