The Page Flow Implements Business Logic In Web Applications

Haijun Li
Hexi University Information Technology Center
HXU ITC
Zhang Ye,Gansu,China
Lihj20040516@163.com

***Abstract*-This paper discusses the application of the technology of NetUI flow page in the business logic of Web applications. The page flow uses declarative programming method, and has session state and modular design, which can be nested to use and effectively separate the logical code in the Web applications, to make the application structure clear, easy to be read and modified. The paper references a business logic of the user's login authentication to illustrate the implementation process of the entire page flow.**

Keywords-Page Flow;Implements ;The Business Logic ;In Web Applications;

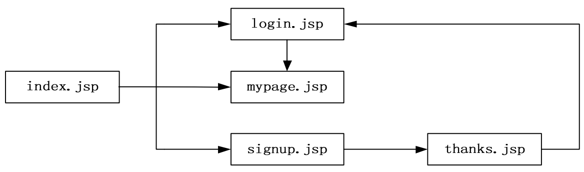
I. INTRODUCTION

Traditional Web applications need to write a lot of business logic, the presentation layer and business logical layer can not be completely separated. When the Web application program is expanding, the application program will become difficult to be read and can not be modified, leading to poor portability. NetUI page flow technology can effectively solve this problem. It can separate the presentation layer and business logical layer of the web applications, and the first one is completed by the JSP page, and the latter is by the page flow controller.

II. THE MODE OF PAGE FLOW IN WEB APPLICATIONS

1) Logical Page Flow

A lot of logic codes are required to be wretten in a treditional web application the application. For instance,when providing the site of "my page" for the registered users, the site's home page needs to implement more complex logical functions, which decide whether to enter the authentication form or to the user's customized page directly, the logical function is shown as Fig1:



Fi
gure1. Logical page flow

The logical flow supports multiple paths from the home page of the applications to the user's mypage.jsp page:

- The user may directly navigate from index.jsp to mypage.jsp (by clicking a link), if the user has already logged in.
- If the user has not already logged in, her attempt to navigate from index.jsp to mypage.jsp will be intercepted and the user will be taken to the login.jsp instead. After successfully logging in, the user will be automatically taken to mypage.jsp
- The user may directly navigate from index.jsp to login.jsp (by clicking a link). After logging in, the user will be automatically taken to mypage.jsp . In the event of a login failure, login.jsp will be redisplayed to give them another opportunity to authenticate themselves.
- If the user desires to register with the site, he can click a link that will take him to signup.jsp . After signing up, the thanks.jsp will be displayed ,which offers a link to the login.jsp page.

When the Four business logical processes are achieved in the standard of html page, they need to be judged through business logic and every page is directly connected to the other pages.

When using the page flow, business logic is achieved by the behavior between pages and behavior interchanging. This case is shown as Fig2

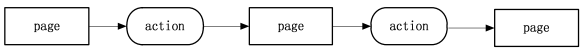


Figure2：Action Navigate

The business logical functions need to be separating from the web site so that a page flow can be used and the index.jsp.Home.jsp page of the applications isn't connected to the login.jsp page or user's mypage.jsp,but are connected to the Java code,which decides how to the application, made by the java code to which decides how to connect. Is shown as Fig3:

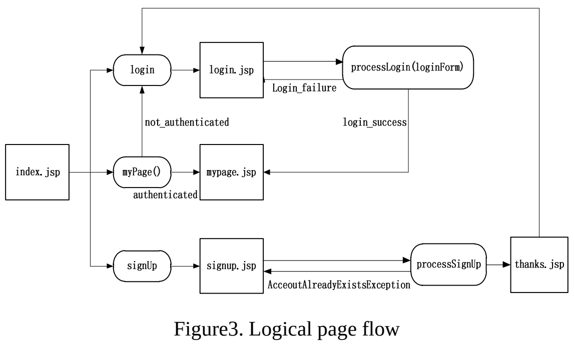


Figure3. Logical page flow

In figure 3, index.jsp is navigated to mypage.jsp during which the user needs to through mypage behavior. The perform to this behavior must to test and determine whether the user has been through authenticated . If the user has logged in, it will guide the user to mypage.jsp directly;

otherwise, it will guide the user to login.jsp.

2) Page Flow Programming Mode

Page Flow is a part of NetUI . NetUI is the piece which Beehive used to build the front-end of a web application. It contains two pieces: Page Flow and a powerful set of JSP tags. Page flow can help you build a well-structured web applications and separate navigation control from the page.Thus,the reuse of navigation and page flow are avoided and make jsp source code clear and easy to be maintained and easier to design business logic.

