



- 书中所有示例的源代码
- 第三方免费插件

Struts
开发利器

真正解决
实际问题

突出实战+剖析详尽+项目案例

Struts

实用开发指南

——基于MVC+MyEclipse的Java
Web应用开发

高红岩 编著

- ◆ MVC架构与Struts的映射
- ◆ 实际应用中如何配置中心配置文件
- ◆ Web应用中控制层和视图层代码的分配
- ◆ Struts三大标签库技术细节及应用
- ◆ 完整解析某典型简易论坛模型案例

 科学出版社

北京科海电子出版社
www.khp.com.cn

TP311.56

336D

2007

Struts 实用开发指南

——基于 MVC+MyEclipse 的 Java Web 应用开发

高红岩 编著

科学出版社

内 容 提 要

这是一本讲解如何使用 Struts 框架开发 Java Web 应用程序的实用性图书, 书中在具体讲解 Struts 开发技术的同时, 结合 MVC 模式与 MyEclipse 工具, 演示了实际应用项目开发的全过程。

全书共分 10 章, 内容以实用为主, 技术和实例相辅相承。书中首先给出一个 Struts 示例, 让读者对程序的结构有大致了解, 然后作者详细介绍了 MVC 架构与 Struts 的映射关系, 并讲解在实际应用中如何配置中心配置文件, 如何分配控制层和视图层代码。接下来, 作者结合具体示例介绍了 Struts 的三大标签库及其应用。最后, 全书以一个简易论坛模型的案例作为全书的终结, 通过这个案例, 读者可以对 Struts 技术在实际 Web 应用开发中的地位及其应用有深入的了解。

本书总结性、归纳性极强, 操作思路明晰, 通篇体现出作者的丰富经验, 既可以作为广大工程技术人员的参考, 也适合大中专院校相关专业的学生学习参考。

图书在版编目 (CIP) 数据

Struts 实用开发指南——基于 MVC+MyEclipse 的 Java Web
应用开发/高红岩编著. — 北京: 科学出版社, 2007
ISBN 978-7-03-019096-3

I. S… II. 高… III. ①软件工具—程序设计②JAVA
语言—程序设计 IV. TP311.56 TP312

中国版本图书馆 CIP 数据核字 (2007) 第 085494 号

责任编辑: 何 武 / 责任校对: 李玉茹 刘雪莲
责任印制: 科 海 / 封面设计: 林 陶

科 学 出 版 社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京市鑫山源印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2007 年 7 月 第 一 版

开本: 787×1092 1/16

2007 年 7 月第一次印刷

印张: 20.00

印数: 0001-4000

字数: 487 000

定价: 36.00 元 (1CD)

(如有印装质量问题, 我社负责调换)

前言

Web 科技的发展催生了很多附属技术，在 Java/JSP Web 领域中就出现了 Struts，Struts 大大提高了跨平台语言 Java 的开发效率。

为什么写本书

目前市面上的 Java 技术书籍已经包含了几乎所有的相关技术，包括 Struts。但在这些 Struts 技术图书中，很少将 MyEclipse 开发工具与 Struts 框架结合起来做详细介绍，这样，读者看完这些 Struts 书后，明白了 MVC 是什么，明白了 Action 要做什么，也会明白那些大大小小的标签的功能，但如果手头有项目开发工作，需要真正运用 Struts，则还是一头雾水。这样的图书是一种字典式风格，图书的写作目的与读者学习后要达到的结果很不一致。

那么，怎样才能满足读者的需求，使其看完书后就能立即上手，把书中学到的技术应用在实际项目开发中？这正是本书写作中贯穿始终的中心思想——让读者能学以致用。

本书汇集了作者几年来从事 Java 和 Struts 编程的开发经验和心得体会，书中每一个案例都是从实战项目中抽取出来，读者认真体会学习，将其略微改造就可成为开发项目中的解决方案。

本书针对 Java 应用最流行的开发工具 MyEclipse 来探讨 Struts 框架技术的实现，步骤详实，讲解通俗易懂，每一步要添加的内容，每一步要达到的目的都跃然纸上，让读者摆脱字典式书籍中的那种痛苦感，一起分享阅读技术书籍的快乐和轻松。

