

基于 MVC 模式的 Struts 框架的研究与应用

张南平¹, 朱富利²

(1. 武汉菲旺软件技术有限公司, 湖北 武汉 430070;

2. 武汉理工大学 计算机科学与技术学院, 湖北 武汉 430070)

摘要: 无论是选用纯 JSP 还是纯 Servlet 设计 Web 应用, 都有扩展性差、不易分工的局限性。Struts 就是把它们联系在一起的一种有力工具。它是一个 MVC 框架, 提供了对开发 MVC 系统的底层支持, 采用的主要技术有 Servlet, JSP 和自定义标签库。文中在介绍 MVC 模式的基础上, 介绍了 Struts 体系结构, 最后结合一个实例阐述了如何用 Struts 实现 MVC 结构, 从而设计出了一个优秀的、完整的 Struts 开发框架。Struts 框架被认为是目前开发 J2EE Web 应用最为理想和成熟的框架, 这种开发模式将会逐渐被广大程序员所接受, 并在实际开发中得到广泛应用。

关键词: Struts 框架; MVC 模型; Servlet; JSP; 标签

中图分类号: TP311.1

文献标识码: A

文章编号: 1005-3751(2006)03-0229-03

Research and Application of Struts Framework Based on MVC Pattern

ZHANG Nan-ping¹, ZHU Fu-li²

(1. Wuhan Philwong Software Technology Co., Ltd, Wuhan 430070, China;

2. School of Computer Sci. and Techn., Wuhan Univ. of Tech., Wuhan 430070, China)

Abstract: There will be localization of less expansibility and hard to divide the work whether using whole JSP or whole Servlet to design Web application. While the Struts is a powerful utilizable tool that relate them. It's an MVC frame and offer a bottom aegis to exploit MVS System. Servlet, JSP and custom base of tag are the main technologies of Struts. In this paper, on the base of introduction of the MVC pattern, introduced the framework of Struts. At last, combined an example, explained how to achieve MVC framework with Struts, then an excellent and integrate Struts framework is designed. At present, Struts framework is considered to be the idealest and maturest framework of J2EE Web application, this framework will be accepted gradually by most of programmers, and will be used widely in practical applications.

Key words: Struts framework; MVC model; Servlet; JSP; tag

0 引言

如今, SUN 公司提出的 J2EE (Java 2 Enterprise Edition) 正逐渐成为企业级网络应用软件开发的标准, 它提供了一种基于组件的设计、开发、集成、部署的企业应用系统的方法。为了提高应用系统中组件的重用能力和事务控制的灵活性, 越来越多的开发人员采用 MVC (Model - View - Controller) 设计模式来分离系统的用户界面、业务逻辑及业务控制。Struts 体系结构就是 MVC 设计模式的一种具体实现, 它为 Web 应用提供了一个通用的框架, 使得开发人员可以把精力集中在如何解决实际业务问题上。此外, Struts 框架提供了许多可供扩展和定制的地方, 使得应用程序可以方便地扩展框架, 更好地适应用户的实际需求。

1 MVC 设计模式概述

MVC 是 Model - View - Controller 的简称, 它是一种设计模式, 最近几年被推荐为 SUN 公司 J2EE 平台的设计模式。它强制性地把应用程序的输入、处理和输出分开^[1]。MVC 把应用程序分成 3 个核心模块: 模型、视图和控制器, 它们分别担负不同的任务。图 1 显示了这几个模块各自的功能以及它们的相互关系。

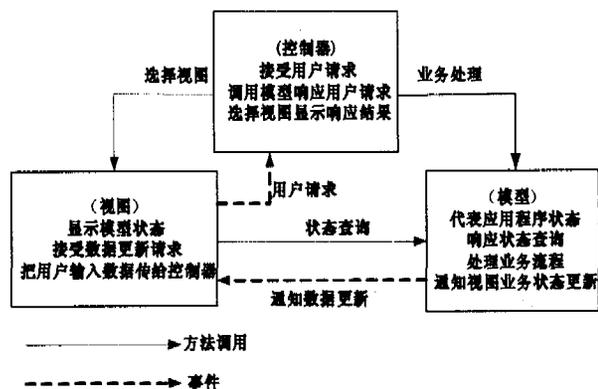


图 1 MVC 设计模式

收稿日期: 2005-06-08

作者简介: 张南平 (1961—), 男, 湖南人, 教授, 硕士生导师, 研究方向为计算机网络, ERP, 电子商务。

1.1 Model

它是业务逻辑处理模块。是对企业业务流程中的对象进行抽象,封装了对象的属性和对象隐含的业务逻辑。

1.2 View

它是用户视图模块。它为用户提供了输入手段,并触发 Controller 运行,然后通过 Model 访问企业数据,并用某种方式来显示这些数据。

1.3 Controller

它是流程控制模块。它管理着用户与 View 的交互,能从 View 中取得数据并传给 Model 去执行业务逻辑,然后它根据 Model 的执行结果,选择一个合适的 View 把结果展现给用户。

