

Research on the Design and Realization of the Framework of Library Management System Based on JSF and Hibernate

XU Jufang ZHAO Zhengwen
Graduate School, Southwest Petroleum University, P.R.China, 610500

Abstract With the rapid development of network technology, Java Server Faces and the key technology of Hibernate are in urgent need in Web application which is based on the B/S architecture. This paper analyzes how to use Java Server Faces and Hibernate framework to design and realize the library management system, and then gives its framework including architecture, function and design.

Key words Library Management System, JSF, Hibernate, MVC (Model View Control).

1 Introduction

Java Server Pages mingles words and HTML label presenting pages that belong to the static content with labels producing dynamic content while dealing with web applications based on the B/S structure. The problem brought by this method is that the logic of dealing with the User Interface state must be needed in webs. But JSF which grows out of JSP and presents in webs in a way of special labels is able to use the third party or the user-defined special present layer and enclose program logic in order to reduce the modification of present layer's code. At the same time, as for the persistent layer which is between the logic layer and the data layer, professional knowledge is required to realize robust persistent layer by JDBC coding directly. Besides, it is impossible for people dealing with enterprise application and exploitation to develop one's own persistent layer from the beginning. But it is easy to combine applications with systems to achieve the persistent layer if you choose Hibernate belonging to the open code and ORM software.

This paper advances a framework based on JSF and Hibernate to exploit library management system, and takes it for example to prove this way is feasible.

2 Advantages of JSF and Hibernate

2.1 Comparison between JSF and Traditional technology

What's JSF? It's an event driven model based on components. It is similar to the exploitation model which is independent of GUI applications. It brings component model that is used to be applied in GUI applications to Web application system exploitations. Compared to the traditional technology, JSF has the following advantages.

2.1.1 Reduction of code in UI module

User Interfaces of Java Web application systems used to be composed of JSP pages. It mingles the static content with the dynamic one. The problem brought by this way is that it must include logic dealing with the UI state in webs. For this reason, JSP increases code in UI module. While JSP is looked as present layer technology in JSF, JSF components are present in webs in a way of special labels so that it can use the third party or user-defined special present layer. The following web code segment shows role authority distribution by a group of HTML checkboxes.

```
<h:selectManyCheckbox value="#{roleManageListBean.funcValue}">
    <f:selectItems value="#{roleManageListBean.funcItems}" />
</h:selectManyCheckbox>
```

In the above code, there are neither circles nor conditional estimations. All of these have been encapsulated in JSF components represented by these two special labels <h:selectManyCheckbox> and <f:selectItems>.

2.1.2 More modularized UI code

JSF components provide MVC model, and Figure 1 shows the attributes of applied objects. JSF components declare events possibly produced by properties and Event Listener matching the component

to deal with events. Event Listener has rights to define the attributes of JSF components. Matching different Renders, the JSF UI component can take on a new look including buttons or links or even other labels.

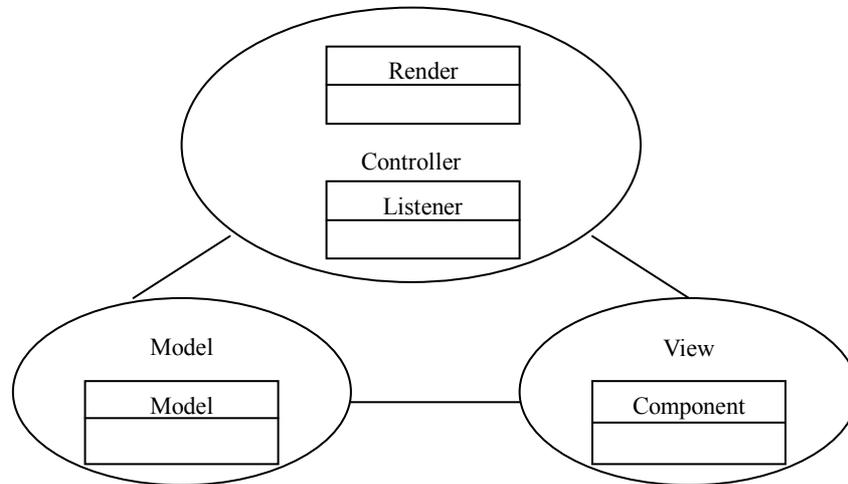


Figure 1 MVC design in Java Server Faces

Besides UI components, JSF also defines secondary tool objects, including validators that can test and verify the validity of input data and converters that can convert data type. The result is producing a very modularized UI program model. Developers can use familiar GUI framework to exploit Web application systems which are easy to maintain.