本书特色

(1) 案例丰富。每一节都有经典的技术解决方案，教读者学会使用 Struts 的方法解决 Web 应用问题，以 Struts 的角度学习基于 MVC 模式的开发。全书最后以一个完整的模仿“百度贴吧”的实例，将前面章节的技术进行了综合性汇总。

(2) 标签讲解细致。了解 Struts 的读者很清楚标签在该框架中的重要性，本书重点讲解了 Struts 的三大标签库，书中对每个标签的讲解都伴有示例分析。另外，书中明细列出了一个标签的多种使用方法，同时结合示例代码进行解说，一目了然，学习效果显著。

(3) 归纳总结性强。本书通篇体现出作者在 Struts 方面的经验技术总结。在每个技术讲解之后，作者都会总结归纳技术的要点难点，操作思路明晰，便于学习记忆。读者技术提升空间大。

(4) 源代码支持。书中所有示例的源代码请见本书配套光盘。

本书主要由高红岩编写，参加编写的人员还有王雷、张金果、赵力菲、黄绍义等。本书在写作过程中力求精益求精，但难免存在一些不足，敬请读者批评指正。

高红岩
写于吉林市龙潭区

目 录

第 1 章 Struts 和基于 Eclipse 的配置与开发	1
1.1 几个基本的技术要点	1
1.2 Struts.....	2
1.3 框架 (FrameWork)	2
1.4 JSP 开发的两个 MVC Model 方法	3
1.4.1 MVC Model.....	3
1.4.2 MVC Model 1.....	4
1.4.3 MVC Model 2.....	5
1.4.4 Struts 中的 MVC Model.....	6
1.5 配置 Eclipse 下的 Struts 开发环境	7
1.5.1 配置 JDK 环境变量要点	8
1.5.2 Eclipse 的安装很“绿色”	9
1.5.3 MyEclipse 的安装	9
1.5.4 Tomcat 的安装	13
1.5.5 测试开发环境	15
第 2 章 Struts 简介.....	23
2.1 Struts 的工作流程.....	23
2.2 关于 Struts 的实例.....	25
2.2.1 添加 Struts 框架支持文件	25
2.2.2 视图层 V-View 的添加	28
2.2.3 控制层 C-Controller 的添加	33
2.2.4 模型层 M-Model 的添加	37
2.2.5 结尾前小小的改动	40
2.2.6 struts-config.xml 文件	40
2.2.7 部署项目并运行	42
第 3 章 应用中的 C-Controller 控制层.....	44
3.1 ActionServlet 类的作用.....	44
3.1.1 process()方法的执行过程.....	47
3.1.2 process()方法执行过程总结	48
3.2 Action 类的作用	49

3.2.1	Action 的工作	49
3.2.2	在 Action 类中进行用户名验证的实例	49
3.2.3	ActionErrors 和 ActionError 类的关系和使用	54
3.3	ActionForward 类的功能及两种在 Eclipse 中创建 Action Forward 类的方法	56
3.3.1	ActionForward 类的功能	56
3.3.2	在 Eclipse 中创建 ActionForward 类的两种方法	56
3.3.3	带参数的 ActionForward	57
3.4	使用 ForwardAction 进行页面或 Web 组件的跳转	57
3.4.1	什么情况下使用 ForwardAction	58
3.4.2	一个 ForwardAction 类的实例	58
3.4.3	第二种创建 ForwardAction 类的方法	62
3.5	IncludeAction 让年久失用的 Web 组件复用	62
3.5.1	使用 IncludeAction 包含 JSP 文件	63
3.5.2	使用 IncludeAction 包含进 Servlet 组件	68
3.6	DispatchAction 将 Action 类变得更少	75
3.7	LookupDispatchAction 实现一个表单包含多个提交按钮	79
3.8	用 SwitchAction 切换不同的 Struts 模块	85
第 4 章	中心配置文件 struts-config.xml	94
4.1	Struts 1.2 版 struts-config.xml 文件结构	94
4.2	struts-config.xml 配置文件中的子元素	95
4.2.1	<data-sources />子元素	95
4.2.2	<form-bean />子元素	95
4.2.3	<global-forwards />子元素	95
4.2.4	<action-mappings>子元素	96
4.2.5	<message-resources>子元素	97
4.2.6	关于配置 struts-config.xml 文件	97
4.2.7	元素详解	99
4.2.8	attribute 和 name 的区别	105
第 5 章	V-View 视图层中的 ActionForm	107
5.1	ActionForm 类的结构	107
5.2	ActionForm 生命周期	108
5.3	DynaActionForm 使用方法	109
5.4	DynaActionForm 实例	110
5.5	Action 和 ActionForm 配置精例	115
5.5.1	完整的 action 功能	115
5.5.2	仅有 Form 的 action 超级链接功能	116
5.5.3	仅有 Action 的 action 执行链接式请求后就转发	116

