

基于 MVC 模式的快速 Web 应用开发 平台的设计与实现^{*}

王 禹, 王建新, 江 南, 王 斌

(中南大学 信息科学与工程学院, 湖南 长沙 410083)

摘 要: 在探讨 Web 应用开发平台的基础上, 提出了基于 MVC(Model- View- Controller) 模式的 Web 应用平台的开发方法, 并详细论述了应用开发平台的核心体系结构、技术实现方案及其关键技术。利用该快速 Web 应用开发平台可以迅速搭建企业级的 Web 应用, 其代价要比利用传统的 J2EE 开发平台低, 具有良好的应用前景。

关键词: Web 应用平台; MVC 模式; 应用逻辑组件框架; 对象 - 关系映射基础组件

中图法分类号: TP31; TP393 文献标识码: A 文章编号: 1001-3695(2004) 11-0204-03

Design and Implementation of Rapid Web Application Platform Based on MVC Pattern

WANG Yu, WANG Jian-xin, JIANG Nan, WANG Bin

(College of Information Science & Engineering, Central South University, Changsha Hunan 410083, China)

Abstract: Introduces development technique about MVC pattern rapid Web application platform on the basis of exploring Web application platform and detailedly discusses it 's kernel architecture, technic realization scheme and key technologies. Based on this rapid Web application development platform, the enterprise Web application can be constructed quickly, and its cost is fewer than traditional J2EE development platform, so it has glorious prospect.

Key words: Web Application Platform; MVC Pattern; Application Logic Framework; Object- Relational Mapping Component

近年来, 随着网络技术和 Internet 的迅速发展, 基于 B/S 模式的 Web 应用, 因其具有易用性、通用性和良好的可扩展性等优点而发展迅速, 正逐渐成为企业应用信息系统市场的主流。但是, 当前基于 Java 的 Web 应用软件开发过程中, 还存在如下一些不尽人意的地方^[1]:

(1) 程序可重用程度很低。多个项目常常有功能类似的部分, 但由于设计之初没有考虑到或不能深入考虑, 导致程序虽然类似却不能重用, 只能重写的局面。完成一个项目积累下来的只有经验而没有代码, 造成工作量的浪费。

(2) 维护工作烦琐困难。即使不考虑由于需求模糊造成的追加, 项目后期维护仍然烦琐。用户的几乎任何一点修改都需要程序员参与才能够完成, 而这类维护中有近一半以上的修改都只需要非常简单的改动就可以完成。

(3) 程序应变能力较弱。由于当前应用软件的业务和用户均不够成熟, 需求的变化就是一个客观存在的必然事实, 而目前应用软件的应变能力无论从技术体系还是程序编码都不够强, 这也导致了维护工作的困难。

为了能够有效地提高 Web 应用程序的可重用性、可维护性和可扩展性等方面, 我们结合 MVC(Model- View- Controller, 模型 - 视图 - 控制器) 模式来设计一个基于 J2EE 技术的 Web

应用开发平台, 它能够提供大量的高度抽象且功能强大的基础组件框架; 特别地, 它还能提供丰富灵活的客户化程序和应变体系。基于这个 Web 应用开发平台用户或程序员可以快速地编写出自己的应用软件项目或产品^[2]。

1 平台的体系结构与实现方案

1.1 Web 应用平台的中心体系结构

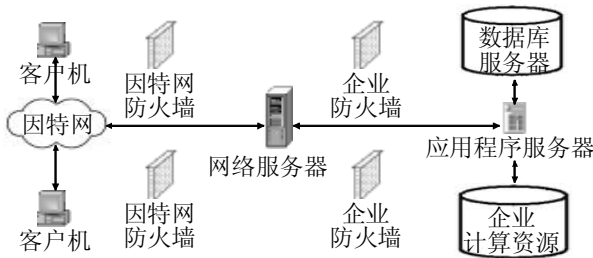


图 1 Web 应用平台的中心体系结构

图 1 说明了 Web 应用平台的中心体系结构的各个组件, 每个服务器代表所提供的服务。服务器实际可能位于同一机器, 或对于大型的 Web 主机设备而言, 可能存在几个各自执行相似任务的服务器。这类 Web 应用体系结构的长处之一是大部分的 Web 应用程序可以通过向用于传输应用程序的服务器组或 Web 区添加更多机器, 升级以支持更多用户。该 Web 应用平台体系结构中各部分的内容有:

(1) Web 浏览器客户机。在 Web 应用程序中, 用户接口通过一个 Web 浏览器进行传输。浏览器理解的主要语言是超文本标记语言 HTML, 大部分当前浏览器还具有执行 JavaScript