2.2 Analysis of Hibernate

Hibernate offers larger and more functional persistent services between objects and relation database. Developers can do persistent layer exploitations according to Java basic semantic by using Hibernate. Hibernate Query Language offered by Hibernate is a kind of facing objects query language. It provides a rapid, effective and convenient channel between object database and relation database.

Essentially, hibernate is a middleware offering database service. It plays an important role in connecting Java application with relation databases. Compared with other ORM tools, Hibernate has its own advantages and characteristics.

Hibernate which is a light object encapsulation of JDBC is an independent object persistent layer frame. While JDO is a heavy encapsulation of JDBC. So Hibernate can be used where JDBC is applied.

Hibernate which has a good rate of code reuse is fit for every kind of application field. But other ORM tools are related to business function obviously and applications vary from business to business.

Hibernate is closely relevant to JDBC driver and database, but it has nothing to do with Java applications or App Server. There are no compatibility problems.

3 System realization based on Java Server Faces and Hibernate

At present, the system is realized by using Java Server Pages and JDBC. The main problems of the system caused by the disadvantages of JSP and JDBC are that webs can't use dynamic labels and user-defined present layer and that there is much redundancy in the code accessing database. All problems like these lead to lower efficiency, harder maintenance and so on. The primary reasons why the author uses the framework based on JSF and Hibernate to realize this system are as follows:

Firstly, it may use exploitation model of MVC. Fortunately, JSF itself has offered the MVC model and connected View and Model so as to improve exploitation efficiency, system maintenance and expansion.

Secondly, it simplifies the access to database by Hibernate framework.

3.1 System frame

The library management system should realize the following requirements:

Firstly, books, magazines, readers and librarians should be registered in the system.

Secondly, readers can borrow books from the library in case that they satisfy certain conditions. And readers are able to book books or magazines that can't be borrowed from the library.

Thirdly, new books are brought by the library. At the same time, old ones or outdate ones are abandoned or done with by the library.

Fourthly, librarians are responsible for books' management and can add, update or delete the information about books and readers.

Lastly, the system should have a good extension and a perfect Graphic User Interface. Besides, it can execute on all the platforms that are currently popular environment such as UNIX, Windows, OS2 and so on.

