



普通高等教育“十一五”国家级规划教材

王颖玲 编著

基于Struts和Hibernate 技术的Web开发应用

21世纪计算机科学与技术实践型教程

丛书主编 陈明

清华大学出版社



“十一五”国家级规划教材

王颖玲 编著

基于Struts和Hibernate 技术的Web开发应用

21世纪
计算机
科学
与
技术
实
践
型
教
程

丛书主编
陈明

清华大学出版社
北京

内 容 简 介

本书介绍了开发轻量级 J2EE Web 应用的流行框架 SSH 中的 Struts 和 Hibernate 框架,以及在开发工作中用到的 MyEclipse 和 MySQL 软件的常用操作。全书共分 7 章,从架构思想的引入和架构技术的分析开始,逐步讲解各种技术架构的原型实现,并以登录、注册、退出三项基本功能进行架构的实例讲解。最后通过一个完整的案例整合 Struts+Hibernate 架构,并给出详细的开发步骤分析。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

基于 Struts 和 Hibernate 技术的 Web 开发应用/王颖玲编著. —北京:清华大学出版社, 2011.9

(21 世纪计算机科学与技术实践型教程)

ISBN 978-7-302-26145-2

I. ①基… II. ①王… III. ①软件工具—程序设计—高等学校—教材 ②JAVA 语言—程序设计—高等学校—教材 IV. ①TP311.56 ②TP312

中国版本图书馆 CIP 数据核字(2011)第 122622 号

责任编辑:谢琛 薛阳

责任校对:时翠兰

责任印制:杨艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260

印 张:9.5

字 数:223 千字

版 次:2011 年 9 月第 1 版

印 次:2011 年 9 月第 1 次印刷

印 数:1~4000

定 价:20.00 元

产品编号:040593-01

《21 世纪计算机科学与技术实践型教程》

编辑委员会

主 任：陈 明

委 员：	毛国君	白中英	叶新铭	刘淑芬	刘书家
	汤 庸	何炎祥	陈永义	罗四维	段友祥
	高维东	郭 禾	姚 琳	崔武子	曹元大
	谢树煜	焦金生	韩江洪		

策划编辑：谢 琛

《21 世纪计算机科学与技术实践型教程》

序

21 世纪影响世界的三大关键技术：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制定科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21 世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21 世纪计算机科学与技术实践型教程》就是为了填补这部分空白。将理论与实际联系起来，使学生不仅学会了计算机科学理论，而且也学会应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，他们把积累的经验、知识、智慧、素质融合于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细的思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编 陈明

2005 年 1 月于北京

前 言

为了帮助众多初学者快速掌握轻量级 J2EE 的开发方法,笔者精心编著了本书。它是笔者在项目实战中的经验总结,也是笔者在 Struts/Hibernate 技术的高职教学生涯中根据学生的学习规律和讲义整理而成的。本书根据读者的学习规律,首先通过实例介绍基本概念和基本操作,然后在读者掌握了这些基本概念和操作的基础上,再对内容进行深入的讲解,严格遵循由浅入深、循序渐进的原则。本书按照掌握 Struts/Hibernate 知识的先后顺序进行编排。本书对于每一个实例,从环境配置开始,到最后的运行都有详尽的介绍,从而使读者很容易就能运行实例,掌握开发方法,并体会到学习的快乐,不断增强学习的动力。

本书从头到尾都是按照读者的学习兴趣安排知识点的。我们本着用到什么介绍什么的原则,介绍了开发基于 Struts 和 Hibernate 框架的应用程序的最小软件集合和操作集合。目的就是开发为向导介绍技术,而不是为技术而技术。

本书适合的读者

本书介绍了开发轻量级 J2EE Web 应用的流行框架 SSH 中的 Struts 和 Hibernate 框架,以及在开发工作中用到的 MyEclipse 和 MySQL 软件的常用操作。使用本书的读者需要具备 Java Web 应用开发的基本技能。本书适合用作 Struts 和 Hibernate 框架的入门书籍。

