

Design and Realization of Ancient Fabric Information System Based on SSH

Feng Zheng

College of Electrical Engineering, Zhejiang University
Hangzhou, 310027, P. R. China
E-mail: zzfzwan@163.com

Senlin Zhang

College of Electrical Engineering, Zhejiang University
Hangzhou, 310027, P. R. China
E-mail: slzhang@zju.edu.cn

Meiqin Liu

College of Electrical Engineering, Zhejiang University
Hangzhou, 310027, P. R. China
E-mail: liumeiqin@zju.edu.cn

Zhen Fan

College of Electrical Engineering, Zhejiang University
Hangzhou, 310027, P. R. China
E-mail: fanzhen@zju.edu.cn

Abstract—Ancient fabrics are of high importance to the modern fabric industry, and have attracted large amounts of amateurs. Therefore, they should be protected properly while accessible. In order to solve the problem, this paper designed and implemented a system to provide services for ancient fabric managers, product designers, system administrators and ancient fabric amateurs. The system is designed and implemented on B/S architecture, with its database built in MySQL. By using SSH framework (Struts2, Spring and Hibernate), the system is going to satisfy the requirements of the four groups of people. Compared to the existing information management systems, the system in this paper focuses specifically on culture relic information management, and is able to manage different kinds of relic information comprehensively. It will be easier for professionals and ordinary people to get in touch with ancient fabrics, and therefore promote the beauty of the culture as well.

Keywords-SSH ;IoC; Spring; Struts2; Hibernate

I. INTRODUCTION

As one of the birth places of world's ancient civilizations, China has brought the world countless heritages with its long history and splendid culture. Apparently, ancient silk fabrics are one of the most attractive heritages. Ancient silk fabrics are delicate as well as vulnerable, and should be protected properly in special storerooms against damp and worm damage because of its chemical and physical properties. Consequently, their true faces can rarely be seen by the public. However, the detail information of ancient silk fabrics is of high importance both to the modern fabric industry and ordinary people, for it does not only reflect the glorious culture, but also provides abundant knowledge to modern silk designing and producing.

Therefore, an ancient fabric information system is strongly needed in order to make full use of its detail information. With the rapid development of J2EE frameworks nowadays, it would not take long to construct a web application. However, there is a lightweight framework named SSH framework which is a better solution of the ancient fabric information system in this

paper, for the reason that it has low coupling, high expansibility and maintainability [1].

This paper will firstly introduce the SSH (Struts2, Spring, Hibernate) architecture in three parts respectively, and then analyze the integration of each part. After that, the author is going to design and realize the system based on this architecture [1].

II. THE SSH FRAMEWORK

A. Brief Introduction to SSH

SSH is an integrated framework of Struts2, Spring and Hibernate. The system based on SSH is usually divided into four parts: the presentation layer, the business logic layer, the data persistence layer and the domain layer. With all the four parts together, the system can help web developers build a reusable and expansible web application in a short term. Among the three components of SSH framework, Struts2 is foremost in importance. It is responsible for the separation of model layer, view layer and controller layer, and controls the data flow between the three layers [2]. Hibernate framework, which is an Object/Relational Mapping (ORM) solution, provides support to the persistence layer. Spring is a lightweight container framework with inversion of control and aspect oriented programming. Many logic businesses can be accomplished by Spring with basic JavaBean [3].

B. Introduction to Struts2

Struts2 framework can be divided into three parts: view, controller and model. Among the three parts, the controller integrates the view layer with the model layer. Model layer is the carrier of data which works between front end system and database. In the model layer, Struts2 can easily work with many kinds of data access technology, including EJB, JDBC and Object Relation Mapper. In the view layer, Struts2 can work with a lot of presentation layer technology such as JSP, Velocity and so on. View layer is the place where users can actually interact with the system. Controller is the bridge between model layer and view layer, and it is responsible for reading users' data and command from view, and then transmits them to the model.

MVC framework separates the system into three parts so as to enhance the system scalability and maintainability [4]. The workflow of Struts2 is shown in Fig .1.