5.5.4 仅有 JSP 的 action.....	117
5.5.5 两个 action 对应一个 Form	117
5.5.6 两个 action 对应两个 Form	118
5.6 ActionForm 中文乱码问题解决方案.....	118
5.7 Struts 中的 ActionErrors	119
第 6 章 Struts-HTML 标签库	120
6.1 普通 HTML 与 Struts 中 HTML 标签的区别及 Struts 标签公共特征	120
6.2 显示 Struts-HTML 标签的 Snippets 窗口.....	121
6.3 <html:base />设置相对根路径	123
6.3.1 标签简介	123
6.3.2 使用示例 1	123
6.3.3 使用示例 2	124
6.4 <html:cancel />取消当前的提交	126
6.4.1 标签简介	126
6.4.2 使用示例	126
6.5 <html:checkbox />复选框	130
6.5.1 标签简介	130
6.5.2 使用示例	130
6.5.3 <html:checkbox />复选框在 Action 中状态的改变.....	132
6.5.4 <html:checkbox />复选框的初始化	133
6.5.5 在 Action 中通过数据库的数据控制<html:checkbox />复选框的选中状态	134
6.6 <html:errors />出错提示.....	138
6.6.1 显示局部错误信息	138
6.6.2 显示全局错误信息	140
6.6.3 生成错误信息在不同版本使用上的区别	142
6.7 <html:file />文件上传功能.....	142
6.7.1 定制 JSP 页面	142
6.7.2 定制 ActionForm 类.....	142
6.7.3 设计重要的 Action 类.....	143
6.7.4 查看 struts-config.xml 配置文件	145
6.8 <html:form />表单提交	145
6.9 <html:hidden />保密的数据传送	147
6.9.1 开发实例	147
6.9.2 <html:hidden>如何设置默认值.....	150
6.10 <html:html>定义 HTML 文件.....	151
6.11 <html:image>定义图像提交按钮	152
6.12 <html:img>在页面上显示图像	153
6.13 <html:link>变幻莫测的超级链接	157

6.13.1	Link type 为 action 的情况	158
6.13.2	Link type 为 forward 的情况	160
6.13.3	Link type 为 href 的情况	161
6.13.4	Link type 为 page 的情况	162
6.13.5	带参数超级链接的情况	163
6.13.6	在<html:link>中嵌入 JSP 脚本	165
6.13.7	带参数超级链接的问题解决实例	165
6.13.8	给 Struts 中<html:link>标签加确认对话框	168
6.13.9	用<html:link>标签生成 BBS 主题列表功能	168
6.14	<html:errors>的更新版<html:messages>	174
6.15	<html:multibox />分组类型的复选框	181
6.15.1	用<html:multibox />做一个选择“爱好”的实例	181
6.15.2	<html:multibox />初始化时即呈 checked 状态	183
6.16	用<html:select />和<html:option />实现下拉列表	187
6.16.1	用<html:select />和<html:option />实现下拉列表	187
6.16.2	<html:select />和<html:option />实现列表单选	188
6.16.3	<html:select />和<html:option />实现列表多选	188
6.16.4	<html:select />和<html:option />标签设置下拉列表初始化值	189
6.16.5	<html:select />和<html:option />设置列表单选初始化值	190
6.16.6	<html:select />和<html:option />设置列表多选初始化值	190
6.16.7	如何获取<html:select />和<html:option />下拉列表单选值	192
6.16.8	如何获取<html:select />和<html:option />列表多选值	194
6.17	使用<html:options>动态生成<html:select />和<html:option>列表内容	195
6.17.1	将数据库的内容动态生成<html:select />和<html:option>列表内容	196
6.17.2	初始化<html:select />和<html:options>列表生成的内容	198
6.18	使用<html:optionsCollection>动态生成 <html:option>列表内容	198
6.18.1	使用<html:optionsCollection>动态生成<html:option>列表内容实例	198
6.18.2	使用<html:optionsCollection>标签中的 property 属性来生成下拉列表	202
6.19	<html:password>、<html:text>、<html:textarea>标签的使用	205
6.19.1	<html:password>标签的 redisplay 属性实例	205
6.19.2	使用 style 的 CSS 样式改变<html:text>标签的外观	207
6.19.3	<html:textarea>标签的使用	208
6.20	<html:radio>标签的使用	208
6.21	<html:submit>和<html:reset>标签的使用	209
6.22	总结	209
第 7 章	Struts-Logic 标签库	210
7.1	<logic:iterate>标签的功能	210
7.1.1	打印数组中的内容	211