通过对本书的学习,读者可以掌握 J2EE Web 开发技术,同时通过本书的实战演练,能够积累一定的实际开发经验和技巧,掌握 J2EE Web 开发的思路,为以后的提高打下坚实的基础。

本书可以作为计算机实践型课程的教材,也可以作为 IT 培训机构的教材或是自学材料。

本书特点

- ✓ 讲解通俗易懂:本书在行文中追求朴实易懂,语言尽量简练,以易读性为第一要务。在编写时充分站在读者的角度描述问题,对于每一个案例,给出全面详尽的步骤分析和操作说明。
- ✓ 结构主次分明:本书着重讲解开发中常用的技术和工具,使得读者在学习首先掌握最关键的开发技术,而不用为技术难题所困扰。当读者逐步熟悉开发所使用的常用技术和工具之后,通过进一步的研究学习将很容易地进行技术的升级,并解决开发中遇到的难题。

- ✓ 由浅入深：为了让读者能很快地进行 J2EE Web 应用开发，每一章都从一个简单的应用示例入手，让读者快速了解本章工作内容，然后再详细讲解本章涉及的基本原理和知识。最后再通过一个详细的示例来巩固学习成果。这种学习过程适合初学者的接受规律。
- ✓ 实战性强：本书遵循面向工作过程的思想编著而成，在理论部分过后，提供了一个具有原型性的网上购物子系统的需求，以供读者进行练习，融会贯通前面各个章节的内容，从而使读者可以通过本书的学习快速进行实战项目的开发。

本书的组织结构和主要内容

本书内容总体上分为 7 章，从架构思想的引入和架构技术的分析开始，逐步讲解各种技术架构的原型实现，并以登录、注册、退出三项基本功能进行架构的实例讲解。最后通过一个完整的案例整合 Struts+Hibernate 架构，并给出详细的开发步骤分析。各部分的内容如下：

1. J2EE Web 架构基础篇：开发基于 JSP、JavaBean、Servlet 技术的 MVC 模式的原型系统。

2. Struts 架构篇：开发 Struts 架构原型。

(1) 通过原型系统的 Struts 框架开发，讲解 Struts 原理和核心组件 Action 和 ActionForm。

(2) 通过原型系统的优化，讲解 DispatchAction 的应用。

(3) 通过原型系统视图组件的扩展，讲解 Struts 的标签库和国际化处理。

3. Hibernate 架构篇：开发 Struts+Hibernate 架构原型。

(1) 讲解 Hibernate 的核心技术，包括持久化对象、Hibernate 基础对象、配置文件和集成 Struts 进行 Web 开发的过程。在 Struts 架构基础上增加 Hibernate 支持。

(2) 在 Struts 架构基础上增加 Hibernate 支持，重点关注 Hibernate 查询的使用。

4. 实战篇：开发基于 Struts+Hibernate 架构的学生成绩与课程管理系统。

(1) 讲解项目需求。

(2) 以功能为单位展开分析。

(3) 以 MVC 思想贯穿讲解架构的整合。

尽管笔者尽了最大努力，本书难免会有不妥之处，欢迎各界专家和读者朋友批评指正。

作 者

2011 年 8 月

目 录

第 1 章 J2EE 和 MVC	1
1.1 J2EE 简介	1
1.1.1 J2EE 是什么	1
1.1.2 J2EE 技术	2
1.1.3 轻量级 J2EE 开发	3
1.2 软件架构	3
1.2.1 MVC 模式	3
1.2.2 N 层架构	6
1.3 构建 MVC 应用之登录功能	6
1.3.1 功能需求描述	6
1.3.2 登录功能应用架构分析	7
1.3.3 数据库分析与建立	8
1.3.4 视图层实现	8
1.3.5 控制层实现	12
1.3.6 模型层实现	15
1.3.7 运行	20
1.4 实验与能力拓展	23
第 2 章 Struts 基本原理和应用	25
2.1 Struts 入门	25
2.1.1 Struts 简介	25
2.1.2 Struts 开发环境的配置	26
2.2 Struts 基本原理与核心组件	29
2.2.1 Struts 基本原理	29
2.2.2 struts-config.xml 配置文件	30
2.2.3 Struts 控制器组件	32
2.2.4 Struts 的 FormBean	33