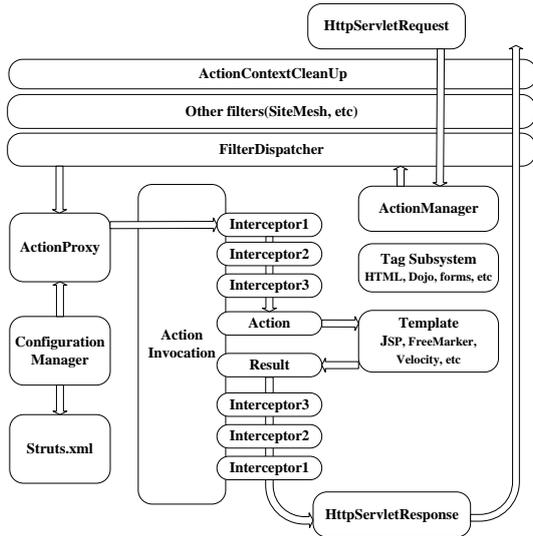


Figure 1. Workflow of Struts2 framework

C. Introduction to Spring

Spring is a popular framework aiming to developing complicated enterprise application by using layered structure. The core of Spring framework is an lightweight container, which realizes Inversion of Control (IOC) and provides Aspect Oriented Programming (AOP) technology. Besides, Spring also provides an implementation of MVC web framework. Bean Factory, which is a kind of container of IOC, is the most basic and important part of Spring framework. Bean Factory is like an assembly line, under whose management people can create objects or use dependency injection to manage the relationship between components. The relationship between each component therefore is code independent, so Spring realizes the decoupling of the system. Another feature supported by Spring is the aspect oriented programming (AOP), which can separate each part of business logic, and reduce the coupling among the various parts to improve the reusability of the program, and at the same time improve the efficiency of the development. By using Spring framework, developers only need to focus on the business logic itself, while Spring takes the rest work for them [5]. The components of Spring are shown in Fig .2.

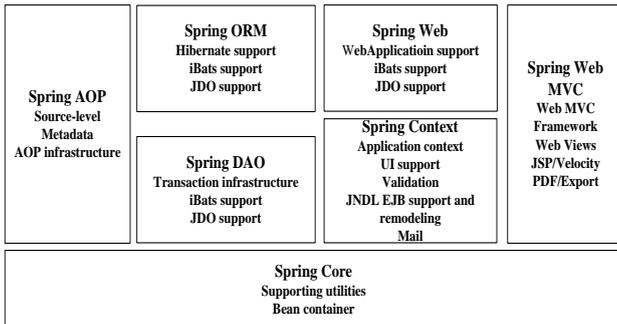


Figure 2. Components of Spring framework

D. Introduction to Hibernate

Hibernate is an Object/Relational Mapper framework. With the help of Hibernate ORM, developers can build applications whose data lives longer than the application process easily. As an ORM framework, Hibernate works in persistence layer when it is applied into relational databases. It is considered as a connection between Java application and relational database, for its ability to represent relational records through object oriented approach in Java language. With the help of Hibernate, developers can easily manage and manipulate the relational database [6]. Besides, Hibernate framework does not only support the mapping from Java class to database table (including the mapping from Java data types to SQL data types), but also provides object oriented data query retrieval mechanism. In this way, the development time on processing SQL and JDBC is greatly shortened. The workflow of Hibernate is shown in Fig .3.

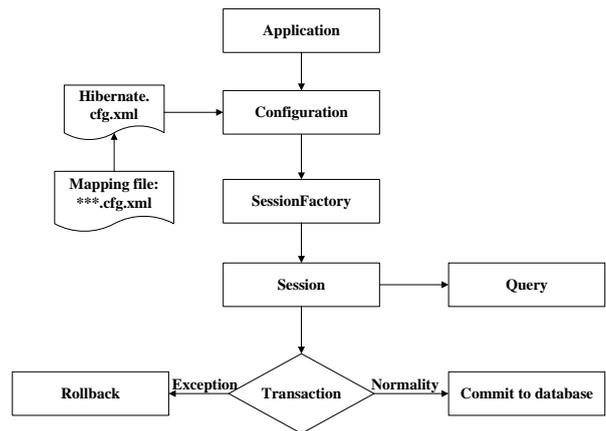


Figure 3. Workflow of Hibernate framework

III. THE INTEGRATION OF SSH

A. The integration of Spring and Struts2

In the web application, there is no need to manually create the Spring container. The Spring container can be automatically created and loaded by a Java class named ContextLoaderListener. It implements an interface named ServletContextListener. By setting ContextLoaderListener into element <listener/> in web.xml, Spring container can be initialized as soon as the web application started. In order to make Spring take charge of the object creation, developers must put the essential information about the object into the configuration files of Spring.

