

一种软件开发框架的设计优化^①

袁立国, 陈中育, 李方鹏, 郭 婷

(浙江师范大学 数理与信息工程学院, 金华 321004)

摘 要: 针对三层架构存在的缺陷, 提出一种五层开发框架的设计方法. 采用 MVC 设计模式引入了控制器, 该控制器降低了表示层和业务逻辑层之间的耦合度; 通过 ORM 技术增加了数据持久层, 解决了关系数据模型和面向对象模型之间的“阻抗不匹配”问题.

关键词: 三层架构; 软件开发框架; MVC; ORM

A Kind of Design and Majorization of Software Development Framework

YUAN Li-Guo, CHEN Zhong-Yu, LI Fang-Peng, GUO Ting

(College of Mathematics Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004, China)

Abstract: In order to solve the defects that consist in 3-tier application, a design method of 5-tier application is presented. By using MVC design pattern we introduced the Controller which can reduce the coupling between UI and BLL, while, by using the technology of ORM we add the tier of Data Persistence which can solve the problem of impedance mismatch between Relation Data Model and Object Oriented Model.

Key words: 3-tier application; software development framework; MVC; ORM

1 引言

随着用户要求的不断提高, 系统的复杂度和代码量逐渐增大. 为了细化开发人员的分工、有利于系统的维护和代码的复用等, 通常把整个系统分为若干个层次^[1]. 目前, 最流行的是三层架构设计, 其架构模型如图 1 所示:

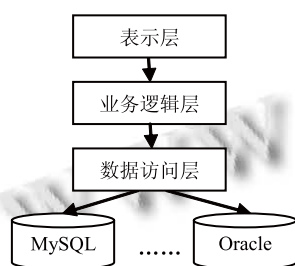


图 1 三层架构图

三层架构的优点主要表现为: 开发人员可以只关注整个结构中的某一层; 可以降低层与层之间的依赖; 有利于各层逻辑的复用^[1]. 但是, 在长期的使用中, 三

层架构逐渐暴露出以下缺陷: 第一, 表示层和业务逻辑层虽然在逻辑上实现了分离, 但是在物理上两层的代码并没有实现完全分离, 存在较高的耦合度^[2]; 第二, 关系型数据库和面向对象编程语言之间存在“阻抗不匹配”问题^[3].

针对三层架构中存在的问题, 国内一些学者给出了自己的解决办法. 文献[2]中将 MVC 设计模式与三层架构相结合, 很好的解决了表示层和业务逻辑层之间的高耦合度问题; 文献[3]中利用 ORM(Object/Relation Mapping, 对象/关系 映射)技术, 解决了关系数据模型和面向对象模型之间的“阻抗不匹配”问题. 然而, 以上的解决方式只是针对某一方面的问题, 并没有站在软件开发框架的高度从整体上去解决问题.

在实际应用系统开发中, 对于三层架构的缺陷, 不同的开发者针对具体的项目采取了不同的解决办法, 其主要思想是对原有的三层架构进行更细的划分, 形成多层架构. 文献[5]中基于 .NET 平台将系统分为六层, 文献[6]中基于 JAVA 平台将系统分为五层. 这种多层

^① 基金项目:浙江省自然科学基金资助项目(LY12F02009);国家自然科学基金项目(61272007)

收稿时间:2012-10-19;收到修改稿时间:2012-11-29

划分的方法能够使得某个具体的项目具有更好的性能,但是存在不同项目间系统层次划分混乱的问题,不利于软件开发框架的重用,降低了软件开发的效率。

因此,本文在原有三层架构的基础上,通过利用 MVC 设计模式和 ORM 技术,提出表示层、业务逻辑层、数据访问层、数据持久层和数据库层的五层软件开发框架。这样,不仅解决了三层架构中存在的问题,而且为系统的开发提供了一个可以参考和重用的软件开发框架模型。