7.1.2 打印 HashMap 中的内容	211
7.1.3 打印 ArrayList 中的内容	212
7.2 <logic:redirect>重定向的标签	213
7.3 <logic:forward>转发的标签	214
7.4 <logic:empty>和<logic:notempty>标签的作用	214
7.5 <logic:present>标签的作用和与<logic:empty>的区别	215
7.6 用<logic:equal>和<logic:notEqual> 判断等于和不等	216
7.6.1 使用<logic:equal>和<logic:notEqual>判断变量	217
7.6.2 使用<logic:equal>和<logic:notEqual>判断 Bean 的属性值	217
7.7 用<logic:lessEqual>和<logic:lessThan> 判断小于等于和小于	218
7.7.1 <logic:lessEqual>和<logic:lessThan>判断变量	218
7.7.2 <logic:lessEqual>和<logic:lessThan>判断 Bean 的属性值	219
7.8 用<logic:greaterEqual>和<logic: greaterThan> 判断大于等于和大于	220
7.8.1 <logic:greaterEqual>和<logic: greaterThan>判断变量	220
7.8.2 <logic:greaterEqual>和<logic: greaterThan>判断 Bean 的属性值	221
第 8 章 Struts-Bean 标签库	223
8.1 Bean 标签库的功能	223
8.2 <bean:write />标签打印 Bean 中的属性值	223
8.2.1 <bean:write>标签打印变量	223
8.2.2 <bean:write>标签打印 Bean 的 property 属性值	224
8.2.3 <bean:write>标签 format 属性的应用	227
8.2.4 <bean:write>标签 filter 属性的应用	227
8.3 <bean:parameter />标签读取 HTTP 请求的参数	228
8.3.1 使用<bean:parameter />标签读取单个 http 参数	228
8.3.2 使用<bean:parameter />标签读取数组型 HTTP 参数	229
8.4 <bean:message>标签显示资源文件中的文本消息	230
8.5 在<bean:define>标签中定义变量	235
8.5.1 在<bean:define>标签中定义字符串常量	236
8.5.2 <bean:define>标签复制 Bean	236
8.5.3 用<bean:define>标签复制现有 Bean 的属性给新的 Bean 属性	236
第 9 章 关于 Struts 的其他内容	237
9.1 Struts 资源文件国际化	237
9.1.1 MyEclipse 保存资源文件的编码哨兵	237
9.1.2 使用 MyEclipse 资源文件的插件 jinto	237
9.2 在 Struts 的 URL 中传递中文参数	238
9.3 从不同的资源文件中显示信息	240
9.3.1 从不同的资源文件中显示信息的实例	240

9.3.2 优化新建资源文件目录结构	243
9.4 没有登录不能访问非 index.jsp 的 JSP 页面	244
9.5 设置应用的默认页面	252
9.6 URL 重写技术	254
9.7 使用 Struts 多语言切换的情况	259
9.8 添加 Struts 包的操作	263
9.9 实现跨页表单的提交	265
第 10 章 简易论坛模型的实例	274
10.1 实例目标	274
10.2 功能模块简介	274
10.3 模块设计	275
10.3.1 用户注册	276
10.3.2 显示主题列表	284
10.3.3 用户登录	295
10.3.4 修改个人用户信息	297
10.3.5 查询用户	301
10.3.6 删除用户	307
10.3.7 删除主题及删除回复	308
10.4 总结	309

