基于MVC设计模式的Struts框架及其应用的研究

冯相忠
(浙江海洋学院 信息学院,浙江 舟山 316004)

摘 要:Struts是基于MVC设计模式的非常优秀的Web应用框架,在Web应用开发中很好地将显示和逻辑分离,提高了代码的可重用性和灵活性。文中首先分析了Struts框架结构及其对Model,View和Controller层的实现原理,并总结了Struts框架的优点,然后给出了采用Struts框架的一个应用实例的实现过程,在应用实例业务逻辑的实现过程中采用了Jbas技术,实现Java代码与数据库操作代码(SQL语句)的分离。

关键词:Struts框架;MVC设计模式;Jbas
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2006)08-0131-03

Research of Struts Framework and Its Application Based on MVC Design Pattern

FENG Xiang-zhong
(College of Information Technology, Zhejiang Ocean University, Zhoushan 316004, China)

Abstract: Struts is an excellent framework based on MVC design pattern for Web applications. It separates view and business logic and increases reusability and flexibility of the code for Web development. This paper briefly analyzes the principle of Struts framework technology and the implementing technology for model, view and controller of Struts based on MVC design pattern. And summarizes the advantages of Struts framework. An implementation of example is given based on Struts framework. In implementation of business logic, separates Java and SQL by Jbas technology.

Key words: Struts framework; MVC design pattern; Jbas

0 引 言

Struts是Apache组织的一个开放源码项目,是目前最流行的MVC体系结构之一,它提供了对MVC系统的底层支持。在Web系统设计中,采用基于MVC模式的Struts框架结构,可以使所设计的Web系统能很好地将显示和逻辑分离,并易于扩展、易于维护、可重用性强[1,2]。因此近年来被越来越多地运用于很多大型系统的开发中,成为Web应用开发中最为流行的框架之一。

1 Struts框架的体系结构

1.1 MVC设计模式

MVC(Model-View-Controller)设计模式由Smalltalk设计,是一个经典的软件体系结构,它采用了"分治"的思想,将表示和数据相互分离,将系统分成3个组件,即模型(Model)、视图(View)和控制器(Controller)[3,4],如图1所示。其中模型表示应用程序的业务逻辑,是应用程序的核心;视图是模型的外在表现形式,也是用户所见到的JSP页面,一个模型可以对应一个或多个视图;控制器是模型与视图的联系纽带,控制器接受用户的请求,把用户数据传给业务逻辑模块,并调用相应的业务逻辑模块进行处理,最后根据用户所需要的响应调用相应的视图模块生成结果页面返回浏览器。

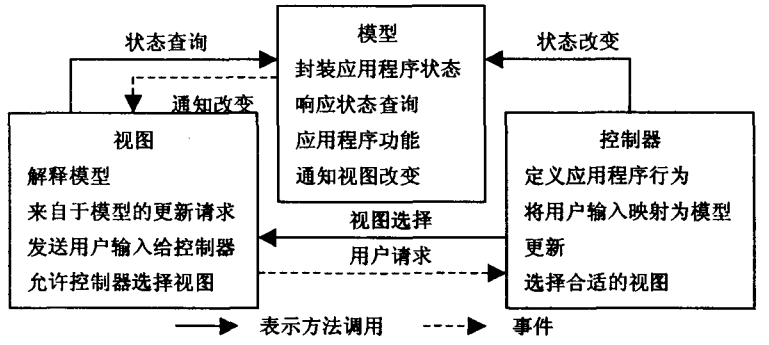

图1 MVC模式结构

MVC设计模式的优点是:

(1)各组件之间实现了松散的耦合;

(2)控制器和视图可以随着模块的扩展而扩展;

(3)将业务规则封装到组件可以提高模块的可重用性;

(4)可以实现并行开发。

收稿日期:2005-11-28

作者简介:冯相忠(1962-),男,内蒙古赤峰人,副教授,主要研究方向为计算机应用技术。

1.2 Struts框架

Struts框架是在MVC模式基础上开发的新一代Web框架,在它的支持下可以快速开发基于Web的应用,而且提高了MVC的分层应用。它的标记库具有强大的页面开发功能[5]。Struts框架的处理流程清楚地体现了MVC系统的特点,简单的Struts组件结构如图2所示。客户端通过Browser发出请求后,请求被Struts的控制器ActionServlet获得,ActionServlet在Struts-config.xml配置文件中查找有效映射,然后将相应的ActionMapping对象转发给Action处理器对象进行处理。Action处理器对象访问ActionForm中的数据,处理和响应客户的请求,它还调用后台的Bean组件,这些组件封装了具体的业务逻辑。Action处理器对象根据处理结果通知控制器,控制器进行下一步的处理。