和 Java 的内置能力。当前主要应用的浏览器有 IE, Netscape 和 Opera 等。

(2) Web 服务器。它处理 HTTP 请求, 并确定如何生成一个请求响应。在本体系结构中, Web 服务器用于响应静态 Web 页, 把请求传递到一个应用程序服务器, 以及接收应用程序服务器的返回内容以生成响应内容。当前的平台采用主流的 Apache 服务器作为 Web 服务器。

(3) 应用程序服务器。该平台是基于 J2EE 的架构, 应用程序服务器包括服务程序引擎和 EJB 容器。本平台是基于 J2EE 架构的轻量级 Web 应用实现, 主要采用服务程序引擎部分来实现应用逻辑。服务程序引擎还提供对 JSP 的支持, JSP 是与平台和 Web 服务器无关的。该平台采用的服务程序引擎来自 Apache 的应用服务器 Tomcat 4. 1. 12。

(4) 数据库服务器。在该体系结构中, 数据库可以通过运行 EJB 或服务程序的 JDBC API 程序调用进行访问。该平台的数据库服务器采用 MS SQL Server 2000。

(5) 传统遗留系统。该平台采用的 Java 技术是将不同系统连接在一起的应用程序的理想平台。JDBC 允许建立大量通用数据库的统一数据库连接, Java 消息服务(JMS) 是实现提供消息中介服务标准化接口的良好起点, Java 与 XML 及简单对象访问协议(SOAP) 的组合可以提供从支持不同语言和通信机制的各种系统到传统应用程序的访问。

(6) 防火墙。防火墙及代理了限制应用程序在随机套接字上通信的能力。如果客户机上的代码需求独立于浏览器之外与服务器通信, 则该通信应采用 HTTP 或 HTTPS 协议。

1. 2 Web 应用平台的技术实现方案

Web 应用平台的程序结构是基于 MVC 模式来实现的。MVC 模式中包括三类组件: Model 组件用于描述应用程序中的数据以及操作该数据的方法, 与用户界面无关; View 组件负责将数据展现给用户; Controller 组件将用户操作翻译为对 Model 组件的操作。接到 Controller 传来的操作之后, Model 再更新 View, 以反映出数据的变化^[3]。

Web 应用平台的技术实现主要采用 JSP, Java Servlet, JavaBeans 和 JDBC 等技术来实现基于 J2EE 架构的轻量级 Web 应用的开发。其中, 通过 JDBC 进行数据库访问的 JavaBeans 代表模型(数据), Servlet 充当控制器(处理请求), JSP 页面则是模型的视图^[4]。

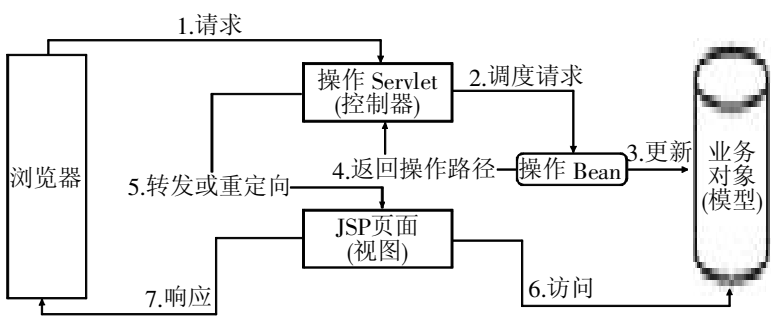


图 2 Web 应用平台的程序结构

如图 2 所示, Web 浏览器发出 HTTP 请求后, 通过 Web 服务器接收后传递, 然后由应用服务器中的服务引擎调用操作 Servlet 来处理, 操作 Servlet 再将请求调至 JavaBeans, 即操作 Bean。操作 Bean 更新代表业务对象模型的数据 Bean, 并向操作 Servlet 返回一个操作路径选择器。操作 Servlet 利用这个路径选择器将请求转发或重定向至 JSP 页面, 接着, JSP 页面访

问通常具有定制标记的业务对象, 并向浏览器发回响应。

2 平台的主要功能与组成内容

Web 应用开发平台作为 Web 应用系统开发的通用化基础平台, 其主要的功能要求有:

- (1) 将数据建模、数据显示与用户交互三者分开, 使得程序设计的过程更清晰;
- (2) 简化程序扩展时的副作用分析, 从而提高系统的可扩展性;
- (3) 将应用程序的功能封装在众所周知的 API 后面, 提高系统的可维护性, 减少重复代码, 将应用程序的功能和数据表现分开, 提高可复用程度;
- (4) 提高系统的灵活性, 把数据模型、用户交互和数据显示等部分设计为可接插组件;
- (5) 可以用于部分发布, 支持渐近式升级;
- (6) 提供自定义的系统通用组件集, 包括改善数据库访问性能的连接池组件, 用于应用程序错误处理的组件以及其他实现通用化的用户与权限管理等组件。

