

Struts-MVC 设计框架在软件过程管理系统中的应用

The Application of Struts-MVC Software Framework Architecture in Software Process Management System

罗杰云 贺敏伟 (广东江门五邑大学信息学院 529020)

摘要: 本文主要分析了 Struts 的工作原理, 以及在软件过程管理系统中的应用。最后, 给出了一个 Struts-MVC 开发的实例。

关键词: Struts MVC J2EE 框架

1 引言

随着软件信息产业的高速发展, 软件体系结构的日益复杂, 软件开发过程的规范化程度已经成为一个软件企业能够在国际市场上得到认可的必要评测手段。开发能帮助软件企业进行开发过程的规范化、科学化和可控化管理的实用工具显得尤其重要, 软件过程管理系统正是为适应这种要求而开发的。系统功能模块主要包括: 过程流程控制、问题跟踪、报告生成、任务安排、质量保证、文档存储等。

J2EE模型提供了一个很好的企业应用框架及解决方法, 提供了灵活的技术选择, 但它只是一个应用开发框架, 不能解决系统的所有问题; 同时, 多层企业应用框架在提高了软件的重用性和分解了问题复杂性的同时, 也使得代码庞大, 层与层之间的控制关系复杂。因此, 对于复杂的企业应用软件的开发来说, 一个好的软件架构也是至关重要的, 是决定大型企业应用软件质量的关键因素。首先, 它能使软件开发有章可循, 结构清晰, 缩短开发周期; 其次, 它能提高软件的可维护性与可扩展性。

MVC(Model-View-Controller)是一个经典的软件体系结构, 已经得到广泛运用。在J2EE企业应用开发中, 采用MVC软件体系结构, 能够简化软件开发, 提高软件性能和可维护性, 达到提高软件质量的目的。而Struts是使用servlet和JavaServer Pages技术的一种MVC实现, 它在实现WEB应用系统中具有得天独厚的优势。本文主要讨论struts-MVC原理及在软件过程管理系统中的应用。

2 Struts 工作原理

Struts 是MVC的一种实现, 它继承了MVC的各项特性, 并根据J2EE的特点, 做了相应的变化与扩展。

2.1 Struts 的概览

如下图1所示:

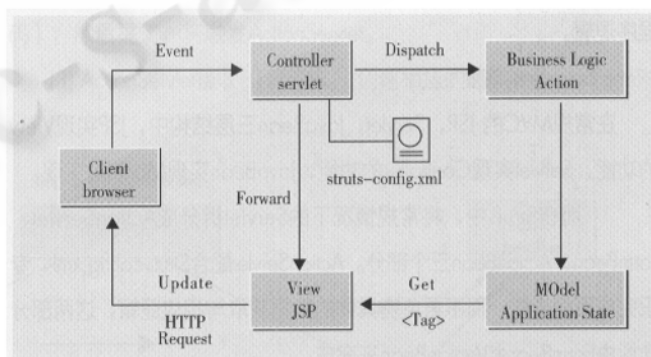


图1 Struts 概览

• Controller (控制器)

控制器接收来自浏览器的请求, 并决定将这个请求发往何处。决定执行什么业务逻辑功能, 然后将产生下一步用户界面的责任委派给一个适当的View组件。在Struts中, 控制器是以servlet实现的一个命令设计模式。

• Business Logic (业务逻辑)

业务逻辑更新模型的状态，并帮助控制应用程序的流程。在Struts中，这是通过作为实际业务逻辑“瘦”包装的 Action 类完成的。

• Model (模型) 的状态

模型表示应用程序的状态。业务逻辑对象更新应用程序的状态。

• View (视图)

视图是用来呈现Model的内容，决定如何把数据表示出来，Model的改变将触发View的改变，以维持数据的一致性。View通常是一个JSP文件。主要由JSP生成页面完成视图，其中没有流程逻辑，没

理解应用程序，尤其是对于大型应用程序而言。

(2) 网页设计人员不必费力地阅读Java代码来理解应用程序的流程。

(3) Java开发人员也不必在更改流程以后重新编译代码。

2.2 Struts 的主要组件

在struts中，主要由三个类 (ActionServlet、Action、ActionForm) 和一个配置文件struts-config.xml组成，三个类的关系如下面的图2UML图所示：

