

基于 MVC 设计模式的 Struts 框架的应用方法

寇毅, 吴力文

(大连理工大学 信息管理系, 辽宁 大连 116024)

摘 要: 基于单纯采用 JSP 构建网站时存在扩展性差、不易分工的问题。文中介绍了 MVC 模型和 Struts 框架, 并结合一个实例讨论了如何设计基于 Struts 的应用, 最后提出了 Struts 的一些不完善之处以及在实际应用中的改进方法。

关键词: MVC 模型; Struts 框架; JSP; 设计模式; J2EE

中图分类号: TP311.52 **文献标识码:** A

Application Methods of Struts Framework Based on MVC Designed Pattern

KOU Yi, WU Li-wen

(Department of Information Management, Dalian University of Technology, Dalian Liaoning 116024, China)

Abstract: In building website simply by JSP, there are some problems such as poor expansibility and difficult work division. Based on such problems, this paper introduces MVC model and struts framework and discusses how to design a struts-based application by an example. This paper also discusses the deficiency of struts framework and gives some methods of improving it.

Key words: MVC Model; Struts framework; JSP; design pattern; J2EE

1 引言

当前, SUN 公司提出的 J2EE (Java 2 Enterprise Edition) 已经成为企业 Web 应用开发的标准, 这当中的 Servlet/JSP 技术也在众多网站得到了广泛的应用。然而, 在仅使用 JSP 技术开发的 Web 应用中, 业务逻辑处理、动态显示用的 Java 代码以及 HTML 混合在一起, 使得阅读、调试和维护都困难重重。

为了使开发的 Web 应用具有良好的可扩展性和易于分工, 我们必须要做到:

- (1) 将业务逻辑与用户界面分离;
- (2) 将动态页面显示逻辑与静态的 HTML 代码分离。

由 Apache Jakarta 项目组提供的 Struts 应用框架正是用来解决这些问题的, 它基于 MVC 的设计思想, 很好地实现了业务逻辑与用户界面的分离, 同时, 它提供了大量的自定义标记库, 可以在 JSP 页面中分离动态显示逻辑与静态的 HTML, 使页面代码更清晰, 可读性大大提高。

2 Struts 框架的体系结构

2.1 MVC 设计模式简介

MVC 设计模式来源于面向对象语言 Smalltalk, 它的主要思想是将应用程序分为三个部分: 模型 (Model)、视图 (View) 和控制器 (Controller), 以提高应用程序的可扩展性。这三个部分的作用分别描述如下:

- (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。
- (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。

(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。

MVC 模型在基于 Java 技术的 Web 应用中扩展形成了 MVC 模型 2 (如图 1), 在模型 2 中, Model、View、Controller 分别由 JavaBean、JSP、Servlet 担当。Servlet 先接受客户端的请求, 然后调用相应的 Bean 进行业务逻辑的处理, 根据处理的结果再调用相应的 JSP 页面, JSP 页面依据存储结果的 Bean 生成 HTML 页面, 并返回给客户端。

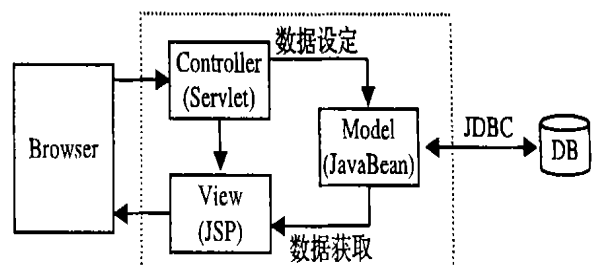


图 1 MVC 模型 2 的结构

Struts 框架就是基于 MVC 模型 2 提供了一套自己的 Servlet、类和标记库。

2.2 Struts 的体系结构

Struts 由一组相互协作的类、Servlet、标记库等构成, 它提供了自己的控制器 (Controller), 同时整合了其它的一些技术去实现 Model 层和 View 层。它的体系结构如图 2 所示。

收稿日期: 2003-05-26; 修订日期: 2003-08-26

作者简介: 寇毅 (1979-), 男, 辽宁大连人, 硕士研究生, 主要研究方向: 信息系统、系统集成、软件工程; 吴力文 (1949-), 男, 辽宁大连人, 副教授, 主要研究方向: 信息系统、电子商务、系统集成。