2.3	开发基于 Struts 的应用	35
2.3.1	需求说明	35
2.3.2	开发基于 Struts 的用户登录功能	35
2.3.3	结合案例回顾 Struts 原理	42
2.4	实验与能力拓展	44
第3章	使用 DispatchAction 优化控制层	45
3.1	默认的 Action 类	45
3.1.1	默认的 Action	45
3.1.2	解读 Action 类的 execute() 方法	45
3.2	DispatchAction	46
3.2.1	使用 DispatchAction 的必要性	46
3.2.2	DispatchAction 的使用	46
3.3	使用 DispatchAction 改进原型系统	47
3.3.1	使用 DispatchAction 为原型系统添加注册功能	47
3.3.2	使用 DispatchAction 为原型系统添加退出功能	51
3.4	实验与能力拓展	52
第4章	使用 Struts 标签进行页面处理和国际化	53
4.1	Struts 中国际化处理	53
4.1.1	国际化	53
4.1.2	资源文件	54
4.1.3	国际化处理过程	57
4.2	Struts 视图组件	58
4.2.1	Struts-html	58
4.2.2	Struts-bean	61
4.2.3	Struts-logic	64
4.3	为原型系统添加国际化处理	67
4.4	实验与能力拓展	71
第5章	Hibernate 入门	72
5.1	Hibernate 介绍	72
5.1.1	基础理论	72
5.1.2	Hibernate 简介	73
5.2	Hibernate 应用框架	74
5.2.1	Hibernate 体系结构	74
5.2.2	Hibernate 核心组件	74

5.3	Hibernate 核心	75
5.3.1	Hibernate 配置文件	75
5.3.2	Hibernate 映射文件	76
5.3.3	Hibernate 运行原理	79
5.4	应用 Hibernate 开发原型中的注册功能	79
5.4.1	Hibernate 应用开发流程	79
5.4.2	Hibernate 开发原型中的注册功能	80
5.5	实验与能力拓展	92
第 6 章	Hibernate 查询	93
6.1	Hibernate 查询介绍	93
6.2	HQL 查询基础	94
6.2.1	from 子查询	94
6.2.2	select 子查询	94
6.2.3	where 子查询	95
6.2.4	order by 子查询	96
6.2.5	统计函数查询	96
6.2.6	联接查询	97
6.3	Criteria 查询	97
6.3.1	Criteria 查询的使用步骤	97
6.3.2	创建 Criteria 查询	98
6.3.3	使用 Restriction 类为查询增加限制	98
6.4	应用 HQL 查询实现原型系统的登录功能	99
6.5	实验与能力拓展	101
第 7 章	项目练手：网上购物子系统	102
7.1	系统概述	102
7.2	系统功能演示	103
7.3	系统设计	105
7.3.1	数据库设计	105
7.3.2	创建数据库	105
7.3.3	目录和包结构	107
7.4	工程准备	107
7.5	工程的设计和实现	108
7.5.1	配置工程的 Struts 和 Hibernate 框架支持	108
7.5.2	为工程添加公共类	108
7.5.3	实现 DAO 模式的公共类	111

7.6	用户管理功能的设计和实现	111
7.6.1	用户管理功能的逻辑设计	111
7.6.2	用户管理功能的模型层实现	112
7.6.3	登录和注册功能的视图层实现	123
7.6.4	用户管理功能的控制层实现	132

第 1 章 J2EE 和 MVC

本章导读

本章主要介绍本书所要使用的软件架构,并基于这种思想,实现了一个用户管理的原型系统,包括登录、注册和退出功能。对于本章思想的理解有助于读者在后续的章节中从整体上把握各种技术框架和应用的实现脉络。