开发平台的内容包含一系列互相合作的类、Servlet 控制程序、JSP 定制标记库以及一些实现通用化功能的组件集, 它们共同组成了可重用的基于 MVC 的框架。

开发平台主要由以下几个部分组成: 表示控制器(Controller) 部分的 Web 应用逻辑组件框架, 表示视图(View) 部分的 Web 用户界面组件框架, 表示模型(Model) 部分的对象 - 关系映射(ORM) 的基础组件, 以及有关的 JSP 定制标记组件库和其他通用化组件集。

3 平台设计与实现的关键技术

(1) 控制器功能的 Web 应用逻辑组件框架的设计与实现从 JSP 页面或 Servlet 调用中接收请求, 调用相应的操作 Bean 及数据 Bean 实现应用逻辑处理, 最后再转移到其他的 JSP 页面或 Servlet 调用的操作。应用逻辑框架中包括四种类型的对象:

- Action(接口) 实现此接口的应用程序的特定操作;
- ActionFactory(类) 生成操作实例;
- ActionServlet(Servlet) 将请求映射到操作;
- ActionRouter 将请求转发或重定向至 JSP 页面。

应用逻辑框架中的对象类型在运行过程中的协作关系如图 3 所示。一般情况下, 可以从 JSP 页面或另一个 Servlet 调用操作 Servlet(ActionServlet), 作为表单提交或链接激活的结果。根据请求的类型, 该 Servlet 重新检索相应的操作类型。操作 Servlet(ActionServlet) 从操作库中获得一个操作后, 它便调用此操作的 Perform 方法, Perform 方法实现了特定应用程序的功能, 通常用于更新业务对象。Action. perform 返回一个操作路径选择器, 它具有一个 URI 和 Boolean 变量, 指示是否将请求转发或重定向至这个 URI。操作 Servlet 使用操作路径选择器调用路径选择器的 Route 方法, Route 方法将请求转发或重定向到适当的 Web 组件^[5, 6]。

基于框架增加应用开发时, 可以考虑按以下几步实现: 实现 Action 操作, 使其能操作业务对象模型, 而且可能在适当

的范围内为 JSP 页面(视图)存储 Bean,以便访问; 实现 JSP 页面,使其能访问业务对象以及特定范围内的 Bean; 增加到应用程序属性文件的映射,以便使上述产生的操作及页面与逻辑名等同。