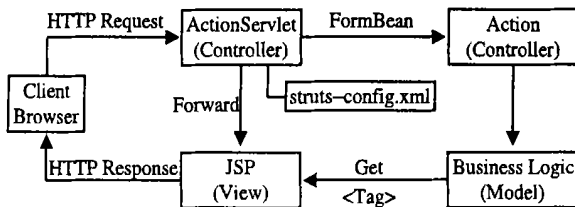


图2 Struts的体系结构

可以从如下几个方面来理解这个体系结构：

(1) 视图：视图主要由 JSP 建立。Struts 自身包含了一组标记库 (TagLib)，这也是 Struts 的精华之一，它们可以简化 JSP 页面的代码，灵活运用它们可以大大提高开发效率。

(2) 模型：在 Struts 中，模型 (Model) 由一系列的 JavaBean 来实现，可以自己定义业务处理的 Bean，Struts 本身则提供了一种非常有用的 Bean——FormBean，它可以在 View 和 Model 间保持数据。

(3) 控制器：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。

3 基于 Struts 框架的应用设计实例

这里我们结合一个实例——某求职网站上的个人自荐系统，来说明如何设计基于 Struts 框架的 Web 应用。在该系统中，个人用户合法登录网站后，可以填写或修改自己的简历，然后把简历提交到服务器保存；用人单位在查询页面里输入查询条件（比如性别、户口所在地、所学专业等），然后提交给服务器，服务器会返回符合条件的人才列表。

做基于 Struts 框架的项目开发，关键是要有一个好的整体模型，计划好系统中包括哪几个模块，每个模块各需要什么样的 FormBean、JavaBean，各种处理结果都通过哪些 JSP 页面来展现，同时应由专人负责配置 struts-config.xml 文件。本系统的设计模型如图 3 所示：

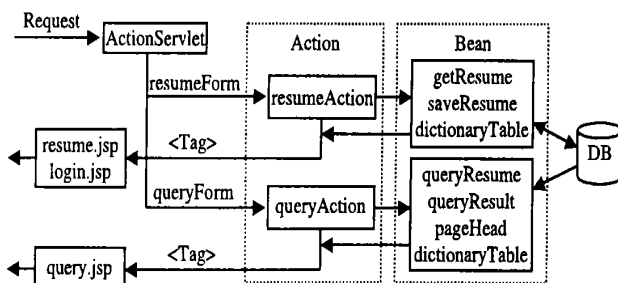


图3 系统的设计模型

ActionServlet 接受所有的 HTTP 请求，然后根据配置文件的内容，决定将请求映射到哪一个 Action 对象，本系统中有两个 Action 对象，分别对应着个人简历的提交 (resumeAction) 和用人单位的查询 (queryAction)。

resumeAction 首先会验证用户是否已经登录，如果没有登录则重定向到登录页面 (login.jsp)，验证通过后根据请求参数决定下一步的处理，如果是查看简历信息，则调用 getResume 对象把该用户的简历读取到 FormBean 中，然后调用 JSP 页面显示 Bean 里的数据；如果是保存简历信息，则调用 saveResume 对象将 FormBean 里保持的简历信息存入数据库。

queryAction 接收用户的查询请求，调用 queryResume 对象

来执行实际的查询工作，查询结果集放入一个列表对象中，列表里的每一项对应一个 queryResult 对象，用来封装一条结果记录，这样在 JSP 页面里通过 Struts 提供的 iterate 标记就能够逐条读取记录。另外，为了分页显示查询结果，我们还定义了一个 pageHead 对象，它封装了每次查询的总页数、当前页码、总记录数，以便在 JSP 页面中利用自定义的标记来显示分页导航的链接信息。

为了在 JSP 页面的下拉列表里显示数据库字典表的内容，我们还事先创建好了 dictionaryTable 对象，它里面封装了几个列表对象，每一个列表对应一个要读取的字典表内容，这样我们就可以利用 Struts 提供的 options 标记自动从 dictionaryTable 对象中读取记录了。

经过这样设计的系统，用户界面和数据处理已经完全分离，再加上在 JSP 页面中使用了自定义标记，使页面中没有了 Java 的脚本代码，这样 Web 界面的设计和后端程序的编写就有了清晰的界线，便于开发团队的分工，并且维护起来也很方便。

