

J2EE开发模式的选择

胡以谰 张立平 (北京理工大学计算机系 100000)

摘要: 本文介绍了MVC开发模式的优点和实现方式以及MVC的应用实例。

关键词: J2EE MVC Servlet JSP EJB

1 引言

对于过去C/S结构的开发方式,我们不必考虑太多的开发模式选择方面的问题,但是随着互联网的发展以及多层结构的普及,开发模式的问题就变的越来越重要了,这里我们给大家介绍一下基于多层结构的MVC开发模式。

2 MVC开发模式介绍

如果要开发一个好的J2EE应用,首先正确选择设计模式。什么是设计模式呢?设计模式简单的说就是一个可重用的方案。在我们的开发过程中,经常会遇到一些相同或者相近的问题,每次都会去寻找一个新的解决方法,为了节省时间提高效率,我们提供一些能够解决这些常见问题的,被证实可行的方案,构成一个统一的资源库。一个设计模式描述了一个被证实可行的方案。这些方案非常普通,是有完整定义的最常用的模式。这些模式可以被重用,有良好的伸缩性,而这些设计模式的优势将在设计J2EE应用时得到体现。

很多开发基于J2EE应用的开发人员都是从传统C/S结构开发或从ASP开发模式下转型过来的,所以他们的开发方法也沿袭了旧的设计模式。但是这种开发方式对于新的技术新的要求已经捉襟见肘,已经不适合互联网业的发展要求。旧的开发模式中,程序员会把页面设计、商业逻辑、数据逻辑等都放在同一个程序中,这种状况会导致程序代码过于混乱不利于管理,可重用性差,不利于团队开发等问

题。为了更好的解决这些问题我们推荐一种开发模式即MVC开发模型。

2.1 M: Model

模型,主要用Servlet,EJB,CICS,MQ等技术实现。用来处理由Controller传递过来的对数据的访问请求,根据请求生成返回结果集。EJB是Sun公司根据Java语言的特点以及Web应用开发的需要制订的一套分布式组件结构开发规范。EJB主要用于封装商业逻辑和数据逻辑。CICS,MQ实现交易处理的功能。开发人员可以把用户的商业逻辑,数据逻辑封装成独立的组件,以实现EJB组件的可重用性。

2.2 V: View

视图,用于生成输入,输出页面,主要用JSP,XML技术实现。JSP类似于ASP技术,即在HTML静态页面上加入动态JSP标记以实现显示动态内容。JSP技术非常简单易学,可以让开发人员在掌握较少编程技巧的基础上就能开发网页,使开发人员把工作重心放在页面设计上,而不必关心数据的出处和逻辑。

2.3 C: Controller

控制器,是一系列接收用户动作的对象,他们把用户的请求转换成Model可理解的请求,并决定显示那一个View。主要用Servlet技术实现。Servlet是在服务端执行的Java小程序其实现的功能类似于CGI实现的功能。

所有的商业逻辑、数据逻辑都作为Model,View通过Model访问数据,并根据客户端的要求来显示数据。View必须保证当Model改变的时候,数据显示也必须同时改变。Controller用来结合Model和View,把客户端来的请求转换成Model能够理解并执行的请求,并且根据请求以及执行结果来决定下一次显示那一个View。MVC结构适用于那些多用户的,可扩展的,可维护的,具有很高交互性的系统,那么采用这种开发模式有什么好处呢?

(1) 有利于团队分工合作。对于Web开发,分工合作是尤其重要的事。如果采用MVC开发模式,一般我们会把开发团队分成几个小组:一组负责内容设计,一组负责程序流程,一组负责