第 1 章

Struts 和基于 Eclipse 的配置与开发

1.1 几个基本的技术要点

在概述什么是 Struts 之前，我们要先弄清楚几个相关的技术术语：Apache、ASF、Jakarta、Tomcat。

1. Apache

Apache 是一种开放源代码的 Web 服务器，它的功能类似于微软的 IIS，但要远远超过 IIS，不管从安全性、跨平台性，还是一些企业级的部署和应用。Apache 作为自由软件之一，像其他自由软件一样，它们都是由许许多多的自由开发人员投入了大量的时间和精力来实现并逐步完善的，这也是 Apache 能成为最流行的 Web 服务器原因之一。不过从第一个版本以来，尽管不断有新的漏洞被发现，但由于其 OpenSource（开放源代码）的特点，漏洞总能被很快修补，因此，其安全性还是相当高的，使用也是最普遍的。

2. ASF

ASF（Apache Software Foundation）是 Apache 软件基金组织的缩写。随着 Apache 服务器的广泛应用，现在的 Apache 已经不仅代表一个软件，而是具有一些开放源代码及企业级应用的软件项目机构。Apache 软件基金会（ASF）正式创建于 1999 年。

3. Jakarta

ASF 这个组织包含了很多软件项目，Jakarta 是 ASF 旗下的一套 Java 解决方案的开源软件项目的名称，它包括了很多子项目，Tomcat、Ant、Struts 等也是 Apache 下的开源项目，同时也是 Jakarta 的关联项目。Jakarta 里的项目主要面向 Java 技术。

4. Tomcat

Tomcat 是一个免费开源的 Servlet 容器，它是 Apache 基金会 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。由于有了 Sun 公司的参与和支持，最新的 Servlet 和 JSP 规范总能在 Tomcat 中得到体现。Tomcat 不仅仅是一个 Servlet 容器，它也具有传统的 Web 服务器的功能：处理 HTML 页面，简单的域名管理，配置 JNDI 等。但是与 Apache 相比，它的处理静态 HTML 的能力远不如 Apache。可以将 Tomcat 和

Apache 集成在一起, 让 Apache 处理静态 HTML, 而 Tomcat 处理 JSP 和 Servlet, 这样可以发挥各自的所长。

1.2 Struts

Struts 是 Jakarta 的一个子项目, 它提供了一种方法, 可以在一个 Web 应用程序中同时使用 Java Server Pages (JSP) 和 Servlet。它的目的是要解决完全由 JSP 或完全由 Servlet 实现的传统应用程序中固有的问题, 例如, 用 JSP 很难将 Java 代码同网页的数据显示分开, 也很容易将 Java 代码同 HTML 混在一起, 结果做出的项目运行速度较慢, 而且在后期维护中工作量较大。

Struts 只是一个 MVC 框架(Model/View/Controller Framework), 用于快速开发 Java Web 应用程序, 这样以分三层的结构来开发软件项目, 不但使开发的结构明了, 而且还有助于项目的维护。Struts 的重点在 C (Controller) 控制端, 也为 V (View) 视图端提供了一系列定制的标签 (Custom Tag)。但 Struts 几乎没有涉及 M (Model) 模型端, 所以 Struts 可以采用 Java 实现的任何形式的商业逻辑。

MVC (Model/View/Controller) 模式是国外用得比较多的一种设计模式, 最早在 Smalltalk 编程语言中出现。MVC 包括三类对象: Model 是应用对象, 也就是功能逻辑; View 是它在屏幕上的表示, 也就是 UI 界面; Controller 定义用户界面对用户输入的响应方式, 相当于请求和应答的事件。

使用 Struts 是免费的。

1.3 框架 (FrameWork)

框架是一个应用程序的半成品, 框架提供了可在应用程序之间共享的可复用的公共结构。开发者把框架融入到他们自己的应用程序中, 并加以扩展, 以满足他们特定的需要。框架和工具包的不同之处在于: 框架提供了一致的结构, 而不仅仅是一组工具类。框架其实就是一组组件, 供你选用以完成你自己的系统。简单地说就是使用别人搭好的舞台, 你来做表演。而且, 框架一般是成熟的, 不断升级的软件。

可以说, 一个框架是一个可复用的设计构件, 它规定了应用的体系结构, 阐明了整个设计、协作构件之间的依赖关系、责任分配和控制流程, 表现为一组抽象类以及其实例之间协作的方法。