4 Struts 应用的进一步改进

Struts 框架虽然有很多的优点，但它的产生时间还很短，有些细节问题还有待完善。就上面的系统来说，我们使用的是 Struts 的当前发布版本 1.0.2，这其中就存在如下的几个问题，我们在此提出相应的改进方法：

(1) 中文请求的处理问题

中文处理的问题在 JSP/Servlet 中由来已久，Struts 的当前版本也没有解决这个问题，并且，由于 ActionServlet 自动从 HTML FORM 中提取请求数据到 FormBean，因此，如果请求数据中包含中文，则 FormBean 中读取到的都是乱码。在上面的系统中，我们采用了 Servlet2.3 中新提供的 filter 特性来解决这个问题。我们只要编写一个编码转换的类实现 filter 接口，并且在 web.xml 文件中做相应的配置，就可以让所有的 request 数据在被接收之前转换为 GB2312 编码。该编码转换类的源程序如下：

```
// ChangeCharsetFilter.java
package myFilter;
import java.io.IOException;
import javax.servlet.*;

public class ChangeCharsetFilter implements Filter {
    protected String encoding = null;

    // 要制定的编码,在 web.xml 中配置
    protected FilterConfig filterConfig = null;

    public void destroy() {
        this.encoding = null;
        this.filterConfig = null;
    }

    public void doFilter ( ServletRequest request, ServletResponse
        response, FilterChain chain ) throws IOException,
        ServletException {
        if (request.getCharacterEncoding() == null) {
            if (encoding != null)
                request.setCharacterEncoding(encoding);
            // 设置 request 的编码
        }
        chain.doFilter(request, response);
        // 有机会执行下一个 filter
    }

    public void init (FilterConfig filterConfig) throws ServletException {
        this.filterConfig = filterConfig;
        this.encoding = filterConfig.getInitParameter ("encoding");
        // 得到在 web.xml 中配置的编码
    }
}
```

在 web.xml 文件中相应的设置如下：

```
<filter>
  <filter-name> SetCharacterEncoding </filter-name>
  <filter-class> myFilter. ChangeCharsetFilter </filter-class>
  <init-param>
    <param-name> encoding </param-name>
    <param-value> GB2312 </param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name> SetCharacterEncoding </filter-name>
  <url-pattern> / * </url-pattern>
</filter-mapping>
```

(2) 对 JSP 页面的保护问题

由于采用 Struts 框架的应用都是将请求发送给 ActionServlet , 因此原则上不允许用户直接访问 JSP 页面, 否则由于缺少 Action 中适当的初始化工作将导致 JSP 页面抛出异常, Struts 没有提供这样的保护机制, 因此我们只有将 JSP 页面集中到一个目录下, 然后配置 Web 服务器针对该目录进行保护。

(3) 一些实用功能并没有 Struts 标签的直接支持

尽管 Struts 提供了丰富的标签库, 但有一些实用的功能仍然没有得到标签库的直接支持, 需要我们自己定义标签来完成。比如上面系统中的分页导航功能, 这是一个比较常见的功能, 要根据当前检索结果的总页数、当前页等信息来显示分

(上接第 78 页)

(2) 64 基编码 (Base64) 的解码

与可打印编码相对应, 64 基编码会增加三分之一的报文大小。要使用 64 基编码, 首先把每个连续的三个字节 (24 位) 数据组合表示为四个六位数 (总共 24 位)。程序把每 6 位数值当作一个 NVT ASCII 字符来传输, 当数据不包含三字节数据块的倍数时, 使用等号填充。如图 3 所示。

本文设计了一个类 CBase64, 用来实现对 Base64 编码字节的解码。我们以四个 6 位字节为一组, 进行解码。首先根据接收到的 4 个 64 基进行查表 (在 64 基编码表中查), 获得对应的 6bits 字节, 然后以下面所示步骤进行解码:

第一步: (第一个 6bits 字节左移 2 位) | (第二个 6bits 字节右移 4 位)

第二步: (第二个 6bits 字节左移 4 位) | (第三个 6bits 字节右移 2 位)

第三步: (第三个 6bits 字节左移 6 位) | (第四个 6bits 字节)

4.3 界面设计

本文的 IMAP4 电子邮件客户软件的主界面是用 VC++ 编写程序实现的。首先是一个主窗口, 主窗口中左边为树型目录窗口, 右边为邮件窗口用于显示邮件。整个界面具有美观简洁、操作方便等特点。图 4 为同服务器连接成功后的界面。