In the same way, Struts2 can be automatically constructed by class named FilterDispatcher which implements servlet filter interface. With the aim of making Struts responding to requests of clients, URL pattern should be put into the element<filter-mapping/> correctly [7].

For now, these two frameworks can function properly but respectively. It means that the Action object is still constructed by Struts2 itself, and the components inside each Action object is still created by hard coding, which leads to high coupling. In order to integrate Struts2 and Spring, Struts2 framework provides a plugin named struts2-spring-plugin.jar, which changes the object factory

used in Struts2 framework to the object factory of Spring framework. In other words, the Action object which is created by Struts2 to handle the corresponding requests is now constructed and managed by Spring container. The cooperation of Struts2 and Spring is shown in Fig .4.

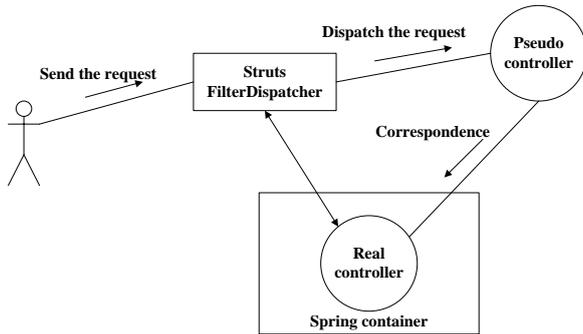


Figure 4. The integration of Spring and Struts2

B. The integration of Spring and Hibernate

Data Access Object (DAO) is a kind of standard design pattern in JavaEE. The core idea behind this design pattern is that, all the database access should pass through DAO component, which means class of DAO should encapsulate all the CRUD (Create, Retrieve, Update, and Delete) operations in order to make the system expandable and transplantable [8]. With the help of unified interfaces, DAO components can be implemented for different kinds of database as long as they all have Java Data Base Connection (JDBC). In this way, the cost of transplanting the web application from one certain database to another database can be minimized.

Spring framework provides various kinds of tools for developers to implement DAO components more easily. By inheriting the abstract classes supplied by Spring framework, people can simplify the development steps. For example, the class HibernateDaoSupport is a simple encapsulation class of a compiled memory image of the database mapping named SessionFactory in Hibernate framework. By setting the information of SessionFactory and DataSource into the configuration files of Spring framework, the Spring container will inject the references of DataSource for SessionFactory [9]. As a consequence, every Data Access Object which is inherited from HibernateDaoSupport class can use SessionFactory freely and correctly.

IV. THE SYSTEM IMPLEMENTATION

A. User access module

To simplify the design of user access module, all the operation authorities in the system can be divided into two categories: user management authorities and ancient fabric management authorities. Each category has four operations: Create, Retrieve, Update, and Delete. The users of ancient fabric information system can be classified into five categories:

- Administrator
- Ancient fabric expert
- Data collectors
- Graphic designer
- General user

Every user has different access level in the system according to which user category he belongs to. The database table designed for user access module is shown as TABLE I.

TABLE I. USER ACCESS TABLE

Name	Data Type	PK	FK
pre_permission_number	integer	Y	
pre_permission_type	varchar(100)		
add_user	tinyint(1)		
delete_user	tinyint(1)		
update_user	tinyint(1)		
query_user	tinyint(1)		
add_relic	tinyint(1)		
delete_relic	tinyint(1)		
update_relic	tinyint(1)		
query_relic	tinyint(1)		
design	tinyint(1)		