本章主要目的是使读者了解 J2EE 的基本组成,分层的软件设计思想——MVC,以及其应用实现。

本章的介绍是基于读者已经具备了 JSP、Servlet 等 Java Web 开发的基本知识的假设展开的。如果读者还没有具备相应的知识,请参考相关书籍。

工作任务

1. 采用 MVC 架构实现用户的登录功能。
2. 采用 MVC 架构实现用户的注册功能。
3. 采用 MVC 架构实现用户的退出功能。

1.1 J2EE 简介

1.1.1 J2EE 是什么

目前,Java 2 平台有三个版本,分别是适用于小型设备和智能卡的 Java 2 平台 Micro 版(Java 2 Platform MicroEdition, J2ME)、适用于桌面系统的 Java 2 平台标准版(Java 2 Platform Standard Edition, J2SE)、适用于创建服务器应用程序和服务的 Java 2 平台企业版(Java 2 Platform Enterprise Edition, J2EE)。

J2EE 是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构。J2EE 技术的基础就是核心 Java 平台或 Java 2 平台的标准版, J2EE 不仅巩固了标准版中的许多优点,例如“编写一次、随处运行”的特性、方便存取数据库的 JDBC API、CORBA 技术以及能够在 Internet 应用中保护数据的安全模式等,同时还提供了对 EJB(Enterprise JavaBean)、Java Servlets API、JSP(Java Server Pages)以及 XML 技术的全面支持。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的

体系结构。

通过提供统一的开发平台, J2EE 降低了开发多层应用的费用和复杂性, 同时提供了对现有应用程序集成的强有力支持, 完全支持 Enterprise JavaBean, 有良好的向导支持打包和部署应用, 添加目录支持, 增强了安全机制, 提高了性能。通过图 1-1 可以了解 J2EE 平台中的主要容器。

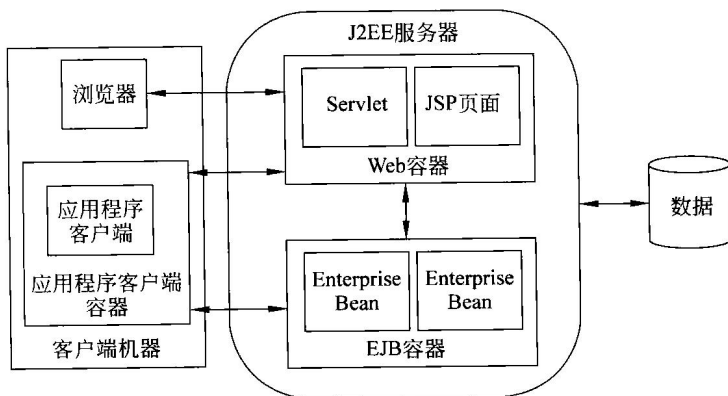


图 1-1 J2EE 框架简图

采用了 J2EE 后的 B/S 结构如图 1-2 所示。

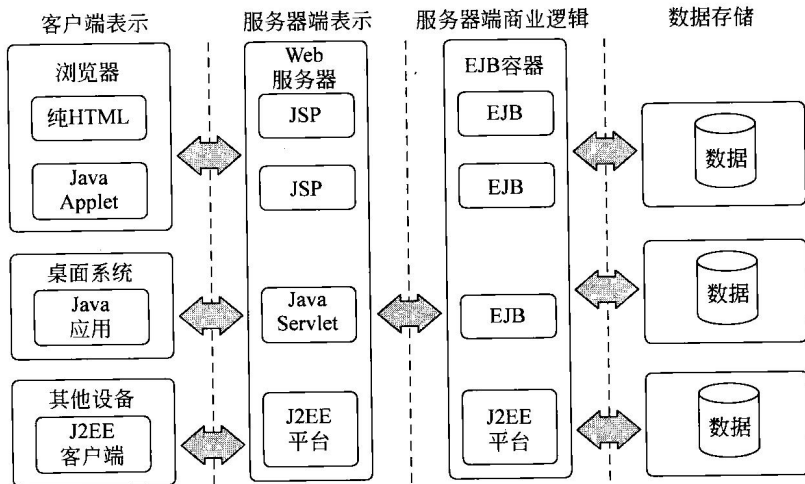


图 1-2 J2EE 应用层次图

通过图 1-1 和图 1-2 可以看出, J2EE 平台开发出来的 Web 应用是采用封装思想对功能分类分层的。通过这种做法, 可以提高应用程序的内聚性, 降低耦合度, 提高组件的可重用性。J2EE 平台支持的应用开发可以降低后期的维护代价, 增强应用的可扩展性。