本文第二节对涉及到的相关技术进行介绍,第三节描述和分析五层软件开发框架,第四节用一个实例对该开发框架的实现进行说明,第五节进行总结。

2 相关技术介绍

本文涉及到的主要技术包括 MVC 设计模式和 ORM 技术,下面将对这两种技术进行简单介绍,本部分知识主要参考文献[7,8]。

2.1 MVC 设计模式

MVC 模式是“Model-View-Controller”的缩写,中文翻译为“模型-视图-控制器”。它把一个应用的输入、处理、输出流程按照 Model、View、Controller 的方式进行分离:视图(View)代表用户交互界面,用于数据的采集,以及接收用户的请求;模型(Model)用于业务流程/状态的处理以及业务规则的制定,是 MVC 最重要的核心;控制器(Controller)用于接收视图提交的请求,将模型与视图匹配在一起,共同完成用户的请求。MVC 使视图和业务逻辑相分离,降低了两者之间的耦合度,而且使得一个模型可以具有多个显示视图,提高了代码的重用性。

2.2 ORM 技术

ORM 是一种常用的持久化技术,其英文全称是 Object/Relation Mapping,中文译名为“对象/关系映射”,用来完成面向对象语言到关系数据库的映射,通过该映射用户可以像操作对象一样来操作关系型数据库。使用 ORM 后就不需要再与复杂的 SQL 语句打交道了,通过创建一个持久化类来映射数据库中的一个数据库表,其中持久化类的属性则映射到数据库表中的字段。当使用面向对象的方式来操作持久化对象时,ORM 框架能自动将这些操作转换成 SQL 语句,从而完成对数据库的操作。这时开发人员可以将重点放到业务逻辑上,而不必关心底层复杂的 SQL 语句。

3 五层框架设计

本小节首先分别介绍 MVC 设计模式和 ORM 技术是如何运用于三层架构的,然后在此基础上构建了五层框架并对其进行详细说明。

3.1 MVC 设计模式的运用

将 MVC 设计模式与三层架构相结合,即将 View 与表示层相对应, Model 与业务逻辑层相对应,并在表示层和业务逻辑层之间引入控制器(Controller)。它们之间的相互关系如图 2 所示:

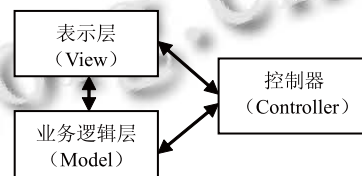


图 2 MVC 设计模式的运用

其工作原理为:表示层将用户请求提交给控制器,控制器对用户请求进行判断,然后调用业务逻辑层处理用户请求,根据处理结果调用表示层中的相应视图呈现给用户。

控制器的引入使表示层和业务逻辑层的代码在物理上相分离,从而降低了两者之间的耦合度,同时使得一个模型可以具有多个显示视图,提高了代码的重用性。

3.2 ORM 技术的运用

通过使用 ORM 技术,在原有三层架构的数据访问层和数据库层之间增加了数据持久层,如图 3 所示:

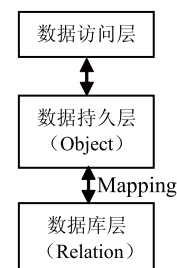


图 3 ORM 技术的运用

其工作原理为:在数据持久层创建持久化类来映射数据库中对应的数据库表,数据访问层通过对数据持久层中的实体类对象进行操作来实现对数据的处理,这样就避免了对关系数据库的直接操作。

数据持久层的增加,使得用户可以像操作对象一

样来操作关系型数据库,从而解决了关系数据模型和面向对象模型之间的“阻抗不匹配”问题.