打个比方, 应用在建筑技术中比较常见的就是钢架结构, 以前在建设一所厂房的时候, 都是一砖一瓦堆起来, 工期时间长不说, 而且不坚固; 现在的工业技术, 使用钢架结构, 支起房屋的框架, 然后在这个钢架结构上可以设计出不同样式, 不同风格的厂房, 比如不同颜色的屋顶, 不同金属板的墙壁等, 这样不仅有利于房屋的拆装和维护, 而且在构造时也非常快捷, 具有自由性。

可见, 建筑工程中的技巧也可以完全应用在软件工程中, 给你一个软件的框架, 在这个框架之上可以做自定义的功能。

1.4 JSP 开发的两个 MVC Model 方法

不管对于 Win 32 软件还是 Web 开发, 设计模式的应用无处不在, 在 JSP 的标准发布之后, 使用 JSP 开发的软件项目大致上可以分为两个方法, 或者叫两种模型: MVC Model 1 和 MVC Model 2。

1.4.1 MVC Model

MVC 英文即 Model-View-Controller, 即把一个应用的输入、处理、输出流程按照 Model、View、Controller 的方式进行分离, 这样一个应用被分成 3 个层——模型层、视图层、控制层。

视图 (View) 代表用户交互界面, 对于 Web 应用来说, 可以概括为 HTML 界面, 但有可能为 XHTML、XML 和 Applet。随着应用的复杂性和规模性的不断变化, 界面的处理也变得具有挑战性。一个应用可能有很多不同的视图, MVC 设计模式对于视图的处理仅限于视图上数据的采集和处理, 以及用户的请求, 而不包括在视图上的业务流程的处理。业务流程的处理交予模型 (Model) 处理。比如一个订单的视图只接受来自模型的数据并显示给用户, 以及将用户界面的输入数据和请求传递给控制和模型。

模型 (Model): 就是业务流程/状态的处理以及业务规则的制定。业务流程的处理过程对其他层来说是黑箱操作, 模型接受视图请求的数据, 并返回最终的处理结果。业务模型的设计可以说是 MVC 最主要的核心。目前流行的 EJB 模型就是一个典型的应用例子, 它从应用技术实现的角度对模型做了进一步的划分, 以便充分利用现有的组件, 但它不能作为应用设计模型的框架。它仅仅告诉你按这种模型设计就可以利用某些技术组件, 从而减少了技术上的难度。对一个开发者来说, 就可以专注于业务模型的设计。MVC 设计模式告诉我们, 把应用的模型按一定的规则抽取出来, 抽取的层次很重要, 这也是判断开发人员是否优秀的依据。抽象与具体不能隔得太远, 也不能太近。MVC 并没有提供模型的设计方法, 而只告诉你应该组织管理这些模型, 以便于模型的重构和提高重用性。在这里可以用面向对象编程来做比喻, MVC 定义了一个顶级类, 告诉它的子类你只能做这些, 但没法限制你能做这些。这点对编程人员来说非常重要。业务模型还有一个很重要的模型, 那就是数据模型。数据模型主要指实体对象的数据保存 (持续化)。比如将一张订单保存到数据库, 从数据库获取订单。我们可以将这个模型单独列出, 所有有关数据库的操作只限制在该模型中。

控制 (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。划分控制层的作用也很明显, 它清楚地告诉你, 它就是一个分发器, 选择什么样的模型, 选择什么样的视图, 可以完成什么样的用户请求。控制层并不做任何的数据处理。例如, 用户单击一个连接, 控制层接受请求后, 并不处理业务信息, 它只把用户的信息传递给模型, 告诉模型做什么, 选择符合要求的视图返回给用户。因此, 一个模型可能对应多个视图, 一个视图可能对应多个模型。

模型、视图与控制器的分离, 使得一个模型可以具有多个显示视图。如果用户通过某

个视图的控制器改变了模型的数据，所有其他依赖于这些数据的视图都应反映到这些变化。因此，无论何时发生了何种数据变化，控制器都会将变化通知所有的视图，导致显示的更新。这实际上是一种模型的变化——传播机制。模型、视图、控制器三者之间的关系和各自的主要功能如图 1-1 所示。