1.1.2 J2EE 技术

J2EE 定义了一个框架和相关的规范, 而实现这个框架的具体工具需要第三方厂商来

完成,不同厂商提供的不同工具的实现方式也是不一样的,比如 Tomcat、WebSphere 和 WebLogic 等,它们包含了 J2EE 定义的若干种技术规范。J2EE 核心技术及其应用如图 1-3 所示。

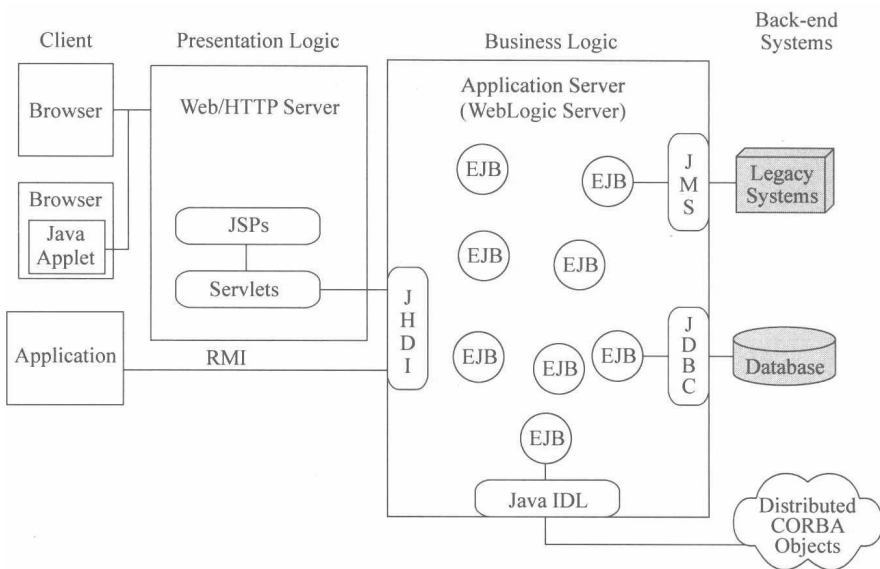


图 1-3 J2EE 核心技术图

1.1.3 轻量级 J2EE 开发

本书不可能对 J2EE 的每一种技术进行详细的介绍,也不可能使读者通过本书能成为 J2EE 专家。本书仅介绍开发 J2EE 应用的两种架构 Struts 和 Hibernate,而不涉及 EJB 等技术。这里还假定使用本书的读者已经掌握了进行 Java Web 开发所必需的 JSP、Servlet 技术。

通常,将不使用 EJB 进行的 J2EE Web 应用开发称为轻量级 J2EE 开发。

1.2 软件架构

1.2.1 MVC 模式

软件架构(software architecture)用于指导大型软件系统各个方面的设计。软件架构是一个半成品,有利于降低开发和维护的成本,使得工作易于开展(就像简历模板、书信格式一样可以给人的工作带来便利),尤其是对新接手的开发人员来讲。

MVC 架构就是能带来上述便利的一种模式。那么 MVC 究竟是什么呢?

MVC 的英文全称是 Model-View-Controller,中文的意思是“模型-视图-控制器”。MVC 模式起源于 Smalltalk 语言,它是 Xerox PARC 在 20 世纪 80 年代为编程语言 Smalltalk-80 发明的一种软件设计模式。MVC 模式的结构由以下三个部分组成:模型

(Model)、视图(View)、控制器(Controller)。有时也称 MVC 模式为三层架构。

要想理解并有效地使用 MVC,必须分层理解 MVC,还必须理解该架构中的三个部分之间是如何通信和合作的。下面通过图 1-4 来初步理解 MVC 三层架构和 Web 应用的关系。