JAMSA 的 64 基编码为: "SkFNUE=" (需要一个填充字节)							
Letter:	J	A	M	S	A		
Hex:	0x4A	0x41	0x4D	0x53	0x41		
Binary:	01 00 10 10	01 00 00 01	01 00 11 01	01 01 00 11	01 00 00 01		
6 Bits:	01 00 10	10 01 00	00 01 01	00 11 01	01 01 00	11 01 00	00 01 00 (用 0 填)
Decimal:	18	36	5	13	20	52	4
Base-64:	S	k	F	N	U	0	E =

图 3

页导航的链接以及页码选择的列表, 如果在 JSP 页面中直接写代码就违背了 MVC 模型的原则, 于是我们自己定义了一个标签, 它从一个 Bean 当中读取相关的信息, 然后自动在页面中生成分页导航的链接, 由于所有的信息都以参数的形式传递给标签, 因此这个标签具有通用性。

5 结束语

Struts 是一种非常优秀的基于 J2EE 的 MVC 应用框架, 虽然从正式发布到现在也只是一年左右的时间, 但已经越来越多地运用于企业平台之上, 我们已在某大型求职网站上成功地应用了 Struts 框架。本文总结了运用 Struts 构建网站的设计思路, 并针对它本身的几点不完善之处提出了改进方法, 这为今后更多的应用采用 Struts 框架提供了参考。

参考文献

[1] 何成万, 余秋惠. MVC 模型 2 及软件框架 Struts 的研究 [J]. 计算机工程, 2002, (6): 274 - 275.
[2] 张砚秋, 陈川, 何明德. 基于 MVC 设计模式构筑 JSP/ Servlet + EB 的 Web 应用 [J]. 计算机工程, 2001, (11): 71 - 73.
[3] Apache Jakarta Project . Apache Struts API Documentation [EB / OL]. Available at <http://jakarta.apache.org/struts/doc-1.0.2/api/index.html>.

5 结论

对于电子邮件的读取目前最常用的协议是 POP3, 它提供了快捷的邮件下载服务, 用户可以利用 POP3 把邮箱里的邮件下载到 PC 上进行离线阅读。但是 POP3 主要的局限在于: 用户在读取邮件前, 必须一次将它们全部下载到自己的 PC 中。而 IMAP4 不仅具有 POP3 的功能, 还很好地解决了这一问题。

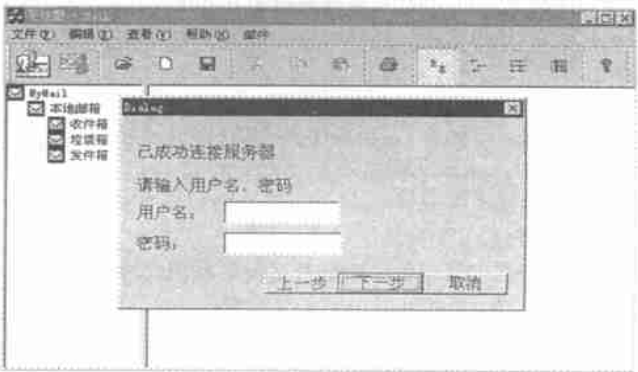


图 4

参考文献

[1] 邓全良. WinSock 网络程序设计 [M]. 北京: 中国铁道出版社, 2002.
[2] 凌云志, 雷晓荣. qmail 实用指南 [M]. 北京: 中国电力出版社, 2001.
[3] RFC 2060. INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev16[S].

论文降重、修改、代写请扫码



免费论文查重，传递门 >> <http://free.paperyy.com>

阅读此文的还阅读了：

1. [基于Struts和Hibernate软件架构实现基金管理信息系统](#)
2. [基于MVC设计模式的Struts框架研究与实现](#)
3. [基于Struts框架实现MVC设计模式的分析与研究](#)
4. [Struts——基于MVC设计模式的应用框架](#)
5. [基于MVC+DAO设计模式的Struts框架应用](#)
6. [基于MVC设计模式Struts框架结构的研究](#)
7. [基于MVC设计模式的Struts框架](#)
8. [基于MVC设计模式Struts框架的应用方法](#)
9. [基于MVC设计模式的Struts框架的研究](#)
10. [基于MVC设计模式的Struts框架应用](#)