Page Flow programming model allows you to create modular page flow, it puts the control logic, description and meta data together into a single application, and through inserting / reusing page flow in to achieve the page flow module .

III. THE PAGE FLOW ACHIEVES BUSINESS LOGIC

1) To Create the Controller Class

The business login of Page Flow can be achieved through the controller class. The way of creating the controller class is like the way of creating java class.The use of inheritance to create a controller class. The code is shown as follows:

```
Examples:
import org.apache.beehive.netui.page flow.Page Flow Co-
ntroller;
import org.apache.beehive.netui.page flow.annotations.J-
pf;
@ Jpf.Controller
public class Controller extends Page Flow Controller
{
}
```

In the code, the use of metadata annotation can compile the controller class, and every controller class needs to use the @jpf.controller annotation to modify. The @ jpf.controller Notes Tip compiler, which is a special class of the page flow control, rather than a typical class of java.

As the Page flow model shown in Figure 2, it has 6 behaviors, including login, mypage, signup, processlogin, processsignup.These behaviors decide which jsp will be shown.The method of Begin is integrant behavior method of the controller, and every the controller class will create these methods.

2) To Implement Business Logic

There are two ways to implement the business logic of Page Flow: Simple Action and Action Method. Simple Action is an annotation of class level , and it is used for decorating the controller class.It can complete the navigation, form submission, form confirmation and other tasks, but it can not execute the decision logic, if the behavior in the Web application needs decision logic and these decisions are executed by the conditional code. Action Method is used to behavior. The grammatical structure of the Simple Action is as follows:

```
    examples
@ Jpf.Controller ( simple Actions =
                 {@ Jpf.Simple Action (name = "some-
                  Name",path = "some Page.jsp", [...-
                  other properties ...])
                                 }
                                 )
```

In the process of the complex business, decision logic is completed by the Action Method.Action Method is a method of java which is given to functions of all methods , the method is capable of navigating users by encircling the page flow, handling the form submission, processing the decision logic and so on. An Action Method is a method of Java, it returns to the type of forward and uses @ jsp.action to annotate and modify.The grammatical structure is shown as follows:

```
examples
@ Jpf.Action ( forwards =
          {@ Jpf.Forward (name = "some Name",-
           path = "some Path.jsp", [... other prope-
           rties ...])
          }
          )
```

The Jpf.action and Jpf.forward annotations are used on each action method to build a mapping between forward names and jsps ,and the method of the behavior turns to show its corresponding jsp page according to the contents of forward.name.

a) Simple Action to Achieve Business Flow

According to the design model of logical flow , begin, login and signup are the simple navigation behaviors, which can use Simple action to achieve.To achieve the Simple action page flow the @ jpf.Simpleaction label can be adopted.The@ jpf.simpleaction label defines a set of behavior and the decision path of jsp. When a special act was instituted, the user will be brought to its corresponding jsp page of the decision path.

The controller class requires a simple action called begin,otherwise this class will not be compiled.The functions of Begin behaviors is an entry point into the flow of the page, and begin behavior can easily navigate the user to the index.jsp page, the code is shown as follows:

```
examples
@ Jpf.Controller (
simple Actions = {
             @ Jpf.Simple Action (name = "begin",
               path = "index.jsp"),
             @ Jpf.Simple Action (name = "login", path
               = "login.jsp"),
             @ Jpf.SimpleAction(name= "sign Up",
               path = "sign up.jsp"),
              }
              )
```

b) Action Methods to achieve business flow

The Mypage affairs which have decision logic need action method to achieve. To creat a method for it, the mypage behavior must make decisions.According to the different results to make different the decisions. If the user is authenticated, the navigation of page flow will display mypage.jsp page, if not, it will be transferred to execute the simple act of login, the logic is shown in Fig4.

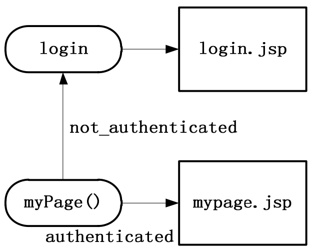


Figure4. the method of decision logic

The method of Action method must be the type of forward , and must be modified by @ jsp.action annotation.The method of mypage will give a forward object back, and @ jpf.action will display corresponding page to the user according to the contents of the object of forward .