The permission verification works as soon as the user tries to carry out an operation in the system. The realization of this automatic mechanism is provided by Struts2's Interceptor class. Before the user's request is submitted to the corresponding Action, it will firstly enter the scope of Interceptor where the permissions inquiry will happen. Therefore, the requests that have no sufficient permissions will be rejected before going into the corresponding Action. The configuration of Interceptor is shown in Fig.5.

```

<interceptors>
  <interceptor name="permission"
    class="system.interceptor.PermissionInterceptor"/>
  <interceptor-stack name="permission-stack">
    <interceptor-ref name="defaultStack"/>
    <interceptor-ref name="permission"/>
  </interceptor-stack>
</interceptors>
<default-interceptor-ref name="permission-stack"/>
  
```

Figure 5. The configuration of Interceptor

It can be seen that all the access control work is done in the PermissionInterceptor class which extends from AbstractInterceptor class in Struts2 framework. After being configured as the default interceptor, PermissionInterceptor will take charge of all the requests by calling function intercept() before they reach the Action class. The workflow inside the method intercept() is shown in Fig .6.

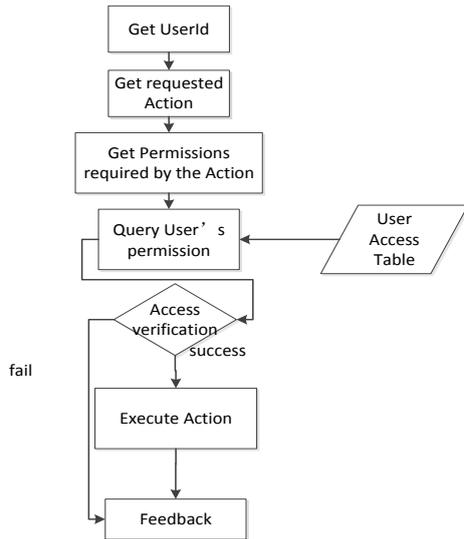


Figure 6. The workflow in function intercept()

The identity of the current user is obtained by looking up in a certain class named `ActionContext` which stores context data of the whole system. Then, the function `getAction()` is called upon to procure the instance of the Action requested by the user. All the Action classes in the system have implemented a certain interface named `PermissionList`, which contains a function needs to be implemented, named `getPermissions()`. After being invoked, the function returns a list of permissions required in order to access the Action class. By comparing the user's permissions and the permissions required in this operation, `Interceptor` decides whether the operation can be executed successfully or not.

By implementing the user permission verification in `Interceptor` class, system developers do not have to conduct their own permission verification within each corresponding Action. Therefore, the designers of Action class only needs to perform the corresponding operations logic, without concern of realizing the functions of the user permission verification, so that the coupling of the system is reduced greatly [10].

B. Ancient fabric information management module

The management of ancient fabric information includes the CRUD operations of the information. A complete structure of ancient fabric information is shown in Fig .7. Because of the high complexity of the structure, the information is divided into the following seven parts:

- **Basic information:** One piece of ancient fabric information contains only one block of basic information, which includes year, size and pictures.
- **Source information:** One piece of ancient fabric information contains only one source information block, which includes pictures, discovery time, and discovery place.
- **Fiber information:** One piece of ancient fabric information contains only one fiber information block, which contains several fiber information units. Each unit includes fiber unit description and the corresponding pictures.

- **Organization information:** One piece of ancient fabric information contains several organization information blocks. Organization information consists of organization structure description, embroiders, pictures and so on.
- **Pattern information:** One piece of ancient fabric information contains several pattern information blocks which include description of pattern and pattern pictures. Besides, each pattern information block contains several pattern information units and several framework information units.
- **Dye information:** One piece of ancient fabric information contains only one dye information block which is consisted of dye species and description. Besides, it may contain several dye information units.
- **Reference information:** Each piece of ancient fabric information has a record of reference information which specifies the source of all the ancient fabric information.

The classification of information shown above is also used in designing the database relation tables. By making the basic information table as the main table and the rest tables as sub tables that connect with main table by external key, the data redundancy can be minimized and kept up with the actual structure of ancient fabric information. The specific information of each table and the relation between them is shown in Fig .7 and Fig .8.