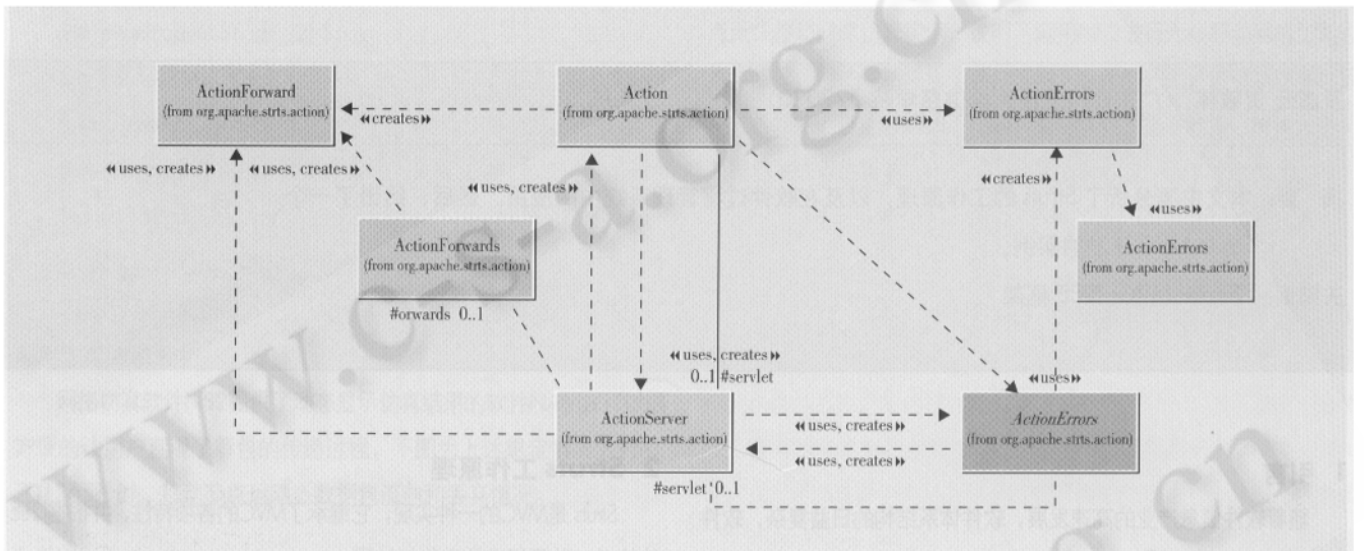


图 2 UML 图

有业务逻辑，也没有模型信息，只有标记。这有利于分开表现逻辑和程序逻辑。

• Servlet的演变

在常规MVC的JSP, Servlet, JavaBean三层结构中，JSP实现View的功能，Servlet实现Controller的功能，JavaBean实现Model的实现。

而在Struts中，将常规情况下的Servlet拆分成ActionServlet、FormBean、ActionBean三个部分。ActionServlet配合Struts-config.xml，专职完成页面导航，而不再负责具体的数据获取与相应逻辑，这两部分功能由FormBean和ActionBean来完成。

• struts-config.xml 配置文件:

Struts-config.xml是ActionServlet的核心，Struts-config.xml集中了所有页面的导航定义。对于大型的WEB项目，通过此配置文件即可迅速把握其脉络，这不管是对于前期的开发，还是后期的维护或升级都是大有裨益的。文件struts-config.xml是一个分层的文本文件，具有以下几个优点：

(1) 它存储了应用程序的整个逻辑流程，这使得人们更容易查看和

• ActionServlet 类

ActionServlet 承担MVC中Controller角色，是该 MVC 实现的 Command 部分，它是这一框架的核心。ActionServlet是一个通用的控制组件。这个控制组件提供了处理所有发送到Struts的HTTP请求的入口点。它截取和分发这些请求到相应的动作类（这些动作类都是Action类的子类）。它主要完成以下功能：

- * 接收客户端请求
- * 根据客户端的URL将请求映射到一个相应的Action类
- * 从请求中获取数据填充Form Bean (如果需要)
- * 调用Action类的execute()方法获取数据或者执行业务逻辑
- * 选择正确的视图响应客户

此外，ActionServlet还负责初始化和清除应用配置信息的任务。

• ActionForm 类

ActionForm 维护Web应用程序的会话状态。Action Form通常也称为FormBean，封装了来自于Client的用户请求信息，如表单信息。此外，ActionForm具有过滤保护作用，只有通过ActionForm验证的数据才

能够发送到Action处理。ActionForm首先利用属性的getter和setter方法来实现初始化。初始化完毕后，ActionForm的validate方法被调用，你可以在其中来检查请求参数的正确性和有效性，并且可以将错误信息以ActionErrors的形式返回到输入窗体。否则，ActionForm将被作为参数传给action的execute方法以供使用。

• Action 类

Action 类是业务逻辑的一个包装。Action 类的用途是将HttpServletRequest转换为业务逻辑。Action通常称之为ActionBean，获取从ActionServlet传来的FormBean，取出FormBean中的相关信息，并做出相关的处理，一般是调用Java Bean或EJB等。