The annotation of jpf.action and jpf.forward are applied to establish a forward name corresponding with the path of jsp for the method of the behavior of each line.Based on the contents of the forward name the method of behavior, then turns to show corresponding jsp page of the path .

By defining two forward names the navigation behavior is added to the controller class, two forwards are authenticated and not_authenticated, in which the authenticated is mapped to mypage.jsp and the not_authenticated is mapped to login.do respectively.

examples

```
public Forward my Page ( )
{ Http Servlet Request request = get Request ( );
  Http Session session = request.get Session ( );
  if (session.get Attribute ("authenticated_user")! = null)
      { return new Forward ("authenticated"); }
  return new Forward ("not_authenticated");
  }
```

The My page method() in the program code causes two navigating results, one is showing my page if the users get authentication .The other is turning to the simple behavior of the login and displaying the page of login.jsp.The method decides which forward value is returned. Through detecting a session attribute is executed or not, Authenticated decision can judge whether the attribute of authenticated_use is set.

c) To Process the Submitted Form

The user must submit personal information in the authentication or registration information.By providing a parameter to the controller method, html form data will be displayed to generate.Then create a java bean to complete the html form submission.The java bean can be used as a controller's own internal definition of the static class , and also can be used as a separate class defined in a document.

The method of Defined process login () obtains a parameter's back from loginfrom, and according to the returning parameters to control the decision-making logic. If the user has registered a correct user's name and password, they will be taken to my page.jsp directly, otherwise, they will be returned to the login.jsp and try it again. In the code, using the class of myapputils to check whether the user's name and password is legitimate, the code is shown as follows:

examples

```
public Forward process Login (Login Form form)
 {
 if (My AppUtils.authenticate (form.get Username( ),-
     form.get Password ()))
 {
 Http Servlet Request request = get Request ( );
 Http Session session = request.get Session ( );
 session.set Attribute ("authenticated_user", form.get-
 Username ());
 return new Forward ("login_success");
 }
 return new Forward ("login_failure");
 }
```

d) To Handle Exceptions

In the business logic, when a new user completes the registration and submits the registration information, but when handling it,finding the user's information has already existed, the user needs to fill out the registration information again.At this moment, the behavior of processisgnup can be designed to throw out an exception,and then the exception of the controller class can back to the original registration page.Use the @ jpf.catch and @ jpf.excetpion hander labels to achieve exception handling.

examples

```
@ Jpf.Controller ( catches = {@ Jpf.Catch (type= Accou-
untAlreadyExistsException.class,method="handleAccou-
ntAlreadyExistsException")})
```

The @ jpf.excetpionhander comment section is defined in the controller class.

examples

```
@ Jpf.ExceptionHandler ( forwards = {
@ Jpf.Forward (name = "signup", path = "signup.jsp")
                                              }
                  )
```

IV. SUMMARY

Page Flow uses a simple programming model to establish a well-structured web applications,which is able to achieve the complex business logic of Web applications.The business logic is easy-implement, clear-code, readable, and easy-transplanted .The design of Page Flow holds the idea of modular design which can be reused, and provides a practical method of design for the development of the large-scale Web applications.

REFERENCES

[1] Hu Qimin,Xue Jinyun,Zhong linhui. Lightweight J2EE architecture based on spring framework and its application[J].Computer Engineering and Applications,2008,(5):115-118.

[2] Zhai Gaoyue, The Research of Web Development on Spring Web Flow[J].Microcomputer Applicatons,2011,(1):47-51.

[3] Kou Yi,Wu LiWen,Application Methods of Struts Framework Based on MVC Designed Pattern,Computer Applications[J].2003,(11):91-93.

[4] Xu Lixin,Zhang Chunhua,Guan Yuling, Design and Implementation of Graphical Tool Based on Web Page Flow,Computer Programming Skills & Maintenance[J].2012,(6):59-60

[5] Li Xiongwen,Fang liang,Zhang Shufang, The Feature Analysis and Comparison between Three Kinds of Web Structures Based on Three-tiered Architecture[J],Applicaton and Research of Computers.2001,(8):61-63

[6] Ed Roman,Rima Patel Striganesh,Gerald Brose,Mastering Enterprise JavaBeans[M].Beijing:Publishing House of Electronics Industy,2005

[7] Thomas Van de Velde, Beginning Spring Framework 2[M], John Wiley & Sons Inc,2007

[8] Johnson R,Hoeller J,Aredsen A.Spring Java/J2EE application Framework[R],2004.