

基于 Java 的网站框架模板技术的研究与应用

宋大承 夏清国¹

(西北工业大学软件学院; 西安 710065; 西北工业大学计算机学院¹, 西安 710072)

摘要 使用框架模板的网站开发技术易于提高网站的开发效率和可维护性。提出了应用 MVC 模式, 采用布局管理器思想的 TAGLIB 技术, 使用 Java 语言实现网站框架模板的方法, 并结合实际介绍了系统设计和实现的关键技术, 对同类系统的开发具有一定的指导作用。

关键词 Java MVC 模式 TAGLIB 技术

中图分类号 TP311.12; **文献标识码** B

随着互联网和电子商务的不断发展, 大部分的网站, 尤其是商业网站的规模越来越大, 导致了网站的开发以及维护成为了目前的一大难题。

应用框架模板的网站以其易于开发和维护的优越性, 在互联网和电子商务的竞争中越来越显现其优势。应用 MVC 模式, 采用标签库 TAGLIB 技术, 使用 Java 语言可以很好地实现框架模板, 这样就可以把网站轻松地应用到框架模板当中, 从而可以提高网站的开发效率和可维护性。并且实现了页面的显示与页面功能的分离, 易于实现网站的风格的统一。

1 MVC 模式概述

MVC (Model/View/Controller) 模式是一种目前广泛流行的软件设计模式, 正在成为在 J2EE 平台上推荐的一种设计模型。MVC 包括三类对象: 模型 (Model)、视图 (View) 和控制器 (controller)^[1]。如图 1。

模型 (Model) 表示业务数据或者业务逻辑, 就是业务流程/状态的处理以及业务规则的制定。业务模型的设计可以说是 MVC 最主要的核心。

视图 (View) 是应用程序中用户界面相关的部

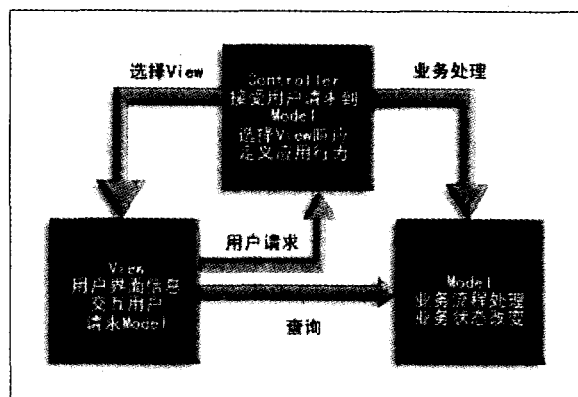


图1 MVC设计模式

分, 是用户看到并与之交互的界面。对于 Web 应用来说, 可以概括为 HTML 界面。

控制器 (controller) 可以理解为从用户接收请求, 将模型与视图匹配在一起, 共同完成用户的请求。一个模型可能对应多个视图, 一个视图可能对应多个模型。

MVC 模式对于 Web 应用的开发是一种非常先进的设计思想, 无论选择哪种语言, 无论应用多复杂, 它都能为理解分析应用模型提供最基本的分析方法, 构造清晰的设计框架。使用 MVC 模式, 可以避免用户界面设计将这些对象混在一起, 而是将它们分离以提高灵活性和复用性。

2 基于 Java 的网站框架模板的原理

2.1 Java 中的布局管理器

在实际编程中,每设计一个窗体,都要往其中添加若干组件。为了管理好这些组件的布局。需要使用布局管理器。将加入到容器的组件按照一定的顺序和规则放置,使之看起来更美观,这就是布局,在 Java 中,布局由布局管理器(LayoutManager)来管理。

Java 中的容器类(Container),它们只管加入组件(Component),使用自己的 add()方法向自己内部加入组件。同时记录这些加入其内部的组件的个数,可以通过其提供的方法类获得组件的数目和相应组件的句柄。然后 LayoutManager 类就可以通过这些信息来实际布局其中的组件了。在大多数情况下,综合运用好这些布局管理器已可以满足需要。

2.2 使用 Template Taglib 实现网站框架

参考了 Java 中的布局管理器,在页面中,可以用 Template Taglib 来实现 Java 中布局管理器的作用。让需要统一风格的页面使用同一“模板(Template)”,这样就可以通过改变“模板”来实现页面风格的改变。当有更多的页面使用同一“模板”时,只须改变“模板”就可以完成所有页面风格的改变,从而易于大型网站风格的统一,易于大型网站的开发和维护。