1.4 MVC 处理过程

总结 MVC 处理过程为:首先控制器接收用户的请求,并决定应该调用哪个模型来进行处理;然后模型根据用户请求进行相应的业务逻辑处理,并返回数据;最后控制器调用相应的视图来格式化模型返回的数据,并通过视图呈现给用户。

2 Struts 实现 MVC 的机制

Struts 实质上就是在 JSP Model2 的基础上实现的一个 MVC 框架。在 Struts 框架中,模型由实现业务逻辑的 JavaBean 或 EJB 组件构成,控制器由 ActionServlet 和 Action 来实现,视图由一组 JSP 文件构成^[2]。图 2 显示了 Struts 实现的 MVC 框架。

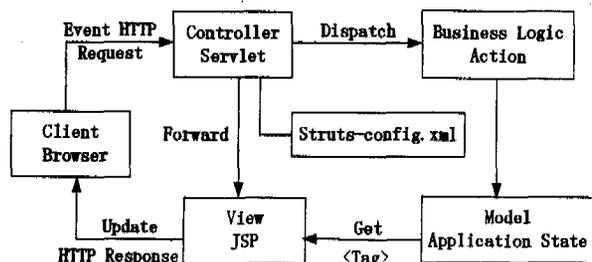


图 2 Struts 实现的 MVC 框架

2.1 Model

Struts 中模型部分分为两类:系统的内部状态和改变系统状态的动作(业务逻辑)。Struts 为模型部分提供了 Action 和 ActionForm 对象。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视图组件以产生响应。ActionForm 组件对象可以通过定义属性描述客户端表单数据。利用 ActionForm 组件对象和 Struts 提供的自定义标记库可以实现对客户端的表单数据的良好封装和支持。通过 ActionForm 组件对象实现了视图和模型之间交互^[3]。

2.2 View

视图主要由 JSP 建立,它是由与控制器 Servlet 配合工作的一整套 JSP 定制标签库构成,通过这些自定义标记可以很好地和系统的模型(Model)部分交互,通过使用这些自定义标记创建的 JSP 表单,可以实现和模型(Model)

部分中的 ActionForm 的映射,完成对用户数据的封装,同时这些自定义标记还提供了像模板定制等多种显示功能。另外,Struts 实现了页面表单和 ActionFormBean 之间的交互,由 Struts 框架自动处理,可以节省大量重复的处理,简化了整个程序。

2.3 Controller

Struts 中的 Controller 主要是它自身提供的 ActionServlet,而 ActionServlet 的核心就是 struts-config.xml 配置文件,它包含了所有页面导航的定义。ActionServlet 接受请求并根据配置文件中的定义将控制转移到适当的 Action 类,其余的控制逻辑以及对 Model 的访问由 Action 类负责完成。

2.4 Struts 的配置文件 struts-config.xml

一个用户请求是通过 ActionServlet 来处理 and 转发的。那么,ActionServlet 如何决定把用户请求转发给哪个 Action 对象呢?这就需要一些描述用户请求路径和 Action 映射关系的配置信息了。在 Struts 中,这些配置映射信息都存在特定的 XML 文件 struts-config.xml 中。在该配置文件中,每一个 Action 的映射信息都通过一个 <action> 元素来配置。这些配置信息在系统启动的时候被读入内存,供 Struts 在运行期间使用。在内存中,每一个 <action> 元素都对应一个 org.apache.struts.action.ActionMapping 类的实例^[4]。

3 Struts 简单实例

下面把 Struts 框架运用到 loginExample 应用中。Struts 框架可以方便迅速地把一个复杂的应用划分为模型、视图和控制器组件,而 Struts 的配置文件 struts-config.xml 则可以灵活地组装这些组件,简化开发过程^[5]。

以下是 loginExample 应用的各个模块的构成:

3.1 模型部分

模型包括一个 JavaBean 组件 LoginForm,它有 username, password 两个属性,代表用户输入的用户名和密码。它提供了 get/set 方法,分别用来读取和设置 username, password 的属性。

LoginForm.java 的源代码如下:

```

import org.apache.struts.action.ActionForm;
public class LoginForm extends ActionForm {
    public String username = "";
    public String password = "";
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
}
  