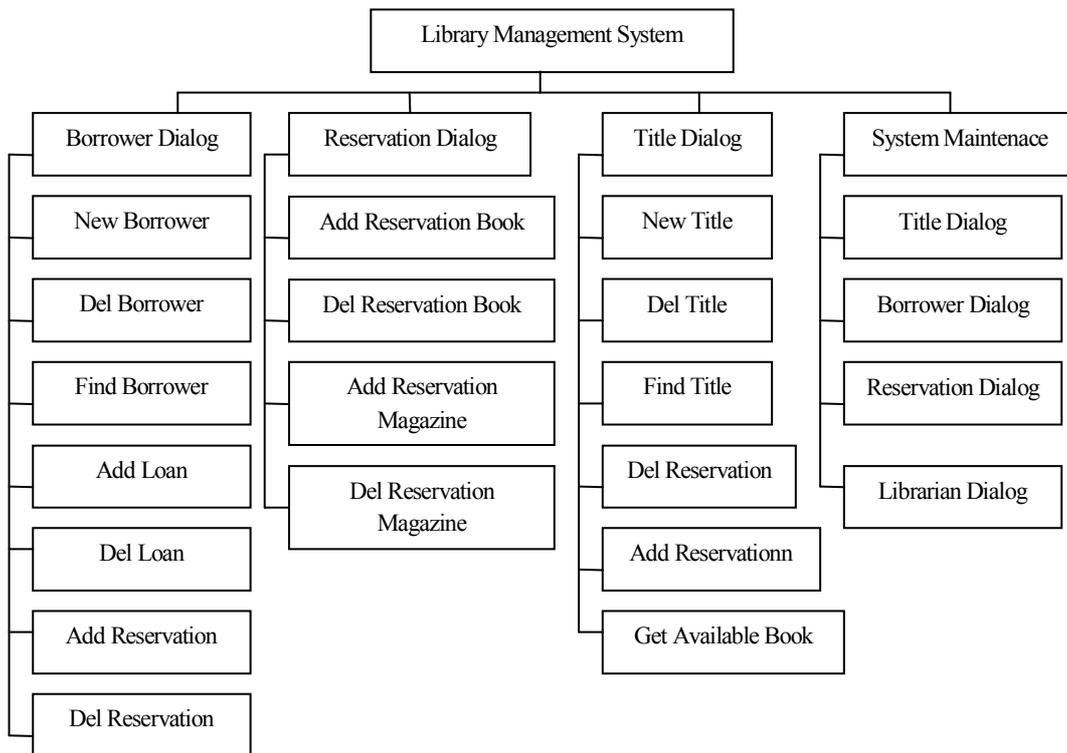


Figure2 System function structure chart

Figure 2 shows the system function structure. The system contains four sub systems consisted of Borrower Dialog, Reservation Dialog, Title Dialog and System Maintenance.

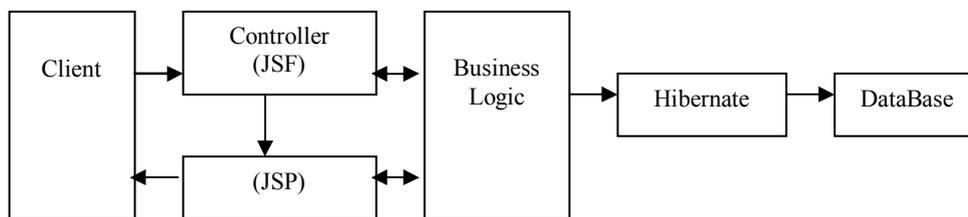


Figure 3 System architecture

Figure 3 gives the System structure. Respectively, it shows the layer which corresponds to application, present, business logic, persistent and database layer .

3.2 Technological difficulty: how to take on error information alarm boxes

According to Figure 3, error information alarm box also conforms to the basic principle of the whole system structure. The key point of the technological difficulty is that how to make error information alarm box take on when an error occurs in the current page, and then make it return to the initial page after the confirmed button is pressed, and finally the state of the page must be consistent with the initial one.

3.3 Analysis of and solution to the difficulty

Firstly, the error information alarm box belongs to a type of web. So it's easy to design the layout of the error page. The relevant key code is given as follows.

```
<h:outputLabel value="#{Info.msgInfo}" />
<h:commandButton value="confirm" action="#{Info.OK}" rendered="true"
image="images/ok.gif"/>
```

The value property of <h:outputLabel> is bound with the parameter of the backing javabean. By doing so the error information is under the logic control of the current page. This principle also goes with <h:commandButton>.The specific code of the javabean corresponding to error information above is as follows.

```
package swpu.tsg.UI;
import javax.faces.context.FacesContext;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.log4j.Logger;
import swpu.tsg.util.Messages;
public class InfoBean {
    protected static Logger log = Logger.getLogger("swpu.tsg.UI.InfoBean");
    protected Messages messages = new Messages();
    protected String message; // just for jsf binding
    protected String exInfoMsg; // for extra info binding
    protected String errKey = "dummy";
    public String getMessage()
    {
        FacesContext ctx = FacesContext.getCurrentInstance();
        errKey =
        (String)((HttpSession) ctx.getExternalContext().getSession(false)).getAttribute("Key.PopInfo");
        // TODO too expensive, delete this log
        log.debug("getting message: " + Messages.getMessage(errKey));
        return Messages.getMessage(errKey);
    }
    //getter and setter
    public String getErrKey() {
        return errKey;
    }
    public void setErrKey(String errKey) {
        this.errKey = errKey;
    }
    public void setErrMsg(String key) {
        message = Messages.getMessage(key);
        log.debug("set message: " + message);
    }
    public String getExInfoMsg() {
        FacesContext ctx = FacesContext.getCurrentInstance();
        String tmpInfoMsg =
        (String)((HttpSession)
```

```

ctx.getExternalContext().getSession(false)).getAttribute("exInfo.PopInfo");
    if (tmpInfoMsg != null) exInfoMsg = Messages.getMessage("detail.PopInfo") + tmpInfoMsg;
    else exInfoMsg = null;
    log.debug("get exInfo: " + exInfoMsg);
    return exInfoMsg;
}