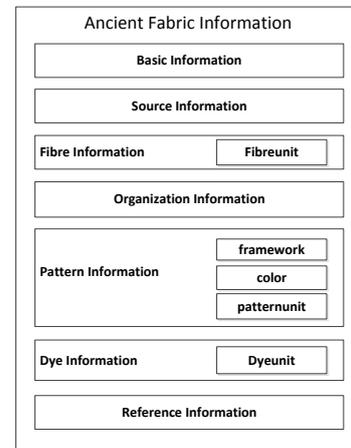


Figure 7. Structure of ancient fabric information

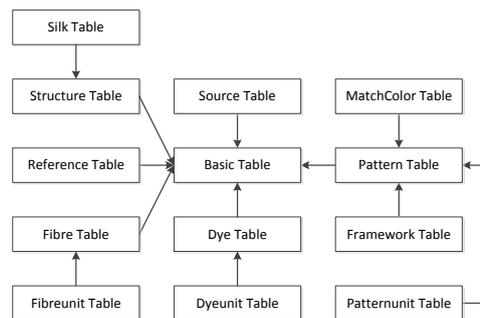


Figure 8. Relation between each database table

In order to realize the CRUD operations on ancient fabric information, the corresponding Action classes,

Business logic layer, Persistent layer and DAO must be provided.

1) *Action class*

With the aim of answering the requests of inserting, deleting, querying and updating, there are four corresponding Action classes named insertFabricAction class, deleteFabricAction class, queryFabricAction class and updateFabricAction class. Each Action class is an encapsulation of request parameters and response parameters, and they will call the corresponding functions in the Manager class to process the requested operations. As a result, an instance of Result class will be returned to help control the operation of the system.

```
List<Basic> viewBasic() throws HrException;
void addBasic(Basic basic) throws HrException;
Basic findBasic(Integer basicNumber) throws HrException;
void updateBasic(Basic basic) throws HrException;
void deleteBasic(Basic basic) throws HrException;
...
```

Figure 9. Methods in Manager class

2) *Business logic layer*

The Manager class in business logic layer provides service for each concrete Action class. It is an abstract encapsulation of DAOs and it hides the underlying details of data access operations. Each business logic method is consisted of several DAOs as a result of the complex structure of the ancient fabric information. The core work of the Manager class is to properly decompose and package the information into the corresponding DAO. And DAO will do the persistence work for the Manager class. By calling the methods in the Manager class shown in Fig .9, the intricacy information of ancient fabric is disassembled into several parts and stored in each database table properly.

3) *DAO class*

There are thirteen DAO classes according to the thirteen database tables. Each DAO class provides data manipulation interfaces for upper layers. Spring framework provides users a powerful tool class named HibernateDaoSupport to simplify the implementation of DAO class. In order to make HibernateDaoSupport work functionally, it must be managed by Spring container so that the reference of SessionFactory can be injected into the class correctly. After that, every DAO class which extends HibernateDaoSupport can get the reference of HibernateTemplate by calling the method named getHibernateTemplate(), and use the methods: get(), save(), update() and delete() to realize the database transection easily. The structure of DAO is shown in Fig .10.

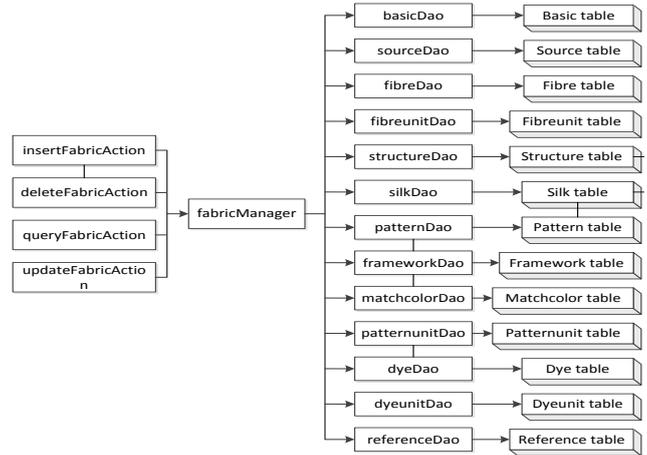


Figure 10. Structure of Data Access Object

4) *Persistent layer*

Each database relation table has a corresponding Persistent Object, with which form the Persistent layer together. Hibernate uses a configuration file as mapping the correspondence between the relation table and Persistent Object. Persistent Object is POJO (Plain Old Java Object) except that it must have the following elements:

- Non-parameter constructor.