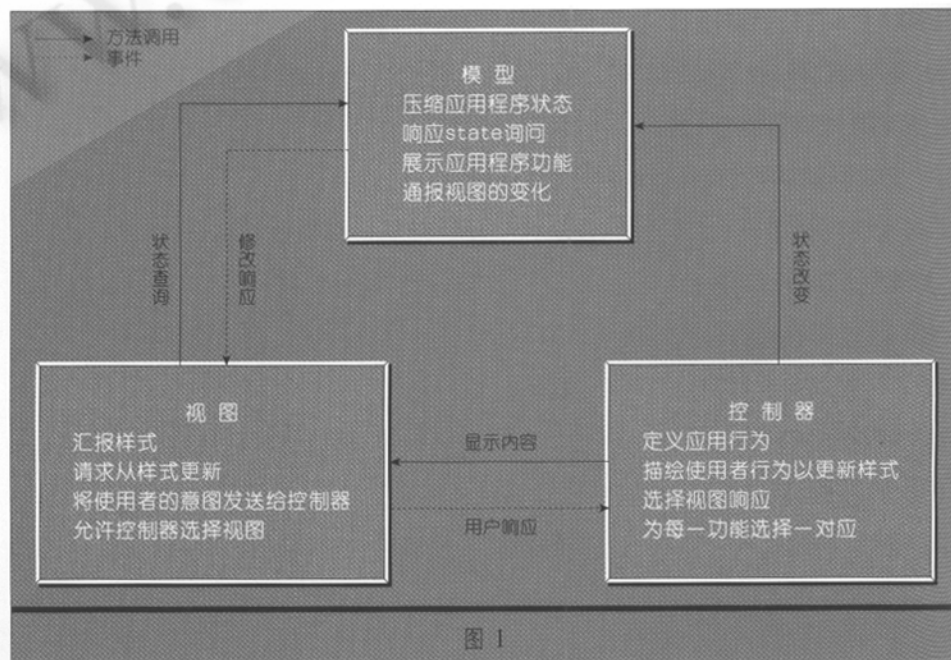


图 1

表 1

	内容设计	程序流程设计	商业逻辑设计	站点管理
角色	页面设计	应用程序设计	组件设计	管理员
技术	HTML, JSP	Servlet, DataBean	AccessBean, EJB	
技巧	美术排版技巧	Java 技巧	Java 技巧, 行业知识	
工具	WebSphere Studio Application Developer (简称 WSAD)			WebSphere

商业逻辑。作为项目设计人员则只要设计好各个模块之间的接口, 各小组就可以各自独立工作了。项目设计人员可以利用象 Rational Rose 等建模工具进行模型设计, 由于各组采用黑盒子开发方式, 所以可以分清开发者的责任。另外, 也可以加快产品推向市场的时间。

(2) 有利于代码的可重用性。由于采用 MVC 开发模式可以单独封装页面设计和商业逻辑, 数据逻辑等不同的模块, 使您的项目实现组件化、模块化, 因此单独封装的组件可用在不同的应用系统中, 尤其有利于与 ERP 系统的集成。在当前的 Web 应用中与 ERP 集成已经成为很多项目必须完成的工作, 所以实现 Web 项目的组件化开发是非常必要的工作。

(3) 有利于后期管理和维护。作为项目管理人员来说不是一个项目开发完成项目就结束了, 项目后期还有很多工作要完成。一个成熟的软件要不断进行升级以满足用户新的需求, 而采用 MVC 模式, 就可以很容易的进行软件升级。例如: 一个采购网站, 随着客户业务的变化给客户的折扣和业务算法也会有所改变, 用户界面及流程则没有任何变化, 那么开发人员只需要重新开发商业逻辑组件, 而其他模块可以不用做任何改动, 就完成客户网站的升级。

由于以上原因, 所以建议大家在开发自己的 Web 项目中首先要考虑好开发模式的问题。有了好的开发模式, 才会有好的项目管理。

3 实例介绍

下面介绍怎样运用该模式来设计实现一个电

子商务网站。

3.1 系统设计

一个典型的 B2B 网站包括的功能是订货、采购、配送、企业 ERP 系统。我们以其中的客户提交订单为例实现 MVC 的设计模式。我们的设计将从模型 (Model)、视图 (View)、控制器 (Controller) 这三方面来介绍。

客户提交订单的过程是: 客户通过浏览器发送订货请求给服务器端 Web 服务器, Web 服务器接收该请求并根据客户的不同级别决定给客户什么样的折扣, 根据不同的折扣向数据库提交订单信息并将处理后的结果返回到客户的浏览器中。