具体Action类的功能一般都在execute方法中完成，其中主要涉及到以下几个方面：

- * 辅助ActionForm进行一些表单数据的检查。
- * 执行必要的业务逻辑，比如存取数据库，调用实体bean等。
- * 更新服务器端的bean数据，因为后续对象中可能会用到这些数据，比如在JSP中利用bean.write来获得这些数据。
- * 根据处理结果决定程序的去处，并以ActionForward对象的形式返回给ActionServlet。

3 实例

用户登录是大多数的网站需要实现的基本功能之一。下面，本文以此WEB应用程序的开发为例，介绍基于Struts架构的方法进行WEB应用程序开发的具体步骤和技术细节。

通常，实现用户登录的WEB应用程序应完成以下功能：

- (1) 提供用户登录界面，这部分程序主要由网页设计人员完成。
- (2) 根据输入的用户名和密码判断用户是否存在以及密码的正确性。

[3] 如果通过合法性检验，则提取该用户的个人资料并设置其状态为已登录，并返回一登录界面；

[4] 如果未通过合法性检验，则设置其状态为未登录，并返回一出错界面；

为了实现以上功能，设计Struts-MVC的过程如图3所示。

3.1 视图的设计

在基于struts-mvc架构开发的用户登录程序中，登录入口界面以及登录结果界面为视图模块；分别为三个.jsp文件：注册首页RegisterPerson.jsp文件，注册成功界面RegisterPersonSuccess.jsp文件，注册失败界面RegisterPersonFailure.jsp文件。

3.2 模型的设计

该部分主要涉及对后台数据库的访问操作和对用户的合法性进行判断。由封装在JavaBean文件ValidateBean.java中的程序完成。

3.3 控制器的设计

这主要包括ActionForm对象和Action对象的构造。

ActionForm对象的构造：由文件RegisterPersonForm.java完成。主要完成数据的收集与存放。

Action对象的构造：Action只负责完成与控制有关的功能。由文件RegisterPersonAction.java完成。

为了实现用户登录，该程序需要进行以下的主要步骤：

(1) 首先，控制器（ActionServlet）进行初始化工作。读取配置文件struts-config.xml，将配置文件中的Action映射定义保存在ActionConfig集合中。