2.2.1 Template Taglib 所包含的标记

Template Taglib:

templateTag extends TagSupport 用来引用模板文件。

pageTag extends TagSupport 用来在模板的相应位置动态包含文件。

contentTag extends TagSupport 用来在模板文件中标记需要动态添加的文件。

includeTag extends TagSupport 用来在模板文件中静态包含一个页面^[2]。

2.2.2 Template Taglib 工作流程

首先执行 templateTag 在 doStratTag 中仅返回 EVAL_BODY_INCLUDE,此时执行 pageTag,用 pageContext.setAttribute(name,file,pageContext.SESSION_

SCOPE)设置以 file 为内容,名为 name 的在 session 范围内的属性,执行完所有 pageTag 后,执行 template 的 doEndTag,将模板文件 temp 包含进来,并返回 EVAL_PAGE,这样就会执行模板文件。在模板文件中如果遇到 contentTag 时,用 pageContext.getAttribute(name,pageContext.SESSION_SCOPE)返回所需的添加的 file,并将其 include 进来。在 templateTag 中 include temp 时,应该是直接包含其运行结果,所以 attribute 的存在范围是 SESSION_SCOPE。另外,includeTag 只是在 temp 中直接 include 一个页面,和<%page include="*. *"%>效果一样。使用这个标记,也是考虑到更加容易维护页面风格的统一和灵活地改变风格,比如改变网站的 LOGO,广告条等。

2.2.3 模板文件

模板文件是一个由 Table 构成的框架,其中可以包含静态文件,如广告条,Logo,Link 等,这些也可以动态处理;但主要的是有关业务的动态文件,例如 NEWS,SHOPPING 等内容。但不同模板就可能有不同多个页面需要添加,这就根据每个页面的具体业务逻辑而定。另外如何将动态添加页面放入模板正确的位置,我们通过 pageTag 中的 name 属性和 contentTag 中的 name 属性来确定。

2.2.4 模板应用文件

模板应用文件是对模板文件的填充,根据模板文件中 contentTag 标记的位置和“name”,添加相应的文件。例如:

```
<html><body><template temp='模板文件名'>
<page name=" " file='动态添加的 JSP 页面 1'>
<page name=" " file='动态添加的 JSP 页面 2'>
...
</template></body></html>
```

3 Template Taglib 的使用

3.1 Template Taglib 在 Web 应用中的部署

在/WEB-INF/web.xml 文件中添加如下标记:

```
<taglib>
<taglib-uri> mytags</taglib-uri>
<taglib-location>/WEB-INF/taglib.tld</taglib-location>
</taglib>
```

在 JSP 页面中应用该 Taglib 时,在每个页面顶部添加如下代码:

```
<%@ taglib uri = "mytags" prefix = "mt" %>
```

“mt”是该 Taglib 的前缀,在使用每个标记时必须以 mt 为相应前缀^[3]。

3.2 模板文件的制作

模板文件是一个由 table 构成的框架,实现了基于 Java 的网站框架,由 table 来定位每一模块的位置和大小,在需要静态添加文件的地方用 include-Tag 标记标明,并给 includeTag 的 fileName 属性赋值;在需要添加动态页面的地方用 contentTag 标记标明,并给其属性 name 赋值,在该模板中每个 contentTag 的 name 是唯一的。

以下是一个简单的模板的例子和其效果图:

templ.jsp:

```
<%@ taglib uri = "mytags" prefix = "mt"%>
<table width = "80%" border = "1" align = "center">
  <tr>
    <td colspan = "3"><! 一标题栏 -->
    <mt:include fileName = "top.jsp"/>
  </td>
</tr>
  <tr>
    <td colspan = "3"><! 一导航条 -->
    <mt:include fileName = "bar.jsp"/>
  </td>
</tr>
  <tr>
    <td width = "20%" valign = "top">
      <table width = "100%" border = "1">
        <tr><td><! 一论坛登陆 -->
        <mt:include fileName = "login.jsp"/>
        </td></tr>
        <tr><td><! 一友情连接 -->
        <mt:include fileName = "link.jsp"/>
        </td></tr>
      </table>
    </td>
    <td valign = "top">
      <table width = "100%" align = "center" border = "1">
        <tr>
          <td><! 一动态连接 1 -->
          <mt:content name = "P1"/>
        </td>
        <td><! 一动态连接 2 -->
          <mt:content name = "P2"/>
        </td>
      </table>
    </td>
  </tr>
</table>
```

```
</tr>
<tr><td><! 一动态连接 2 -->
<mt:content name = "P2"/></td>
</tr>
</table>
</td>
</tr>
<tr>
  <td colspan = "3"><! -- copyright -->
  <mt:include fileName = "bottom.jsp"/>
</td>
</tr>
</table>
```