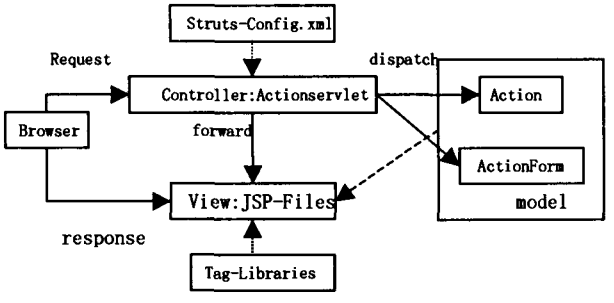

图2 Struts框架的组件结构图

作为一个MVC的框架,Struts对Model,View和Controller都提供了对应的实现组件。

(1) Controller:控制器的作用是从客户端接收请求,并且选择执行相应的业务逻辑,然后把响应结果送回到客户端。在Struts中的Controller功能由ActionServlet和ActionMapping对象构成,核心是一个Servlet类型的对象ActionServlet,它用来接收客户端的请求。Servlet包括一组基于配置的ActionMapping对象,每个ActionMapping对象实现了一个请求到一个具体的Model部分中Action处理器对象之间的映射。

(2) Model: Struts为Model部分提供了Action和ActionForm对象,所有的Action处理器对象都是开发者从Struts的Action类派生的子类。Action处理器对象封装了具体的处理逻辑,调用业务逻辑模块,并且把响应提交到合适的View组件以产生响应。Struts提供的ActionForm组件对象可以通过定义属性描述客户端表单数据,开发者可以从它派生子类对象,利用它和Struts提供的自定义标记库结合,可以实现对客户端的表单数据的良好封装和支持,Action处理器对象可以直接对它进行读写,而不再需要与Request,Response对象进行数据交互。通过ActionForm组件对象实现了对View和Model之间交互的支持。Struts也可根据系统的复杂度使用实体Bean和会话Bean等组件来实现系统状态。

(3) View: Struts中的View部分是通过JSP技术实现的。Struts提供了自定义的标记库,通过这些自定义标记可以非常好地与系统的Model部分交互,创建的JSP表单实现与Model部分中的ActionForm的映射,完成对用户数据的封装,同时自定义标记还提供了像模板定制等多种显示功能。

Struts结构除具有MVC设计模式的特点外,还具有如下的优点。

a. struts将MVC结构的JSP,Servlet代码重新分成了ActionForm,ActionServlet和Action三个部分,使显示、控制和事务逻辑的分工更加清晰,从而对复杂系统的开发起到了简化的作用。

b. Struts的Action类简化了业务逻辑,而复杂的业务逻辑由JavaBean,EJB等来处理。这样使得Action类并不直接访问数据库,而由JavaBean,EJB等来处理。

c. Struts强大的标记库功能,使JSP页面可不内嵌很多的Java代码,使页面的开发得到简化。

d. Struts结构使视图的开发和数据的处理可以完全分开,使开发过程分工更加明确。

2 采用Struts结构的应用实例

Struts结构非常适合于大型的复杂的系统的开发,如笔者在开发手机短信采集系统的过程中,采用了Struts框架结构。以下是其中的短信服务子系统储存箱模块的具体实现过程。

(1) 视图层的实现。

储存箱主要用于存储留言、通告等要发送的信息。

存储箱页面由smsSaveboxInfo.jsp,smsSaveQueryList.jsp和smsSaveboxSendInfo.jsp实现,分别用于信息录入、显示和修改,信息列表分页显示信息发送的录入。在页面的实现中使用了Struts框架的标签库的内容。

(2) 控制层的实现。

控制层的主要任务是接受/截获用户请求,并把接受到的请求映射到相应的Action类,调用模型组件来执行相应的业务逻辑,获取业务逻辑执行结果,根据当前状态以及业务逻辑执行结果,选择合适的视图组件返回给用户。

储存箱的控制层实现由smsSaveboxForm.java,smsSaveboxAction.java程序来实现。

smsSaveboxForm.java继承了ActionForm类,定义属性描述客户端表单数据,把ActionForm类和Struts提供的自定义标记库结合,实现了对客户端的表单数据的良好封装和支持。

smsSaveboxAction.java继承了Action类,实现了储存箱功能的映射转发。对页面中不同的请求映射做了相应的转发。主要实现的操作有:

List:列表显示储存箱信息;View:查看储存箱某条信息的详细内容;Query:查询信息;Delete:删除某条信息记录;ShowNew:新增操作的初始化;PreSend:发送操作的初始化;Send:发送操作; Update:更新存储信息;Insert:增加

存储信息。

(3) 业务逻辑层的实现。

Struts 框架没有提供现成的业务逻辑,但 Struts 允许使用其它模型框架来处理应用的业务逻辑,如 EJB,JavaBean 等。在本系统中采用 Jbas 框架处理业务逻辑,在 Jbas 框架中,把操作数据库的 SQL 语句放到自定义的 SMSCConfig. xml 配置文件中,Java 程序只负责业务逻辑的实现,不包含数据库操作语句,实现了 Java 代码与数据库操作相分离。

处理程序为 smsSaveboxDAO. java,它的部分代码和功能为:

```
public class smsSaveboxDAO {
//载入 Sql 配置文件
XSqlAction xSqlAction = new XSqlAction("SMSConfig. xml");
//通过手机号查找,返回 list
public ArrayList getSmsSaveboxListByMobile(String mobile){
  xSqlAction. initSql(this. getClass(). getName(),"getSmsSaveboxListByMobile");
  xSqlAction. setStringNo(1,mobile);
  ArrayList retList = xSqlAction. executeObjectList();
  return retList;
  }
  …… }
```

其中 SMSCConfig. xml 部分代码及含义如下:

```
<? xml version = "1.0" encoding = "gbk"? >
<jbas - config>
<class name = "com. feng. sms. SmsSaveboxDAO" > \ \ 表示以下 sql 语句是属于的类
  <method name = "getSmsSaveboxList" > \ \ 表示以下 sql 语句是属于的方法
  <item> \ \ 指定了一个完整的 SQL 操作
    <sql isdebug = "true">
      <! [CDATA[
      select * from sms_savebox \ \ 要执行的 sql 语句
      ]]>
    </sql>
    <innerClass name = "com. feng. sms. SmsSaveboxBean" > \ \ 表示返回从数据库中查询出数据存放的类
    \ \ 以下所有 field 中为配置 java 类中与数据库中对应的字段
    <field name = "joint" value = "joint"/>
    <field name = "message" value = "message"/>
    ……
    <field name = "lasteditdate" value = "lasteditdate"/>
    </innerClass>
    </item>
  </method>
  ……
 </class>
 ……
</jbas - config>
```

采用以上 Jbas 框架,可以使相应的 Java 程序中相关的操作实现变得简单,从而使程序的出错几率变少。把 SQL 语句从 Java 程序里移到 XML 文件中,可以实现不修改 Java 程序,只修改 XML 配置文件,就可以达到支持多种或不同种类数据库的目的。

(4) Struts - config. xml 内容的配置。

配置文件 Struts - config. xml:通过 < form - bean > 来配置 Form,通过 < global - forwards > 配置页面的导航,通过 < action - mapping > 来配置映射,并指定它的表单输入页面。配置文件部分代码及功能如下:

```
<struts - config>
<! - - = = 定义 Form Bean = = - >
<! - FormBean 是 struts 的一个概念,本质是 JavaBean 用来自动存储页面表单中各个域的值,并在适当的时候回填表单域 - >
<form - beans>
<form - bean name = "smsSaveboxForm" type = "com. feng. sms. SmsSaveboxForm"/>
……
</form - beans>
<! - - = = 定义 Global Forward = = - >
<global - forwards>
<forward name = "login" path = "/webapp/menu/index. jsp"/>
……
</global - forwards>
<! - - = = 定义 Action Mapping = = - >
```

web. xml 中后缀为. do 的请求被转到这里处理,这里相当于 Struts 的 Model 部分,Model 部分是 Struts 中比较灵活的地方。

```
<action - mappings>
<action path = "/webapp/sms/smsSavebox" type = "com. feng. sms. SmsSaveboxAction"
name = "smsSaveboxForm" scope = "request" validate = "false">
<forward name = "smsSaveboxInfo" path = "/webapp/sms/smsSaveboxInfo. jsp"/>
<forward name = "smsSaveboxSendInfo" path = "/webapp/sms/smsSaveboxSendInfo. jsp"/>
<forward name = "smsSaveboxQueryList" path = "/webapp/sms/smsSaveboxQueryList. jsp"/>
</action>
……
</action - mappings>
</struts - config>
```

系统的整个逻辑流程都包含在 Struts - config. xml 这个分层的 XML 文件中,使开发者易于理解和查看,也不必费力地阅读 Java 代码来理解应用程序的流程,同样也不必在更改流程后重新编译代码。

