

## Design and Implementation of High-quality Course Scoring System Based on Struts and Spring and Hibernate Architecture

Dawei LIU

Department of Foundation  
Engineering College of Chinese Armed Police Force  
Xi'an, China  
wjmicheal@163.com

**Abstract**—In order to solve the requirement of a high-quality course scoring system which is B/S model, this paper proposes a solution schema based on Struts and Spring and Hibernate integrated architecture. After analyzing the integration of these three frameworks, this paper presents the procedure of design and implementation of the system. Result shows that in the construction of business system, Struts and Spring and Hibernate architecture can improve the efficiency of system development and strengthen the extensibility and maintainability of the system.

**Keywords**—Struts; Spring; Hibernate; MVC; loose coupling

### I. INTRODUCTION

Over the years, in the lecture competition of Engineering College of Chinese Armed Police Force, traditional manual scoring method was always used, but there were many disadvantages of this way such as scoring process was not specific, efficiency of the judges was very low, deliberate deception and so on. So in this case, the support of a real-time, efficient scoring management platform based on network is needed urgently.

However, as the complexity of the system requirements and the urgency of the development cycle, it is not easy to develop a powerful and expansible web system using the traditional development methods in a short time[1]. Systems based on SSH(Struts+Spring+Hibernate)architecture has the features as low coupling, high reusability and expansibility, high maintainability and so on[2]. So SSH architecture becomes a better solution of this system. Taking high-quality course scoring system of Engineering College of Chinese Armed Police Force as the development instance, this paper analyzes in detail the application of SSH architecture in the web environment.

### II. OVERALL DESIGN OF SSH ARCHITECTURE

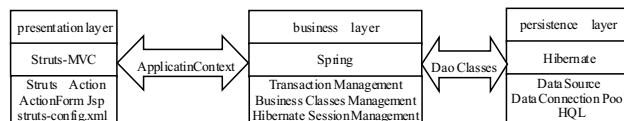


Figure 1. Overview of SSH architecture

SSH architecture is divided into three layers: presentation layer, business layer and persistence layer, as is shown in Figure 1. Each layer's function is very clear, the layers communicate with each other through particular interfaces.

The Struts framework is used in the presentation layer. The centerpiece of Struts is an MVC(Model-View-Controller)-style controller[3], it is responsible for the MVC, it can reduce the coupling of the architecture. The Hibernate framework is used to support the persistence layer which can reduce the difficulty of the business model development[4]. The Spring framework is used to deal with the complex business of the business layer. Web systems developed with this architecture will have lower coupling not only in the whole architecture but also in the partial complex business models, their flexibility and maintainability can also be enhanced[5,6]. This architecture can overcome the insufficiency of developing systems using these frameworks alone.

The three frameworks are needed to be integrated when we use this architecture to develop systems.

#### A. Integration of Struts and Spring

Struts actions will be delegated to Spring when we integrate these two frameworks. Specifically, in Struts configuration file struts-config.xml, we use the class *DelegatingActionProxy* supplied by Spring as the *type* property[7], the class *DelegatingActionProxy* extends from the class *org.apache.struts.action.Action*, it sends the request to the real action. In this way, the class loaded by Struts in the run-time is actually *DelegatingActionProxy* which achieves the proxy of the real action, so that the action instances called by Struts will be managed by Spring finally.

#### B. Integration of Hibernate and Spring

The most important configuration file of Hibernate is hibernate.cfg.xml. The classes needed to be persistent and the related global properties are defined in this file[8]. When these two frameworks are integrated, Hibernate will be managed by Spring. Contents of the former file hibernate.cfg.xml are also managed by Spring. Spring implements the encapsulation and interface conversion of Hibernate by using the class *LocalSessionFactoryBean*[9]. So that we can use a unified way to deal with Spring and Hibernate. In Hibernate application, the Spring class *HibernateTemplate* can complete the work as catching exceptions, managing transactions and so on without any redundant codes.

#### C. Integration of these three frameworks

After finished the two steps of A and B, then copy the jar packages to the classpath, we will complete the integration of

these three frameworks.