```

3.2 视图部分

视图包括 3 个 JSP 文件: login.jsp 提供用户界面, 接受用户输入的用户名、密码, 并提交登陆请求到 Action; main.jsp 提供登陆成功界面, 取得 Action 转发的用户名, 并提示该用户登陆成功; error.jsp 提供登陆失败界面, 取得 Action 转发的用户名, 并提示该用户输入的用户名或者是密码错误, 登陆失败。

1) login.jsp 的源代码如下:

```
<%@ page contentType="text/html; charset=GBK"%>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<html:html>
<head><title>登陆页面</title></head>
<body bgcolor="#F2F6FF">
<html:form action="/login.do" method="post">
<table>
<tr> <td align="right">用户名:</td>
<td align="left"><html:text name="loginForm" property="username" size="10"/></td>
</tr>
<tr> <td align="right">密码:</td>
<td align="left"><html:password name="loginForm" property="password" size="10"/></td>
</tr>
<tr><td></td><td align="left"><input type="submit" value="登陆"></td></tr>
</table>
</html:form>
</body>
</html:html>
```

2) main.jsp 的源代码如下:

```
<%@ page contentType="text/html; charset=GBK"%>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<html:html>
<head><title>登陆成功</title></head>
<body bgcolor="#F2F6FF">
<html:form action="/login.do" method="post">
登陆成功, 欢迎<bean:write name="loginForm" property="username"/>!
</html:form>
</body>
</html:html>
```

3) error.jsp 的源代码如下:

```
<%@ page contentType="text/html; charset=GBK"%>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
```

```
<html:html>
<head><title>错误页面</title></head>
<body bgcolor="#F2F6FF">
<html:form action="/login.do" method="post">
sorry,<bean:write name="loginForm" property="username"/>
用户名或密码错误, 登陆失败!
</html:form>
</body>
</html:html>
```

3.3 控制器部分

控制器包括一个 Action 类 LoginAction, 它的任务是从 LoginForm 中接收用户名和密码, 判断如果用户输入的用户名和密码都是“zhufuli”, 则登陆成功, 跳转到 main.jsp; 否则登陆失败, 跳转到 error.jsp。

LoginAction.java 的源代码如下:

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

public class LoginAction extends Action {

    public ActionForward execute (ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) {

        String userName = ((LoginForm) form).getUsername();
        String password = ((LoginForm) form).getPassword();
        if (userName.equals("zhufuli") && password.equals("zhufuli")) {

            return mapping.findForward("success");
        } else {

            return mapping.findForward("failed");
        }
    }
}
```

3.4 配置文件

用户提交请求到 Action 以及 Action 处理完之后跳转到不同的页面都是通过 ActionServlet 来处理 and 转发的。而 ActionServlet 是根据配置文件 struts-config.xml 中的映射信息来做出决定的。

struts-config.xml 的配置信息如下:

```
<? xml version="1.0" encoding="UTF-8"? >
<! DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN" "http://jakarta.apache.org/struts/dtds/struts-config_1-1.dtd">
<struts-config>
    <form-beans><form-bean name="loginForm" type="LoginForm"/></form-beans>
    <action-mappings>
        <action path="/login" name="loginForm" scope="request" input="/login.jsp" type="LoginAction">
            <forward name="success" path="/main.jsp"/>
        </action>
    </action-mappings>
```

(下转第 234 页)

$$NAV_{CTS} = 2T_{SIFS} + T_{CTS} + T_{RES} + 2\tau$$

以避免冲突。

(5) A 发送 RTS 后,要经过 $T_{SIFS} + T_{CTS} + T_{RES} + 2\tau$ 时期,等待 B 的 CTS 的到来。若未收到 CTS,则退避重发。

(6) A 收到 B 的 CTS (D_j, NAV_{CTS}, P_{CTS}) 后,执行以下步骤:

a) 添加 CUL[k] 到 CUL 信道使用列表中:

$$CUL[k].host = B$$

$$CUL[k].ch = D_j$$

$$CUL[k].rel-time = T_{curr} + NAV_{CTS}$$

$$CUL[k].int = 1$$

b) 用最大功率在控制信道上广播 RES (D_j, NAV_{RES}, P_{RES}) 报文,

$$NAV_{RES} = NAV_{CTS} - T_{SIFS} - T_{RES}$$

$$P_{RES} = Power[B]$$

c) 用 Power[B] 在数据信道 D_j 上传送数据报文(步骤 b)与 c)同步进行)。

若 A 收到的是 B 的 CTS(Test) 信号,则 A 回到步骤 1 退避等待,或者重新选择新释放的数据信道。

(7) 除 A 以外的其他主机收到 B 的 CTS (D_j, NAV_{CTS}, P_{CTS}) 时,将更新自己的 CUL,与步骤 6a) 类似。但应注意:

$$CUL[k].rel-time = T_{curr} + NAV_{CTS} + \tau$$

$$CUL[k].int = \begin{cases} 0, & \text{如果 } Power[B] > P_{CTS} \\ 1, & \text{如果 } Power[B] \leq P_{CTS} \end{cases}$$

若收到的是 B 的 CTS(Test) 信号,则忽略该报文。

(8) 若除 A 以外的其他主机收到 RES (D_j, NAV_{RES}, P_{RES}) 报文,则添加 CUL[k] 到 CUL 信道使用列表中:

$$CUL[k].host = A$$

$$CUL[k].ch = D_j$$

$$CUL[k].rel-time = T_{curr} + NAV_{RES}$$

$$CUL[k].int = \begin{cases} 0, & \text{如果 } Power[A] > P_{RES} \\ 1, & \text{如果 } Power[A] \leq P_{RES} \end{cases}$$

(上接第 231 页)