模板页面 templ.jsp 的效果图如图 2

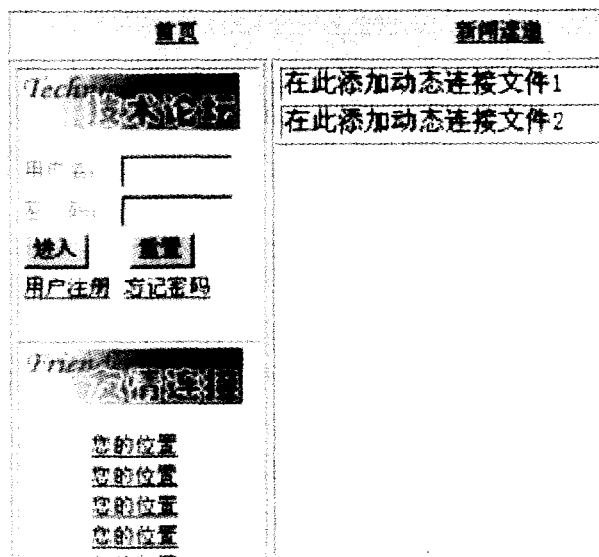


图 2 模板页面效果图

3.3 模板文件的应用

模板文件做好后,就是对它的应用了,只要使用 template 标记确定所要使用的有效的模板文件名,再根据模板文件中的 content 标记,在 page 标记中添加相应的动态连接文件名。

以下是使用 templ 并添加了新闻和购书信息的页面。

usetempl.jsp:

```
<%@ page contentType = "text/html; charset = ISO8859_1" %>
<%@ taglib uri = "/WEB-INF/taglib.tld" prefix = "mt" %>
<html>
  <head>
```

```

<title>useTempl
</title>
</head>
<body>
<mt:template temp = "templ.jsp">
<mt:page name = "P1" file = "news.jsp"/>
<mt:page name = "P2" file = "book.jsp"/>
</mt:template>
</body>
</html>

```

应用页面 usetempl.jsp 的效果图如图 3。

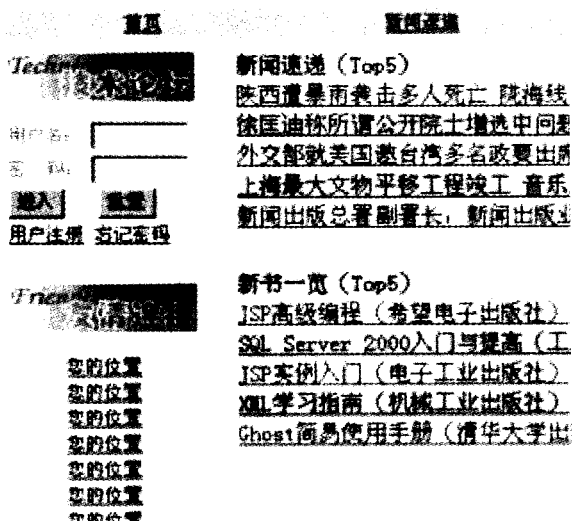


图 3 应用页面效果图

以上页面在模板 templ 中动态添加了 news.jsp 和 book.jsp 两个页面,这两个页面分别从数据库中

读取数据并显示出来.也可以根据业务逻辑需要添加其他页面.如果需要更改页面样式,只要改变模板文件就可以很容易地改变所有使用该模板的页面的样式,其风格由 templ 变成 temp2,只需要改写一个文件,而如果采用以前的独立页面方式,则需要分别改写每个文件,非常繁琐,从中可以很容易看出使用模板的优点。

4 结束语

随着互联网和电子商务的不断发展,大部分的网站,尤其是商业网站的规模越来越大,易于开发和维护的要求越来越高.基于 MVC 模式,应用标签库 TAGLIB 技术,采用 Java 语言实现的网站框架模板方法,具有良好的开发效率和可维护性,在实际工作中可以广泛采用.本文所阐述的方法已经投入到某电子商务网站的开发和使用中,取得了良好的效果。

参 考 文 献

- 1 Marty Hall., Servlet 与 JSP 权威指南. 北京:机械工业出版社, 2002:98
- 2 Phil Hanna 著,闻道工作室,译. JSP 技术大全. 北京:机械工业出版社,2002:56
- 3 Irani F, G 著;司光亚,牛红,等译. JSP WEB 编程指南. 北京:电子工业出版社,2002:125

Research and Application of Websites Frame Template Technology Based on Java

SONG Dacheng, XIA Qingguo¹

(Software College NWPU, Xi'an 710065; Computer College NWPU¹, Xi'an 710072)

[Abstract] The websites developed by frame template technology are easy to improve developing efficiency and maintainability. The technology based on MVC pattern, TAGLIB and Java to develop websites was discussed. Some key techniques on system design and implementation were resolved. These were useful for the development of the similar systems.

[Key words] Java MVC pattern TAGLIB technology