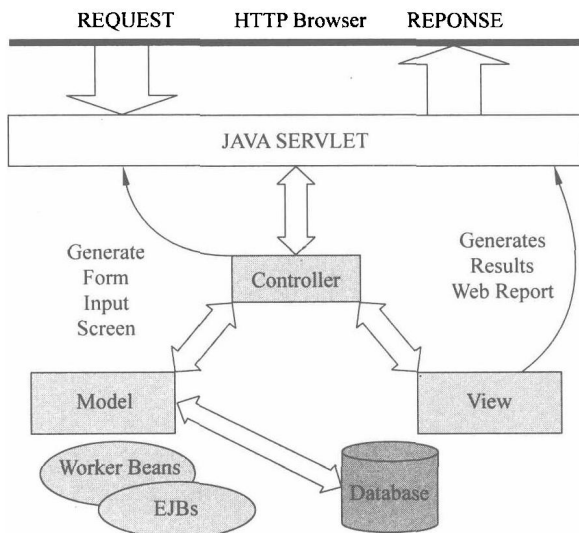


图 1-4 Web 应用中的 MVC 之 Model 2

按照模型层、视图层、控制层进行分解,从而使得整个系统责任明确、接口清晰,各项开发可以同步进行,加快了设计开发的过程。下面分别从几个方面或角度理解 MVC 的 Model2 模式在 Web 应用中的运行。

1. 整体关系

各个模块是通过明确定义的接口通信合作的。在一个 Web 应用中,模型、视图和控制器的设计和实现是独立的,不同分工的人员按照统一的设计接口进行各个部分的实现,然后进行集成。

(1) 浏览器(HTTP Browser)负责显示视图层(View)的各个视图,是客户端和系统交互的唯一介质。

(2) 客户和 Web 应用的交互,即请求的接收和响应的返回都通过控制器进行。而控制器在 Model2 模式中一般由 Servlet 实现。

(3) 后台数据库(Database)中存放 Web 应用的必要数据,但是,数据的处理只通过模型(Model)进行。

(4) 所有经过控制器传递过来的请求或经由 Model 产生的处理结果都由控制器来协调,在模型与合适的视图之间进行分发。

2. MVC 的各个元素

(1) 模型层这里是指业务逻辑的处理和数据的存储。它分为两类模型:业务逻辑模

型和数据模型。模型接收通过控制器(在 Web 应用中通常由 Servlet 担当控制器的角色)转发来的来自视图的请求数据,并返回最终的处理结果。MVC 并没有提供模型的设计方法,而只告诉开发人员应该如何组织管理这些模型,以便于模型重构和提高重用性。**数据模型**是指对数据的持久化,它实现了对视图和模型之间交互的支持。实现是把“做什么(业务处理)”和“怎么做(业务实体)”分离,这样可以实现业务逻辑的重用。对一个开发者来说,就可以专注于业务模型的设计。

(2) **视图层**主要用来展现用户所需的数据,它是用户和系统进行交互的界面。这部分工作可以由美工人员进行开发和维护,一般可以采用 HTML 页面、XML、Servlet 和 Applet 等技术。一般来说,视图只接收来自模型的数据并显示给用户,然后将用户界面的输入数据和请求传递给控制。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,业务流程的处理和状态的改变则交给模型层处理。

(3) **控制层**就是一个分发器。选择什么样的模型处理请求,根据模型处理结果选择什么样的视图,可以完成什么样的用户请求都由控制层决定。控制层就像一个中转站,它从用户那里接受请求,并根据用户的请求,将模型与视图匹配在一起,共同完成用户的请求。

通过将模型、视图和控制器分离,使得一个模型可对应多个视图,一个视图也可以对应多个模型。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将反映出这些变化。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,导致数据的更新。