```

Secondly, adding the following java script code to the error information alarm pages can make the current page return to the previous one.

```

<script language="javascript" type="">
function Return1(URL){
history.back();
}
function closeWin(){
window.close();
}
</script>

```

What is most important is to make sure the type of the http session involved in the page where error occurs belongs to the request type. Otherwise, if the http session belongs to the dialog type, you have to clear the javabean after accessing it to keep the state of the property refreshed when you access the page the next time.

In addition, to solve the problem completely, you still have to do the configuration about faces-config.xml, faces-config-beans.xml, web.xml, message.xml. Faces-config.xml is to point out webs' navigation. Faces-config-beans.xml is to point out javabeans' navigation. Web.xml is to define the time of session's lifecycle. And message.xml is to define the kind of error and the information to show on the error information alarm box.

3.4 System realization

The functions of this system have almost been realized. Users can enter the system by the login page. In the course of using the system, the expected exploitation effect is really satisfied. It has solved the problems existing in the old one. Of course, there are still certain aspects to be improved.

4 Conclusions

By exploiting this system, it has been proved that exerting JSF and Hibernate framework to do web application system exploitation can shorten the exploitation circle of software efficiently, simplify applications and improve the exploitation quality of exploitation to make the system have good mutual character and better maintenance.

References

- [1]Tian ke, Xie Shibo, Fang ma .Settlement of J2EE Data Persistent Layer [J]. Computer Engineering ,2003,29(22):93~95
- [2]Deepak Alurm .Core J2EE Patterns, ISBN 0-13-064884-1, Sun Microsystems Inc, 2001
- [3]Nicholas Kassem et al .Designing Enterprise Applications with the J2EETM Platform. Sun Microsoft systems ,Inc,2000
- [4]Xia Xin, Cao Xiaogang, Tang Yong. Explain the profound things about Hibernate[M], in a simple way.(2nd.ed)Beijing: Electronic Industry Press,2005.6(in Chinese)
- [5][American]Doderny wrote, Sun Yong, Cai Yunzhi translated. Mastering JavaServer Faces [M]. Beijing: Electronic Industry Press,2005.6(in Chinese)
- [6][American]DavidGeary, Cay Horstmann wrote, Wang Jun, Ma Zhenping translated. Core Coding of JavaServer Faces[M]. Beijing: Electronic Industry Press, 2005.4(in Chinese)
- [7]HansBergsten wrote, O'Reilly Taiwan Company compiled. Mutual Webs Design of Java Server Faces[M]. Dongnan University Press, 2006.1(in Chinese)

Download Citation on ResearchGate | Research and Design on Library Management System Based on Struts and Hibernate Framework | If some small libraries use traditional library management system, they maybe waste resources. In order to this problem, a library management...Â Ge Zhen, Qu Pei, " Identity Verification System of Website Based on Struts and Hibernate Framework, " Software Guide, vol. 9, 2007, pp. 13-17. Implementation Model Framework of MVC Based on Hibernate. Jan 2007. 309-311. Wang Qingsong. Wang Qingsong, " Implementation Model Framework of MVC Based on Hibernate, " Journal of Liaoning University (Natural Sciences Edition), vol. 34, 2007, 309-311. If some small libraries use traditional library management system, they maybe waste resources. In order to this problem, a library management system based.Â For IEEE to continue sending you helpful information on our products and services, please consent to our updated Privacy Policy. I have read and accepted the IEEE Privacy Policy. Now coming to JSF, JSF is framework for dynamically generating HTML on the server side. There you can dynamically generate HTML elements directly based on your Java objects. The JSF model has fallen out of favour. It is viewed as a legacy framework against todayâ€™s Javascript frameworks with RESTful API backendâ€™s. This has moved the Java to implementing the RESTful microservices. This approach can often scale better than JSF. The key strength of JSF lies in the component frameworks in the JSF ecosystem. In particular PrimeFaces, or the utility libraries like omnifaces. They let you quickly get