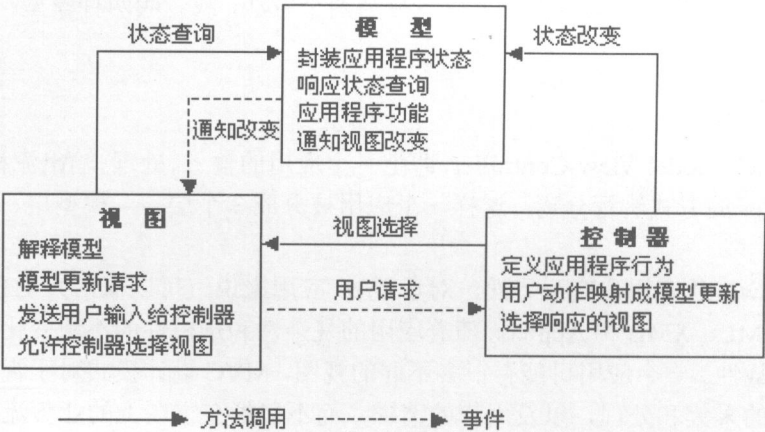


图 1-1 MVC 组件关系和功能

1.4.2 MVC Model 1

在最初开发 JSP 项目过程中，使用最广泛的方法就是 JSP+JavaBean，虽然这是一种对于大型软件项目来说是过时的技术，但这样的方法直到现在有些公司还在乐此不疲地使用，也许是由于项目的需要。虽然这里要学习 Struts，但 JSP+JavaBean 的开发方法还要熟练掌握，它是学习 Struts 的基本功。MVC Model 1 在 UI 图形用户界面端 JSP 文件里，夹杂着大量的 JSP 脚本和 HTML 语言代码及一些 JavaScript 脚本，这就加大了程序的调试、维护的难度。想想，如果在一个包含 10 万行的 JSP 文件中，调试程序的情景是多么可怕。

从图 1-2 可以看出，JSP 是整个应用系统的门户，主要做 3 个重要工作。

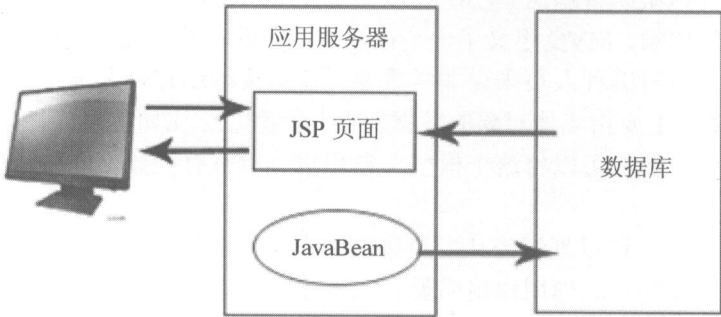


图 1-2 MVC Model 1

- 负责与客户端的所有通信。
- 处理所有的请求。
- 处理所有的答复。

在处理答复的时候，从数据库中存取数据有如下两种方式。

- JSP 自己直接去存取。
- JavaBean 来完成。

因为 JavaBean 可以被放在一个请求上下文或者用户会话中，这样就可以在不同的 JSP 页面之间通信。当然这种模型在开发需求不是很复杂、规模较小的 Web 应用的情况下有很大的优势，比如，JSP 页面可以非常容易地结合业务逻辑（<jsp:useBean>）、服务端处理过程（<jsp:scriptlet>）和 HTML（<html>）。在 JSP 页面中同时实现显示、业务逻辑和流程控制，从而可以快速地完成应用开发。

从工程化的角度考虑，它也有一些不足之处：把表现层和业务逻辑层的程序代码揉和在一起，不利于以后的维护工作以及开发角色的分配，所以这种模式只能适合于小的系统开发和单人开发。

1.4.3 MVC Model 2

由于 MVC Model 1 将表现层和业务逻辑层程序代码揉和在一起，容易产生对软件开发的维护、管理、调试不利的一面，Servlet/JSP 规范的 0.92 版描述了在一个应用中使用 Servlet 和 JSP 的架构，这也就是现在所说的 MVC Model 2，该模型结构即 JSP+Servlet+JavaBean。该 MVC Model 2 又称做以 Servlet 为中心（Servlet Centric）的设计模型。