### III. REQUIREMENT ANALYSIS OF SYSTEM

#### A. Description of the system functions

High-quality course scoring system is used by the judges to score and vote for the teachers in the lecture competition of the college. The system has two kinds of roles, judge role and administrator role. Administrator mainly manages the information of the teachers that take part in the competition. The system sets up a certain number of judge users. The system distinguishes the judge by the IP address of the computer that he uses. In the lecture competition, every judge types a score to the teacher who is lecturing and then submits the score to the server. At last the system takes the average of all the scores that remove the highest and the lowest as the final score of the teacher. The judges can view the teachers' scores who have already finished their lectures while the competition is going on. When all the lectures are finished, judges can vote on the teachers whose scores are higher than others to select the high-quality course teacher. High-quality course scoring system should have the main functions as follows:

- Teacher management. Operated by administrator role. The user can create and modify the information of the teacher including affiliation, title, and content of the lecture and so on. The user can specify the current lecturing teacher and modify the lecture orders of the teachers.
- Judge scoring. Operated by judge role. The user can give a score to the teacher who is lecturing according to different evaluation indexes. When the score is submitted, the user can not score again. The user can view the teachers' scores who have already finished their lectures. Validation of the score can be monitored in this function module.
- Score querying. Operated by administrator role. The user can query the result of the competition in descending order of the scores. The user can also query the result of the competition group by teachers' titles or affiliations.
- Judge voting. Operated by judge role. The user can vote on the teachers to select the high-quality course teacher when the competition ends. Every judge's voting numbers are limited.

### IV. SYSTEM DESIGN

#### A. Design of the presentation layer

The presentation layer uses the MVC design pattern. This layer is divided into three parts, model, viewer and controller which keep relatively independent to implement the decoupling of pages input, handling and output. The viewer part is implemented by jsp and html pages. The model part is implemented by the service components supplied by the business layer. The controller's work is completed in collaboration by Struts class *ActionServlet* and *Action*. Struts class *ActionForm* is responsible to encapsulate the page data

transferred from the client to implement the mutual support of viewer and model.

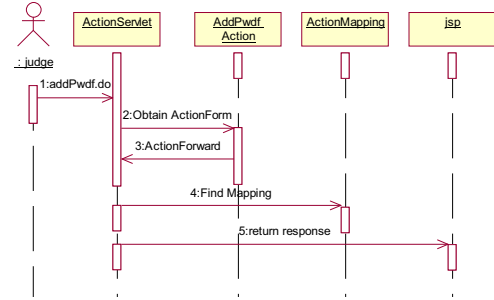


Figure 2. Sequence diagram of the judge scoring module's persistence layer

Taking judge scoring module as example, we illustrate the process of presentation layer. The sequence diagram is shown in Figure 2. The user enters the page and types the score of the teacher, then he submits the page to the action whose *path* property is *addPwdf.do*; The class *ActionServlet* intercepts the user's request and finds the corresponding action class of the *do* file through reading the *path* property in the file *struts-config.xml*. The *ActionServlet* finds the corresponding *ActionForm* through reading the *name* property and then instantiates this *ActionForm*, it fills the *ActionForm* with the properties of the page at the same time, at last the *ActionServlet* instantiates the *Action*; then the *Action* runs the method *execute()*, and gives the execution result back to the *ActionServlet*; at last the *ActionServlet* gives the response page to the user through finding the *ActionMapping*.

#### B. Design of the business layer

The business layer mainly provides business service to the presentation layer. The business layer defines the unified interface *XxxService* of each module to the presentation layer. The class *XxxServiceImpl* implements the methods of the corresponding interfaces. This layer calls the relevant classes according to the need and implements the visit of the database. Taking judge scoring module as example, in this module the system defines the interface *PwdfService* and the implementation class of the interface *PwdfServiceImpl* which encapsulates the class *PwdfDaoImpl* and implements the interaction to the persistence layer.

The business layer uses the IoC( inversion of control) design pattern, on one hand it can implement the controller's calls to the business components, on the other hand it can implement business service components' calls to their collaborators—persistence components DAO(data access objects).

#### C. Design of the persistence layer

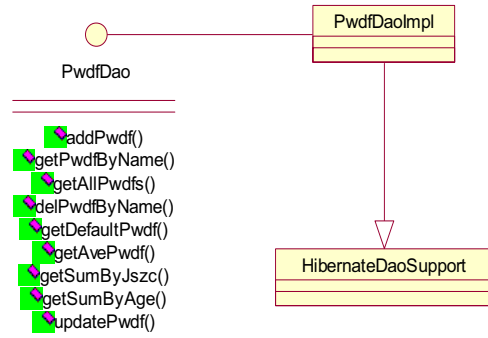


Figure 3. Class diagram of the judge scoring module's persistence layer

To a business system, the aim of designing and developing a system is to supply support for some specific business, the business is often the core of the system, so the separation of the business and the data access is particularly important[10]. In the persistence layer, the system uses the DAO design pattern to implement the separation of business and data access and defines this layer's interfaces and realization classes which extend from the Hibernate class *HibernateDaoSupport*, so that the CRUD (create, update, read, delete) operations can be implemented by Hibernate. The class diagram of the judge scoring module's persistence layer is shown in Figure 3.