(下转第 136 页)

```
public AddMessageServlet()
  //连接数据库
  JDBCBean splBean = new JDBCBean();
  splConn = SplBean.getConnection();
public void doGet ( HttpServletRequest request, HttpServletResponse response)
  向数据库插入数据;
  //调用显示类
  request.getRequestDispatcher("/viewMessages_servlet");
  requestDispatcher.forward(request,response);
public void doPost ( HttpServletRequest request, HttpServletResponse response)
  执行 doGet();
```

/* 当客户端浏览器打开了继承 HttpServlet 类的 Servlet 类,便会发出一个 GET 请求,自动调用 doGet()方法。当客户端发出带有数据的请求后,便会发出一个 POST 请求,自动调用 doPost()方法。 */

● **ViewMessageServlet:**

```
public class ViewMessageServlet extends HttpServlet
public ViewMessageServlet()
  //连接数据库
  JDBCBean splBean = new JDBCBean();
  splConn = SplBean.getConnection();
public void doGet ( HttpServletRequest request, HttpServletResponse response)
  从数据库中取得所有数据;
  调用 mysite\viewMessages.jsp;
  //调用全部数据显示
public void doPost ( HttpServletRequest request, HttpServletResponse response)
执行 doGet();
```

● **mysite\viewMessages.jsp**(略)

● **JDBCBean**

```
public class JDBCBean
public void init()
  driver = getInitParameter("DRIVER");
  password = getInitParameter("PASSWORD");
  url = getInitParameter("URL");
  user = getInitParameter("USER");
private Connection getConnection()
  Class.forName(driver);
  con = DriverManager.getConnection(url,user,password);
  return con;
  对数据库的其他各种操作方法(略)。
```

3 结束语

对于基于 JSP 的 Web 应用开发,利用 JSP/Servlet/JavaBean 技术和 XML 描述语言的 MVC 改进模式实现了表示层、逻辑层和数据库访问层等多层面分离,封装了用户数据,增强了各层、各模块的高内聚低耦合的特性,使得软件在扩展性、复用性和维护性方面上有了极大的提高。虽然不一定完全用此模式实现所有应用系统,但可以将其中的一些思想用于 JSP 网站开发中[5]。

近年来,在 JSP 网站建设中基于 MVC 新技术的应用越来越多。在标签语言(HTML 标签库、Bean 标签库、Login 标签库、JSTL 标签库等)、表达式语言、Struts/Tapestry/JSF、Java Server Face 等技术方面的开发框架都是基于 MVC 的,充分实现了 MVC,尤其是框架技术在 MVC 的基准模式下得到极大的发展和应用,这是值得关注的。

参考文献:

[1] 张砚秋,陈 川,何明德. 基于 MVC 设计模式构筑 JSP/Servlet+EJB 的 Web 应用[J]. 计算机工程,2001(11):71-73.

[2] 田萍芳,李跃新. MVC 模式在 Java B/S 开发中的应用研究[J]. 湖北大学学报(自然科学版),2005,27(2):137-139.

[3] Cswcafe. MVC 设计模式带来更好的软件结构和代码重用[EB/OL]. http://dev.csdn.net,2005.

[4] 飞思科技产品研发中心. JSP 应用开发详解(第 2 版)[M]. 北京:电子工业出版社,2005.

[5] 陈景霞,陈 桦,张鹏伟. 利用 XML 扩展基于 MVC 模式的 Web 应用框架的研究[J]. 微电子学与计算机,2005,22(4):38-41.

(上接第 133 页)

3 结束语

基于 MVC 模式的 Struts 框架结构,把页面层和逻辑层分开,使得各个层间的耦合度降低,提高了系统的可重用性、灵活性和可维护性。采用 Struts 框架结构对手机短信采集系统的实现,更可以体现出 Struts 框架的优越性。它使得软件的开发过程清晰,效率大大提高。

参考文献:

[1] 求是科技. Java 数据库系统开发实例导航[M]. 北京:人民邮电出版社, 2004.

[2] Apache Software Foundation. The Struts User's Guide[EB/OL]. http://Jakarta.apache.org,2005.

[3] 陆荣幸,郁 洲,阮永良,等. J2EE 平台上 MVC 设计模式的研究与实现[J]. 计算机应用与研究, 2003,20(3):144-146.

[4] 祁耀武,李福太,陈逢春,等. J2EE 平台上 MVC 设计模式在电子政务系统中的应用[J]. 计算机应用研究, 2004,21(8): 203-205.

[5] 孙凌燕,陈保岚,孙 健. 基于 Struts 的 Web 应用框架设计与研究[J]. 计算机工程,2005,31(8):57-60.