- Getter and setter methods for all the attributes in the class.

- Overriding of methods named equals() and hashCode().

Each database relation table has a corresponding Persistent Object, with which form the Persistent layer together. Hibernate uses configuration file as follows:

5) *System log module*

System log module is an extremely important part in the whole system. It can record all the actions arising under the system and show them in a certain paradigm. The log system can be used to record information for system troubleshooting and system performance optimizing as well.

In Struts2 framework, there is a component called interceptor which can be used to help build the log system module. Interceptor class and Action class are in the same execution chain in the system if we set corresponding Interceptor class in configuration file of Struts2 framework. Before the request from client is processed by the corresponding Action class, the control will be transferred to Interceptor. Similarly, after the execution of each Action class, the result will return to Interceptor class where system can record the raised action result and other relevant information.

After realization the log function in the method intercept() of PermissionInterceptor class, we need to enable it in struts.xml as shown above.

V. CONCLUSIONS

This paper focuses on the design and implementation of user access module, information management module and system log module, and provides the flow charts of the key business.

With the help of the MVC design pattern in Struts2 framework, each component of the ancient fabric information system is code independent and low coupling. The AOP of Spring framework binds the system level service to the business logic, and the management provided by the IOC container makes the dependency relationship between Java objects more clearly. Hibernate framework allows developers to easily implement the relevant operations of relational database persistence layer by means of POJO [11].

The system adopts hierarchical layer concept which is popular in modern design. According to this concept, upper layer calls the interfaces implemented by the lower layer, and the lower layer providing service for the upper layer. When the specific techniques of a certain level changes, the other layer does not require any change as long as they share the same interface.

The ancient fabric information system is developed based on the integration of three frameworks, which together brings the whole system better maintainability, low coupling, and speeds up the development process of the system as well.

ACKNOWLEDGMENT

This project is supported by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant 2013BAH58F02.

REFERENCES

- [1] Wang J W, Han L P, Zhang J M. Design and implementation of material management system of oilfield based on multi-layer architecture[J]. *Computer Engineering and Design*, 2011, 1: 162-165.
- [2] Li K, Yang K, Zhu S. A web management platform of Internet-based electrical engineering lab: Using SSH framework[C]//*Computer Science & Education (ICCSE)*, 2012 7th International Conference on. IEEE, 2012: 1439-1443.
- [3] Yu Z L, Li Z Q, Ren Y C. E-commerce Software Platform of MVC Architecture Based on SSH Framework[J]. *Computer Technology and Development*, 2012, 22(10): 169-172.
- [4] Chen K, Liu Q, Ai Q. Design and implementation of employee education and training system based on SSH framework[C]//*Software Engineering and Service Science (ICSESS)*, 2013 4th IEEE International Conference on. IEEE, 2013: 467-470.
- [5] Guo W L, Jiang H J, Liu S G. Based on SSH framework RBAC design and Implementation[J]. *Computer Engineering & Software*, 2011, 32: 47-48.
- [6] ZHANG J, LIU H, NI F, et al. Research and Application of Web Application Architecture Based on SSH[J]. *Journal of Natural Science of Hunan Normal University*, 2012, 6: 009.
- [7] Xie J, Xu L F. The Research of Agile Web Development Process and Quality Model[M]//*Network Computing and Information Security*. Springer Berlin Heidelberg, 2012: 446-452.
- [8] Sun Y. Design of Online Examination System Based on SSH Framework[C]//2015 International Conference on Advances in Mechanical Engineering and Industrial Informatics. Atlantis Press, 2015.
- [9] Zhenrong L, Junyun W, Zhitao C, et al. Research on Code Automation Based on SSH Framework[J]. *Science Mosaic*, 2014, 2: 008.
- [10] Liu H. Design of Information Interaction System Based on Android and JAVA EE architecture[C]//First International Conference on Information Sciences, Machinery, Materials and Energy. Atlantis Press, 2015.
- [11] Zhang X, Xiong Z, Wang G, et al. Services Management Model Based on the SOA[J]. 2015.