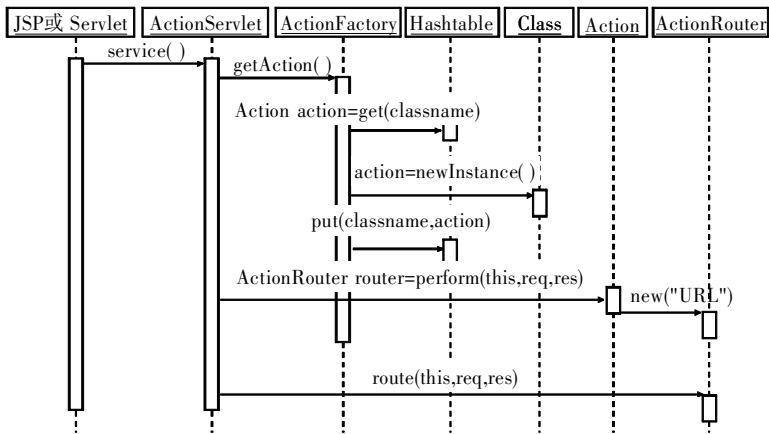


图 3 Web 应用逻辑组件框架中对象类型的协作顺序图

(2) 模型功能的对象 - 关系映射的基础组件的设计与实现

通过数据 Bean 来存取后台数据库的相应数据,即将 Java 对象映射到数据库表中,也就是要创建一个 Java 对象来代表数据库表的其中一行,然后初始化它的各个域,并且通知对象将其添加到数据库中。此即对象到关系映射技术(Object- Relational Mapping, ORM)。平台的 ORM 基础组件设计一个基础类,要能够简化数据 Bean 代码的一个抽象数据库对象类。并且,每一个映射到数据库表的 Java 类必须能够提供它们自己的添加、更新和删除语句,同时还必须能够从 ResultSet 对象中读出各个域。该抽象类 DatabaseObject 的概要说明如下:

```
public abstract class DatabaseObject {
    public Vector getAll( Connection conn) throws SQLException{ }
    public Vector getAll( Connection conn, String whereClause) throws
SQLException{ }
    //返回表中符合条件的所有对象
    public Vector getAll( Connection conn, String whereClause, String addi-
tionalTables) throws SQLException{ }
    //执行一个指定连接的查询串
    public Vector executeQuery( Connection conn, String query) throws
SQLException{ }
    //插入对象到数据库中
    public int insert( Connection conn) throws SQLException{ }
    //返回表中用于 Select 的字段列表
    public abstract String getFieldList( );
    public String getFieldList( String tableName) { };
    public abstract String getInsertStatement( );
    public abstract void prepareInsertStatement ( PreparedStatement s )
throws SQLException;
    //修改数据库中的对象
    public int update( Connection conn) throws SQLException{ }
    public abstract String getUpdateStatement( );
    public abstract void prepareUpdateStatement ( PreparedStatement s )
throws SQLException;
    //删除数据库中的对象
    public int delete( Connection conn) throws SQLException{ }
    public abstract String getDeleteStatement( );
    public abstract void prepareDeleteStatement ( PreparedStatement s )
throws SQLException;
    public abstract String getTableName( );
    public abstract DatabaseObject createInstance ( ResultSet results )
throws SQLException;
}
```

所有业务应用需要的数据 Bean 都可以从这个抽象对象基
础类上继承出来,再结合具体的要求进行相应的扩展。

(3) 视图功能的 Web 用户界面组件框架的设计与实现

HTML 表单用于 Web 应用程序,而用户界面框架(如
Swing 和 AWT)用于传统软件,页面制作者使用表单而不是用
户界面工具包来创建用户界面。所以,为 Web 应用的开发提
供一个类似传统软件用户界面框架的 Web 用户界面组件框
架,这样的作用是能够大大提高 Web 应用界面开发中的代码
重用,也给开发人员提供了一个界面开发中能不断扩展的基
础平台。在 Web 应用表示层的视图部分, JSP 规范没有直接支持
表单, Web 应用平台在此基础上实现 Bean 表单框架,也就是开
发平台的 Web 用户界面框架。而且 JSP 没有提供界面工具中
类似于组件、容器或布局管理器的对象。但是 JSP 具有定制标
记和包含 Web 组件的能力两个特色,利用这两个特色,可以实
现定制的组件、容器和布局管理器。所以, Web 应用平台通过
设计节点、区域和模板实现网页的基础组件来实现可扩展的、
可重用的和可维护的 Web 应用的界面部分。

(4) 平台的 JSP 定制标记组件库和其他通用化组件集的
设计与实现

它包括用于优化数据库访问效率的数据库连接池,系统初
始化设置、客户端的显示、上传下载与页面显示分页等通用控
制组件,以及实现加密解密、第三方打印组件和通用的系统用
户与功能权限管理模块等内容。基于该平台的 Web 应用开发
过程中,可以直接使用上述公用组件所提供的通用化功能,从
而减少了开发 Web 应用系统的工作量。

4 结束语

基于 MVC 模式的 Web 应用平台具有如下特点:能够快速
完备地搭建 Web 应用系统;采取设计模式来扩展显示逻辑、应
用逻辑以及数据模型的复用度;结合区域与模板重用的定制
JSP 页面;提供优化数据库访问的连接池以及丰富的、通用的
工具组件。通过实践证明,基于此通用化平台开发的 Web 应
用系统能够具有以下优良特点:使应用程序模块化;减少了
HTML 和 Java 代码的相关性;允许开发人员为相同的数据提供
多个视图;简化了应用程序流程,使得应用程序更易维护,是一
种进行 Web 应用开发的可靠模型。

参考文献:

[1] 蒋步星 . Web 应用软件开发平台 ExPlatO- Web[C] . 全国第四届
Java 技术与应用交流会, 2001. 285-288.
[2] 齐勇, 等 . 基于 Web 的中间件系统集成框架——应用服务器的
研究[J] . 计算机研究与发展, 2001, 38(4) : 430-437.
[3] 姚慧广, 赵岳松 . Web 编程中 MVC 模型的应用[J] . 微机发展,
2002, 11(3) : 9-10.
[4] 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, 28(6) : 274-281.
[5] 王斌, 王建新, 张尧学, 等 . 基于通信的 MAS 内部自动服务协商
[J] . 小型微型计算机系统, 2003.
[6] 王斌, 张尧学, 陈松乔 . 分布式环境下代理协同的主动黑板结构
设计模式[J] . 计算机工程, 2003.

作者简介:

王禹, 男, 硕士研究生, 主要研究方向为网络信息系统; 王建新, 男, 教
授, 博士, 主要研究方向为计算机网络理论; 江南, 男, 硕士, 主要研究方
向为软件工程; 王斌, 男, 博士, 主要研究方向为组件技术。