(2) 用户在登录页面RegisterPerson.jsp中输入数据（如用户名和密码），然后提交，将触发ActionServlet。

[3] ActionServlet接收HTTP请求。创建相应的RegisterPersonForm对

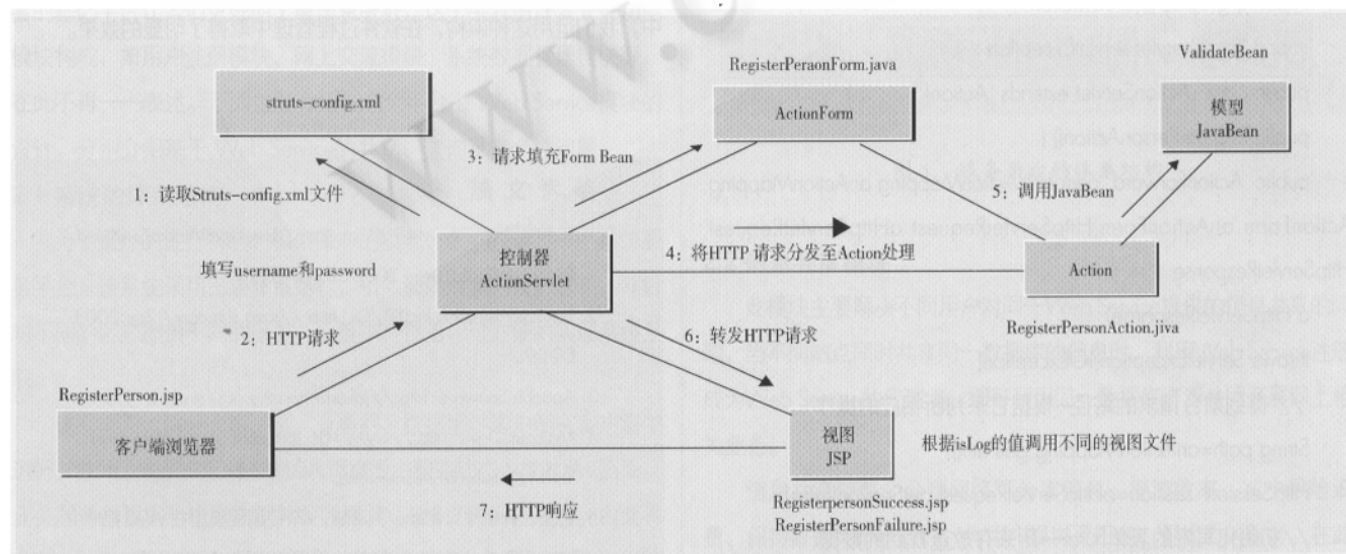


图 3

象实例,用HTTP请求中的参数(username和password)来填充。此外,在RegisterPersonForm对象中还可以调用validate方法来检查请求参数的一般合法性,并且可以返回一个包含所有错误信息的ActionErrors对象。如果合法性检验通过,RegisterPersonForm自动将这些参数信息以JavaBean的方式保存在Servlet Context中,这样它们就可以被其他Action对象或者JSP调用。

(4) ActionServlet从ActionConfig中找出对应于该请求的Action子类RegisterPersonAction并将请求分发到该类进行处理。

(5) 执行RegisterPersonAction中的execute方法,在该方法中调用ValidateBean的execute()方法来进行用户验证。该方法首先利用参数username和password通过JDBC进行读取数据库的操作,判定数据库中是否存在该用户、密码是否正确等。接下来创建RegisterResultBean的实例,当验证通过时,设置表示用户登录状态的参数isLog为true,并从数据库中读取该用户的基本信息并存在ResultBean中;反之,设置isLog为false。

ResultBean存放在当前会话(session)中。

(6) ActionServlet读取存放在当前会话中的RegisterResultBean中的参数isLog的值并进行判断。

(7) 如果isLog为false,ActionServlet调用出错显示页面RegisterPersonFailure.jsp文件,返回登录不成功的页面;如果isLog为true,ActionServlet调用显示页面RegisterPersonSuccess.jsp文件,返回登录成功的基本信息。

下面是流程控制模块的ActionServlet的部分代码:

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.IOException;
import org.apache.struts.action.*;
import javax.servlet.ServletException;

public class ActionServlet extends Action {
    public RegisterPersonAction() {
        public ActionForward perform(ActionMapping anActionMapping,
        ActionForm anActionForm,HttpServletRequest aHttpServletRequest,
        HttpServletResponse
        o HttpServletResponse)
        throws ServletException,IOException{
        // 得到前台请求的路径—根据它来判断相应的操作
        String path=anActionMapping.getPath();
        HttpSession session=aHttpServletRequest.getSession(false);
        //初始化写好的表单Form—用来存放查找到的数据
        RegisterPersonForm form=(RegisterPersonForm)anActionForm;
```

//根据path的值,选择处理方式

```
try {
    if (path.endsWith("RegisterPerson")){//进入注册首页
        //如果按下了确定按钮
        if (aHttpServletRequest.getParameter("confirm")!=null){
            //调用相应的程序得到所要的数据
            ValidateBean RegisterValidateBean new validateBean();
            RegisterValidateBean.setUsername(username);
            RegisterValidateBean.setPassword(password);
            //进行用户验证,结果存放在session中,名为: userinfo
            RegisterValidateBean.execute();
            //取出验证结果作为判断依据
            HttpSession userinfo=(RegisterResultBean)session.getValue("userinfo");
            if (RegisterResultBean.getIsLog())
                Forward =anActionMapping.findForward("RegisterPersonSuccess.
                jsp");
            Else
                Forward =anActionMapping.findForward ("RegisterPersonFailure.
                jsp");
            };
        }
```

4 总结

经过以上分析和实例验证,可以看到Struts是一种优秀的J2EE MVC架构方式。它使得应用的整个逻辑流是在一个分等级的文本文件中,利用ActionServlet配合Struts.config.xml实现了对整个系统的导航。这样就可以比较容易地查看、理解应用,特别是大的应用。它也增强了开发人员对系统的整体把握,提高了系统的可维护性和可扩充性。在项目开发中,我们运用这种架构,在软件过程管理中取得了明显的效果。

参考文献

- 1 王和全, <http://www-900.ibm.com/developerWorks/cn/java/istruts1-1/index.shtml>, 2003-08。
- 2 Sun Microsystems, Inc[EB/OL]. <http://java.sun.com/j2ee/2003-09-09>。
- 3 Apache Software. <http://jakarta.apache.org/struts/>。
- 4 Malcolm Davis. <http://www-106.ibm.com/developerworks/library/jstruts/?dwzone=java>。
- 5 孙莹、许俊华、张毅、贺清峰, MVC编程模型在Web程序中的应用及Java实现[J], 计算机工程与应用, 2001.17。