```
<forward name="failed" path="/error.jsp"/>
</action>
</action-mappings>
</struts-config>
```

4 结束语

Struts 是目前非常流行的基于 MVC 的 Java Web 框架。这种框架使系统开发过程各个模块更加细化。用户界面、业务过程、业务逻辑和业务控制的分离,使得程序开发者能很好地协调,开发过程也更简洁,同时也提高了应用软件的柔韧性。由于篇幅有限,文中实例只是简要地讲述了 Struts 的用法,要想真正掌握其开发技巧,还须进一

(9) B 收到 A 的完整数据报文后,用 Power[A] 在数据信道 D_j 上回送 ACK 报文。

3 总结

通过以上分析,能看出 DCA-PC 协议具有以下特点:首先,能按需为节点分配信道。当节点有信号需要传送时,协议通过 RTS/CTS/RES 握手控制报文取得一个信道,传送任务完成后释放该信道。其次,在这种按需分配传送的机制下,网络信道数目的分配与网络的规模,拓扑和复杂度无关。另外,不需要在全网范围内提供时钟同步。具有基于 CSMA 和 FAMAC 的多信道协议不具备的优势。现有的其他多信道协议往往只是单独考虑信道接入和功率控制的问题,而 DCA-PC 独创性地将二者结合起来。研究表明,使用多信道和动态信道分配技术,可以显著提高网络的吞吐量。通过在数据信道上使用功率控制技术,增加了频率的空间复用度,在提高网络吞吐量的同时,也降低了网络的能耗^[5],对改善 WSNs 的性能具有重要意义。

参考文献:

- [1] Zheng Tao, Radhakrishnan S, Sarangan V. PMAC: An adaptive energy-efficient MAC protocol for Wireless Sensor Networks [J/OL]. <http://ieeexplore.ieee.org/Xplore/guesthome.jsp>, 2005-04.
- [2] 郑少仁, 王海涛, 赵志峰, 等. Ad Hoc 网络技术[M]. 北京: 人民邮电出版社, 2005.
- [3] 赵志峰, 郑少仁. 基于多信道的 Ad Hoc 网络信道接入协议性能分析[J]. 军事通信技术, 2002, 23(4): 1-5.
- [4] Tseng Yu-Chee, Wu Shin-Lin, Lin Chih-Yu, et al. A Multi-Channel MAC Protocol with Power Control for Multi-Hop Mobile Ad Hoc Networks[J]. The Computer Journal, 2002, 45(1): 101-110.
- [5] 任丰原, 黄海宁, 林 闯. 无线传感器网络[J]. 软件学报, 2003, 14(7): 1282-1290.

步掌握相关知识。

参考文献:

- [1] 孙卫琴. 精通 Struts: 基于 MVC 的 Java Web 设计与开发[M]. 北京: 电子工业出版社, 2004. 9-16.
- [2] 黄若波, 程 峰, 程繁科. 实战 STRUTS[M]. 北京: 机械工业出版社, 2005. 10-12.
- [3] 丰 伟. Struts 教程[EB/OL]. <http://www.xpbook.com/soft/2779.htm>, 2004-12-16.
- [4] 孙卫琴. 创建 Web 应用和 Struts 框架的配置文件[EB/OL]. <http://linux.ccidnet.com/pub/article/c322-a181127-p1.html>, 2004-11-24.
- [5] 丁 鹏, 刘 方, 邵志峰, 等. STRUTS 技术揭秘及 WEB 开发实例[M]. 北京: 清华大学出版社, 2004. 20-28.