模型、视图、控制器三者之间的关系和各自的主要功能,即 MVC 模式的功能如图 1-5 所示。

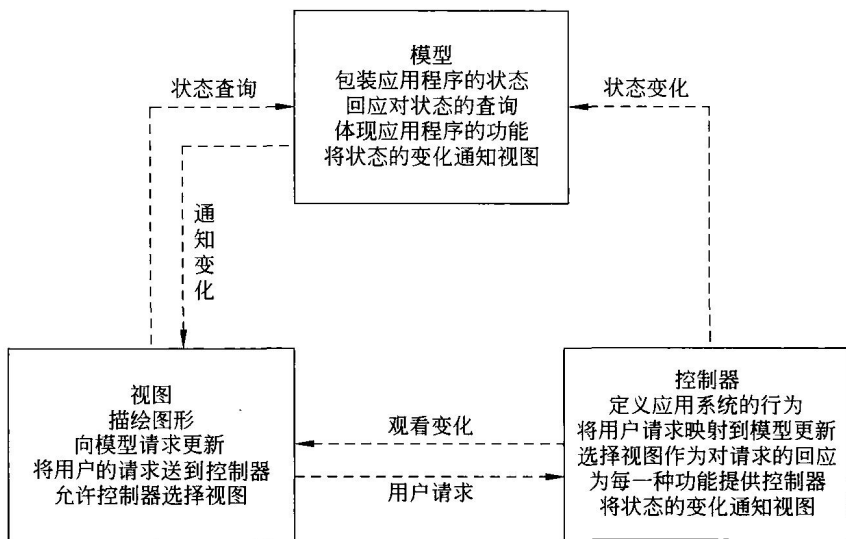


图 1-5 MVC 模式的功能示意图

1.2.2 N 层架构

三层 MVC 架构分离表示和业务逻辑,给工作的展开和后期的维护带来了极大的便利,提高了工作效率,但是,在实际项目工作中,往往会对三层 MVC 结构做进一步的分层处理,来满足不同复杂程度的项目施工要求。因为扩展层次的不确定性,人们通常称多于三层架构的软件架构为 N 层架构或多层架构。

一种常用的 N 层架构是 4 层架构,即表示层(View)、控制层(Controller)、业务逻辑层(BLL)、数据访问层(DAO+Data Object)。实际上是对前述的 MVC 架构中的模型层进行了拆分。

本书中采用的基于 Struts 和 Hibernate 实现的简易 BBS 管理系统就是采用 N 层(4 层)架构实现的。

1.3 构建 MVC 应用之登录功能

1.3.1 功能需求描述

这里选择的案例是大部分 Web 应用系统中都具有的用户登录、注册和退出功能,在本书后面,称这几项功能组成的系统为原型系统。为了理解用户登录和注册的功能要求,下面先从以下几个方面进行分析说明。

1. 用例图

这里要实现的原型系统中的功能包括三个:登录、注册与退出,如图 1-6 所示。

系统的用户主要有三类:未注册用户、登录用户和未登录用户。这三类用户和原型系统中的三个功能的关系及功能描述如下。

1) 登录功能

只有注册了的用户才能登录进入系统。登录功能界面只需要用户提供用户名和密码即可。如果输入正确的用户名和密码,就可进入系统的欢迎界面,否则转到注册页面。

2) 注册功能

未注册的用户必须注册一个用户账号才能执行系统内的特定操作。注册需要提供用户名、密码、确认密码和 E-mail 信息。这里的限制是用户名不能重复。注册成功的用户进入登录页面,可以经由登录功能登入系统。

3) 退出功能

用户登录系统之后,即可操作系统。假定在本原型系统上开发了扩展功能,操作完毕之后,经由退出操作可以注销个人登录信息,使用户失去对系统的管理操作权,提高系统

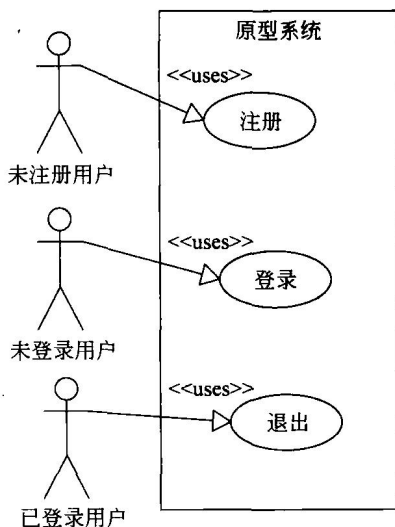


图 1-6 原型系统用例图