3.3 构建五层框架

在原有三层架构的基础上,通过综合利用 MVC 设计模式和 ORM 技术,本文构建了包含表示层、业务逻辑层、数据访问层、数据持久层和数据库层的五层框架设计.其体系结构如图 4 所示:

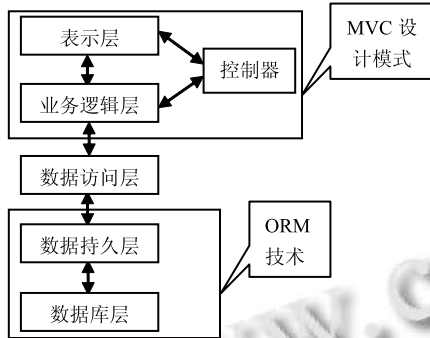


图 4 五层框架图

下面,对该五层开发框架中的每一部分进行详细说明:

第一层为表示层,位于最外层,离用户最近.该层主要负责和用户进行交互,并通过控制器调用业务逻辑层代码来实现判断等操作.该层与 MVC 设计模式中的 View 相对应.

第二层为业务逻辑层,该层在软件体系结构中的位置很关键,处于表示层和数据访问层之间,起到了数据交换过程中承上启下的作用.业务逻辑层是软件

体系结构中体现核心价值的部分,该层专注于实现业务逻辑功能,通过调用数据访问层代码来操作数据库.该层与 MVC 设计模式中的 Model 相对应.

第三层为数据访问层,该层专注于数据访问操作,包含了所有的操作数据的方法,如保存、删除、修改、查询数据.数据访问层不用涉及业务逻辑,可以使用 DAO 组件来完成.

第四层为数据持久层,该层专注于数据的持久化,即利用 ORM 技术,创建持久化类来映射数据库中对应的数据库表.该层一般通过 ORM 框架来实现.

第五层为数据库层,即为所使用的数据库.例如,SQL Server、Oracle、MySQL 等关系型数据库.

另外,还有一个就是控制器,位于表示层和业务逻辑层之间,用来接收表示层中表单所提交的数据,然后调用业务逻辑层并决定跳转页面.本文并没有把控制器作为软件开发框架中的一层,是因为它本身并不输出任何东西,也不进行任何业务逻辑的处理,只起到了一个中介的作用.控制器与 MVC 设计模式中的 Controller 相对应.

3.4 五层框架工作原理

由于层是一种弱耦合结构,层与层之间的依赖是向下的,下层对于上层而言是“无知”的,改变上层的设计对于其调用的下层而言没有任何影响.

五层框架各部分之间协同作用,共同来满足用户的需求.其工作原理如图 5 所示:

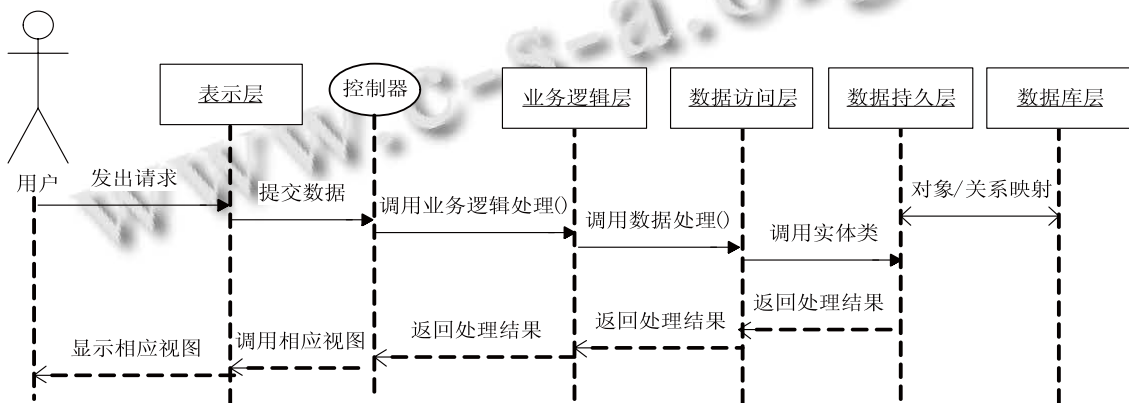


图 5 五层框架工作原理图

下面对该工作原理图进行详细说明:

(1) 用户向直接与其进行交互的表示层发出请求,作为整个事件的输入;

(2) 表示层将用户输入提交给控制器;

(3) 控制器对用户输入进行判断: 如果不需要进行业务逻辑处理,则直接调用相应视图,并在表示层

显示给用户；否则，调用业务逻辑层，对用户输入进行处理，并将处理结果返回给控制器，然后控制器根据处理结果调用相应视图，并在表示层显示给用户；

(4) 业务逻辑层在进行数据处理时并不直接操作数据库，而是调用数据访问层中相应的方法；

(5) 数据访问层对数据进行操作时也不直接操作数据库，而是对数据持久层中的持久化类对象进行操作；

(6) 在数据持久层创建持久化类对象，通过利用 ORM 技术，映射到关系型数据库中的数据库表。

4 案例分析

应用本文提出的五层框架，笔者开发了基于 Struts2 和 Hibernate 技术的“同城购物商城”系统。该系统是针对我国远程电子商务中存在的诚信、效率等问题，基于同城电子商务模式而开发的，主要包括商品信息管理、用户信息管理、订单管理和商品查询等模块。其中，Struts2 是第二代基于 MVC 设计模式的 Web 应用框架，Hibernate 是一个面向 Java 环境的 ORM 框架，它们都是目前系统开发中非常流行的轻量级框架。该系统的架构图如图 6 所示：

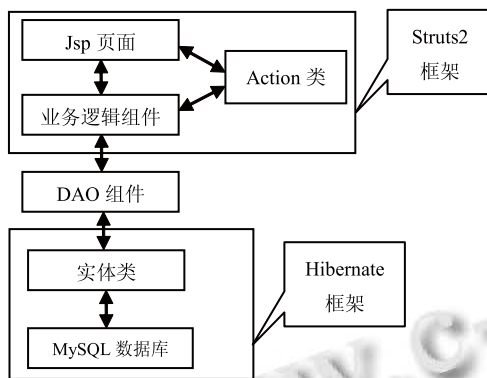


图 6 同城购物商城系统架构图

下面以商品信息管理模块中的商品录入操作为例，来说明如何使用 Struts2 和 Hibernate 框架来实现这种五层架构。由于业务逻辑层和数据访问层的开发不是本文说明的重点，所以这里就不再展开来描述。

(1) 数据库层开发

本系统采用 MySQL 作为数据库，创建一个用来存储商品信息的数据库表 goods，其中商品信息包括商品编号、商品名称、商品条码号、商品分类号、生产商、价格。

(2) 数据持久层开发

持久层的开发主要包括两部分，一个是 Hibernate 配置文件 hibernate.cfg.xml 的开发，一个是实体类和映射文件的开发。下面主要说明一下实体类和映射文件的开发：

前面创建了 goods 表，这时就添加一个实体类 Goods 来与其对应，Goods.java 文件部分代码如下：

```
public class Goods {
    private int goodsId;        //商品编号
    private String goodsName;   //商品名称
    private int goodsBarcode;   //商品条码号
    private String typeId;      //商品分类号
    private String producer;    //生产商
    private double goodsPrice; //价格
    //省略各属性的 setter 和 getter 方法
}
```

添加完实体类 Goods 后，需要添加一个映射文件 Goods.hbm.xml，从而完成 Goods 类的对象/关系映射。Goods.hbm.xml 文件部分代码如下：

```
<hibernate-mapping>
<class name="com.tcgw.po.Goods" table="goods">
<id name="goodsId">
<generator class="identity"></generator>
</id>
<property name="goodsName"></property>
<property name="goodsBarcode"></property>
<property name="typeId"></property>
<property name="producer"></property>
<property name="goodsPrice"></property>
</class>
</hibernate-mapping>
```