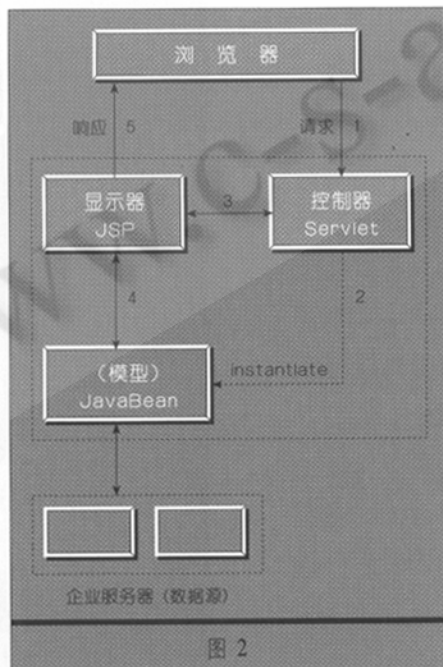


图 2

3.2 控制器设计

控制器是模型和视图联系的纽带, 同时也是系统的控制中心。我们用一个 Servlet 程序来实现

控制器的功能。该程序实现的功能是接受客户提交的订单请求, 调用 JavaBean 把客户订单信息写入数据库, 调用 JSP 页面显示确认信息。

这里我们只列出 Service 方法的源码:

```
public void service(HttpServletRequest req,
    HttpServletResponse res) throws ServletException,
    IOException {
    try
    {
        order-no = request.getParameter("order-no"); // 订
        单号
        product-no = request.getParameter("product-no"); /
        / 产品号
        number = request.getParameter("number"); // 数量
        cont = request.getParameter("cont"); // 金额
        customer-no = request.getParameter("customer-
        no"); // 客户编码
        OrderDB O-jb = new OrderDB(); // 订单提交的
        JavaBean
        O-jb.setOrder-no=order-no; // 给 JavaBean 的参数
        赋值
        O-jb.setProduct-no=product;
        O-jb.setNumber=number;
        O-jb.setCont=cont;
        O-jb.setCustomer-no=customer_no;
        O-jb.execute(); // 执行 JavaBean 中的 execute 方法
        向数据库提交数据
    }
    catch (Exception e)
    {
        response.sendError(500, "Target parameter not
        specified");
        return;
    }
    response.sendRedirect("/Result.jsp"); // 返回给用户确
    认页面
    }
```

3.3 视图设计

视图是与客户交互的窗口。这里，视图就是指HTML页面和JSP动态页面。我们用Input.html文件输入订单信息，经过控制器的处理生成返回的Result.jsp页面。

订单输入页面Input.html如下：

订单号	1000-00209
产品号	CXB-021
数量	
金额	
提交 撤销	

客户看到的返回页面是Result.jsp如下：

订单确认	
订单号	1000-00209
产品号	CXB-021
数量	2
金额	1000
客户名称	姜河
交货时间	2002-01-15

3.4 模型设计

模型是系统的数据核心，是用以描述该系统状态的各种实体的总和。在Web应用中我们一般用EJB或JavaBean来实现数据模型的封装。这里我们为客户订单信息封装了一个JavaBean。由于篇幅的限制这里我们只列出了程序接口。

```
public class orderDB extends java.lang.Object implements java.io.Serializable
{
    public String setOrder-no() { return order-no;} // 给订单号赋值
    public String setProduct-no() { return product-no;} // 给产品号赋值
    public String setNumber() { return number;} // 给数量赋值
    public int setCount() { return cont;} // 给金额赋值
    public String setCustomer_no() { return customer_no;} // 给客户号赋值
    public int setSucceed() { return succeed;} // 给是否成功标志 位赋值
    public String getOrder-no() { return order-no;} // 取订单号
    public String getProduct-no() { return product_no;} // 取产品号
    public String getNumber() { return number;} // 取数量
    public int getCont() { return cont;} // 取金额
    public String getCustomer-no() { return customer-no;} // 取客户号
```

```
public int getSucceed() { return succeed;} // 取是否成功标志
public void Execute() { } // 向数据库中提交数据
}
```

4 结束语

一个成功的项目必然是建立在一个好的开端上才能成功。因此，在我们进行开发前，首先要确立好开发模式。以上介绍的MVC的开发模式只供大家参考。开发人员在选择自己的开发模式和开发工具时，可以根据自己的实际情况和项目情况来确定到底用什么方式和工具。但是一个好的开发模式和一种优秀的开发工具确实可以达到事半功倍的效果。 ■

参考文献

- 1 《设计模式：可复用面向对象软件的基础》作者：Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.
- 2 《J2EE Design Pattern》
- 3 《JavaTM 2 Platform, Standard Edition, v 1.4.0 API Specification》