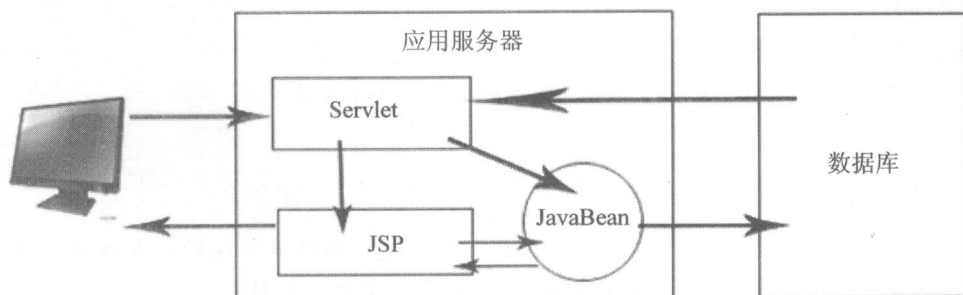


图 1-3 MVC 2 模型

对比图 1-3 与图 1-2 的结构，从 JSP 这个角度来看，JSP 页面至少少了两个任务，即获取和处理用户的请求，因为 Servlet 相当于控制器（Controller）角色，它负责接收客户端请求并处理此请求，将它传递给合适的 JSP，而 JSP 则负责显示给用户。所以 JSP 页面这时候主要做以下两件事情。

- JavaBean 直接与数据库打交道取得数据后，JSP 从 JavaBean 中读取数据。
- 把结果返回给客户端。

根据 MVC Model 2，Servlet 处理数据存取和导航流，JSP 处理表现。MVC Model 2 强制性准则使 Java 软件工程师和 HTML 用户界面设计者分别工作于他们所熟悉的部分。MVC Model 2 应用功能的一部分发生改变并不会牵连其他部分也跟着发生改变。例如，HTML 设计者可以改变软件的外观和风格，并不需要改变后端 Servlet 的工作方式。

任何一种解决方案都是双刃剑，在我们获得的同时必须有一定的付出。MVC Model 2 也带来了如下一些实际的问题。

- 必须基于 MVC 组件的方式重新思考和设计应用结构。原来通过建立一个简单的 JSP 页面就能实现的应用现在变成了多个步骤的设计和实现过程。
- 所有的页面和组件必须在 MVC 框架中实现，所以必须进行附加的开发工作。

以上两点也在 Struts 框架中存在，但并不能掩盖 Struts 优点的光辉。

1.4.4 Struts 中的 MVC Model

Struts 已经提供了一个非常好的 MVC 框架，利用 Struts 开发 MVC 系统时可以大大加快开发速度。在开发时可以采用的开发流程如下。

- (1) 收集和定义应用需求。
- (2) 基于数据采集和显示的原则定义和开发用户界面的需求。
- (3) 为每一个用户界面 JSP 文件定义访问路径。
- (4) 定义 ActionMapping，建立到应用业务逻辑之间的联系。
- (5) 开发满足用户界面需求的所有支持对象。
- (6) 基于每一个用户界面需求提供的属性来创建对应的 ActionForm 对象。
- (7) 开发被 ActionMapping 调用的 Action 对象。
- (8) 开发应用业务逻辑对象 (Bean, EJB 等)。
- (9) 对应 ActionMapping 设计的流程创建 JSP 页面。
- (10) 建立合适的配置文件 struts-config.xml 和 web.xml。
- (11) 开发/测试/部署。

具体在使用 Struts 框架时，对应各个部分的主要开发工作如下。

- Model 部分：采用 JavaBean 和 EJB 组件，设计和实现系统的业务逻辑。根据不同的请求从 Action 派生具体 Action 处理对象，完成“做什么”的任务来调用由 Bean 构成的业务组件。由 ActionForm 的派生类实现对客户端表单数据的封装及简单的检验。
- Controller 部分：Struts 为我们提供了核心控制部分的实现。只需要配置 ActionMapping 对象即可完成 URI 地址的映射及匹配用户界面表单和 ActionForm 类的对应关系。
- View 部分：为了使用 Model 中的 ActionForm 对象，必须用 Struts 提供的自定义标记创建 HTML 表单，即利用 Struts 提供的自定义标记库编写用户界面把应用逻辑和显示逻辑分离。Struts 框架通过这些自定义标记建立了 View 和 Model 之间的联系。Struts 的自定义标记还提供了很多定制页面的功能。
- 需要编辑两个配置文件：web.xml 和 struts-config.xml。通过它们配置 Struts 系统中各个模块之间的交互。

图 1-4 描述了一个 Struts 应用程序的简要执行流程。