(3) 表示层开发

表示层用 Struts2 框架来实现。要完成商品的录入，首先需要添加一个商品录入表单，用来接收商品信息。

下面来添加一个商品录入表单页面 goodsInput.jsp，其部分代码如下：

```
<form action="goodsInput.action" method="post">
<table>
<tr><td colspan="2">录入商品</td></tr>
<tr><td colspan="2"><s:actionmessage/></td></tr>
<tr><td>商品名称:</td>
```

```

<td><input type="text" name="goodsName"/></td></tr>
<tr> <td>商品条码号:</td>
<td><input type="text" name="goodsBarcode"/></td></tr>
<tr><td>商品分类号:</td>
<td><input type="text" name="typeId"/></td></tr>
<tr><td colspan="2">
<input type="submit" value="录入" />
<input type="reset" value="重置" />
</td></tr>
</table>
</form>

```

(4) 控制器开发

控制器用 Struts2 框架来实现。首先需要添加商品录入控制器 GoodsInputAction, 该控制器用来接收表示层表单提交的商品信息, 并调用业务逻辑层来完成商品的录入。GoodsInputAction.java 文件代码如下:

```

Public class GoodsInputAction extends
ActionSupport {
private String goodsName;
private int goodsBarcode;
private String typeId;
private String producer;
private double goodsPrice;
//省略各属性的 setter 和 getter 方法
public String execute() throws Exception {
Goods goods = new Goods();
goods.setGoodsName(goodsName);
goods.setGoodsBarcode (goodsBarcode);
goods.setGoodsPrice(goodsPrice);
GoodsService goodsService =
GoodsServiceFactory.getGoodsServiceInstance();
if(goodsService.inputGoods(goods)) {
this.addActionMessage("录入商品成功!");
return SUCCESS;
}
else {
this.addActionMessage("录入商品失败!");
return INPUT;
}
}
}

```

}

}

创建完 GoodsInputAction, 还需要在 struts.xml 文件中配置该 Action, 配置代码如下:

```

<action name="goodsInput"
class="com.tcgw.action.GoodsInputAction">
<result
name="success">/goodsInputSuccess.jsp</result>
<result name="input">/goodsInput.jsp</result>
</action>

```

该案例是利用 Struts2 和 Hibernate 技术来实现的, 由于本文所提出的五层框架是一种可重用的模型, 所以也可以用其他的方式来实现。

5 总结

针对三层架构设计中存在的问题, 本文提出了一种可重用的五层开发框架设计模型, 并用一个案例的实现对该模型进行了说明。实践证明, 该五层开发框架不仅解决了三层架构存在的缺陷, 而且能够提高开发效率, 增强系统的可维护性和可扩展性。当然, 该五层开发框架也存在着一定的不足, 其执行速度是我们下一步所要研究的方向。

参考文献

- 1 曾长军.用 ADO.NET 实现三层结构.信息通信,2011(5).
- 2 徐朝辉,樊银亭.MVC 设计模式在.NET 三层结构开发中的研究和实现.北京电子科技学院学报,2007(2).
- 3 张皓.Java 中的对象/关系映射[硕士学位论文].武汉:武汉大学, 2005.
- 4 Bernd Bruegge,Allen H.Dutoit.Object-Oriented Software Engineering,2011.
- 5 曾志刚.基于.NET 多层体系结构的应用开发框架的设计、实现及应用[硕士学位论文].合肥:国防科学技术大学,2007.
- 6 盛青.基于 J2EE 架构的软件开发方法研究及实例[硕士学位论文].长春:吉林大学,2004.
- 7 任中方,张华,闫明松,陈世福.MVC 模式研究的综述.计算机应用研究,2004(10).
- 8 董立波,黄玉霞,杨珍.ORM 技术研究.软件导刊,2007(09).
- 9 古乐声,卜烜等著.Java Web 程序设计与项目实践,2011.