## V. IMPLEMENTATION OF THE SYSTEM

After the design, we complete the implementation of the system. To the presentation layer, the system uses jsp and the Struts tags to create interactive pages with the users. Users' requests will be received by the Struts controller *ActionServlet* that decides which action will process the requests through finding the configuration file *struts-config.xml*.

Business layer is the core of the system, it is implemented through the using of Spring framework. In order to create this layer, we mainly complete two parts of work: configuring Spring context and writing business codes. In this system, the Spring configuration file is split into three files which configure respectively the information of all the actions in Struts and their properties, information of the beans needed by the business layer, *SessionFactory* and the transaction management of Spring. When we write the business codes, the business layer obtains the references of the DAO interfaces through configuring them as its properties, so that it can complete the persistence task, the DAO properties are loaded dynamically through the injection of Spring to guarantees the loose coupling of the business layer and the persistence layer.

Part codes of the class *PwdfServiceImpl.java* are as follows:

```

public class PwdfServiceImpl implements
    PwdfService {
    private PwdfDao pwdfDao;
    public boolean addPwdf(Pwdf pwdf,String
ip,String name){
        return pwdfDao.addPwdf(pwdf, ip, name);
    }
    public List getPwdfByName(String name){

```

```

        return this.pwdfDao.getPwdfByName(name);
    }
    ...//others are omitted.
}

```

Implementation of the persistence layer is completed through the using of Hibernate framework. In order to implement this layer, we need to create the relation mapping files of database tables and Java objects, and write the interfaces of persistent classes and its implementations. Since the system integrates Spring and Hibernate, Spring provides a *HibernateTemplate* which well encapsulates the operations of the Hibernate Session, with *HibernateTemplate*, we only need to write HQL (Hibernate query language) statements and call the appropriate template methods, which make us out of the operations for each time to obtain a Session instance, start transactions, commit or rollback transactions, and the complex try, catch, finally operations. So that we can gain sharp code presentation.

## VI. CONCLUSION

This paper presents a Web business system based on SSH integrated architecture, and provides a multi-layer Web application development solution. High-quality course scoring system based on SSH architecture was respectively used in the lecture competition of our college in December 2010 and May 2011. The actual running of the system is highly stable and reliable. Practice shows that, Web applications based on SSH architecture have the features as short development cycle, stability, high reusability and easy maintenance. SSH architecture has been proved to be an effective lightweight J2EE application solution.

## REFERENCES

- [1] Feng Run-min. Design and Implementation of High Education Student Management System Based on Struts&Hibernate&Spring [J]. Computer Engineering, 2009, (06):280-282.
- [2] GUO Guang-jun, XIE Dong, CHEN Li-pin. Design and implementation of teaching support system based on Struts-Spring-Hibernate integration architecture [J]. Computer Engineering and Design, 2009(30):3111-3116.
- [3] [Ted Husted. Struts in action [M]. Greenwich: Manning Publications, 2003.
- [4] LI Jie. Research and Application of Lightweight Data Persistence Technology Based on ORM [J]. Computer Science, 2010,(09):190-193.
- [5] WANG Jian-wen, HAN Li-peng, ZHANG Jun-ming. Design and implementation of material management system of oilfield based on multi-layer architecture [J]. Computer Engineering and Design, 2011(01):162-165.
- [6] [6]LIU Fang, GUO Hong-xia, QI Quan. The Applied Research of E-HR System Based on SSH [J]. JOURNAL OF WUHAN UNIVERSITY OF TECHNOLOGY, 2009(3):98-100.
- [7] JIANG Wei, MA Guangsi. Integration and Process Analysis of Spring and Other Frameworks [J]. Computer Engineering, 2007, (07):79-81.
- [8] WANG Hai-tao, JIA Zong-pu. Web Application Development Based on Struts and Hibernate [J]. Computer Engineering, 2011(09):112-114.
- [9] Craig Walls, Ryan Breidenbach. Spring in action [M]. Greenwich: Manning Publications, 2007.
- [10] Xia Xin. Head First Hibernate [M]. Beijing: Electronic Industry Press, 2005.