基于 MVC 模式的 Web OA 系统的设计与实现

张晓丽[1],路 杨[2]

(1. 河南大学 计算机与信息工程学院,河南 开封 475001;
2. 河南大学 计算中心,河南 开封 475001)

摘 要:为了方便企业的管理和提高协同办公效率,越来越多的企业采用了 Web OA 系统。针对传统设计模式开发的 Web OA 系统存在维护成本高、难以扩展等缺陷,文中提出一种基于 MVC 设计模式的系统开发方案。首先详细介绍了 MVC 模式的设计思想和 Zend Framework 的工作流程及原理,然后阐述了系统设计与实现的整个过程。该设计方案充分利用了 MVC 的结构和特点并已成功应用于某办公自动化系统的开发。实践表明,基于 MVC 模式开发的 OA 系统运行稳定,具有较好的灵活性和高效性,并且大大减少了系统的开发周期,达到了预期的效果。

关键词:MVC;OA 系统;设计模式;Zend Framework;SQL Server 2008

中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2012)08-0063-04

Design and Implementation of Web Office Automation System Based on MVC Pattern

ZHANG Xiao-li[1], LU Yang[2]

(1. School of Computer and Information Engineering, Henan University, Kaifeng 475001, China;
2. Computing Center, Henan University, Kaifeng 475001, China)

Abstract: In order to facilitate the management of enterprises and improve collaboration office efficiency, more and more enterprises adopt the Web OA system. The present Web OA systems which have been implemented by the traditional design model have some deficiencies such as low maintenance cost, poor scalability. It proposed a development plan based on MVC design pattern. By using the structure and characteristics of MVC, the plan has been successfully applied to the implementation of OA system. Practice shows that the Web OA system based on MVC pattern has been running steadily, and has remarkable flexibility and high efficiency, and also shortented the development cycle of system, basically reached the predicted effects.

Key words: MVC; OA system; design pattern; Zend Framework; SQL Server 2008

0 引 言

办公自动化(OA)系统主要是指利用电脑设备、网络及信息资源构建一个办公信息平台,能够实现办公轻松有序的管理,从而大幅度地提高办公效率和质量。因此,OA 系统在一个企业的运作中发挥着重要的作用。传统的设计模式开发流程是直接调用数据库中的数据并用 HTML 显示,应用分层不彻底,结构不清晰。MVC 模式的三层架构设计能很好地解决这一问题,因此,为了满足企业的需求而设计与实现了一套 Web OA 系统,该系统基于 MVC 模式的设计思想,结合 Zend Framework 而开发,相比采用传统的设计模式开发出来的 Web OA 系统具有安全可靠、可维护性高、易扩展等特点。

收稿日期:2011-12-30;**修回日期**:2012-04-02

基金项目:国家自然科学基金项目(61004006)

作者简介:张晓丽(1987-),女,河南商丘人,硕士,研究方向为模式识别;路 杨,博士,副教授,研究方向为模式识别。

1 系统开发技术

1.1 MVC 设计模式

MVC 是一个目前在 Web 应用中比较流行的设计模式,MVC 指的就是 Model(模型)-View(视图)-Controller(控制器),它强制性的把应用程序分为模型、视图、控制器这三个部分,避免模型和视图的混合,从而使一个模型可以对应多个视图[1]。如果模型发生改变,控制器就会将变化告知视图更新页面。MVC 的架构图[2]如图 1 所示。

视图(View):视图是与用户交互的界面。在 MVC 设计模式中它的任务主要用来显示模型的状态,将用户的请求传给控制器并接受数据更新请求。

控制器(Controller):Controller 是 Model 与 View 之间沟通的桥梁,是 MVC 设计模式中最核心的部分。控

制器本身不输出任何东西和做任何处理。它只是接收用户的请求并且调用合适的模型来响应用户的请求，然后根据模型与数据库交互取得的结果选择合适的 View 返回给客户端[3]。

模型(Model):模型的作用主要是与数据库交互，它独立于外在显示内容和形式,是软件所处理的问题逻辑的内在抽象，它封装了问题的核心数据、逻辑和功能的计算关系,独立于具体的界面表达和 I/O 操作[4]。如果数据库发生改变,那么它会通知视图(View)更新。被模型返回的数据不带任何数据格式，所以一个模型能为多个视图提供数据,避免了代码冗余并大大减轻开发者的工作量。

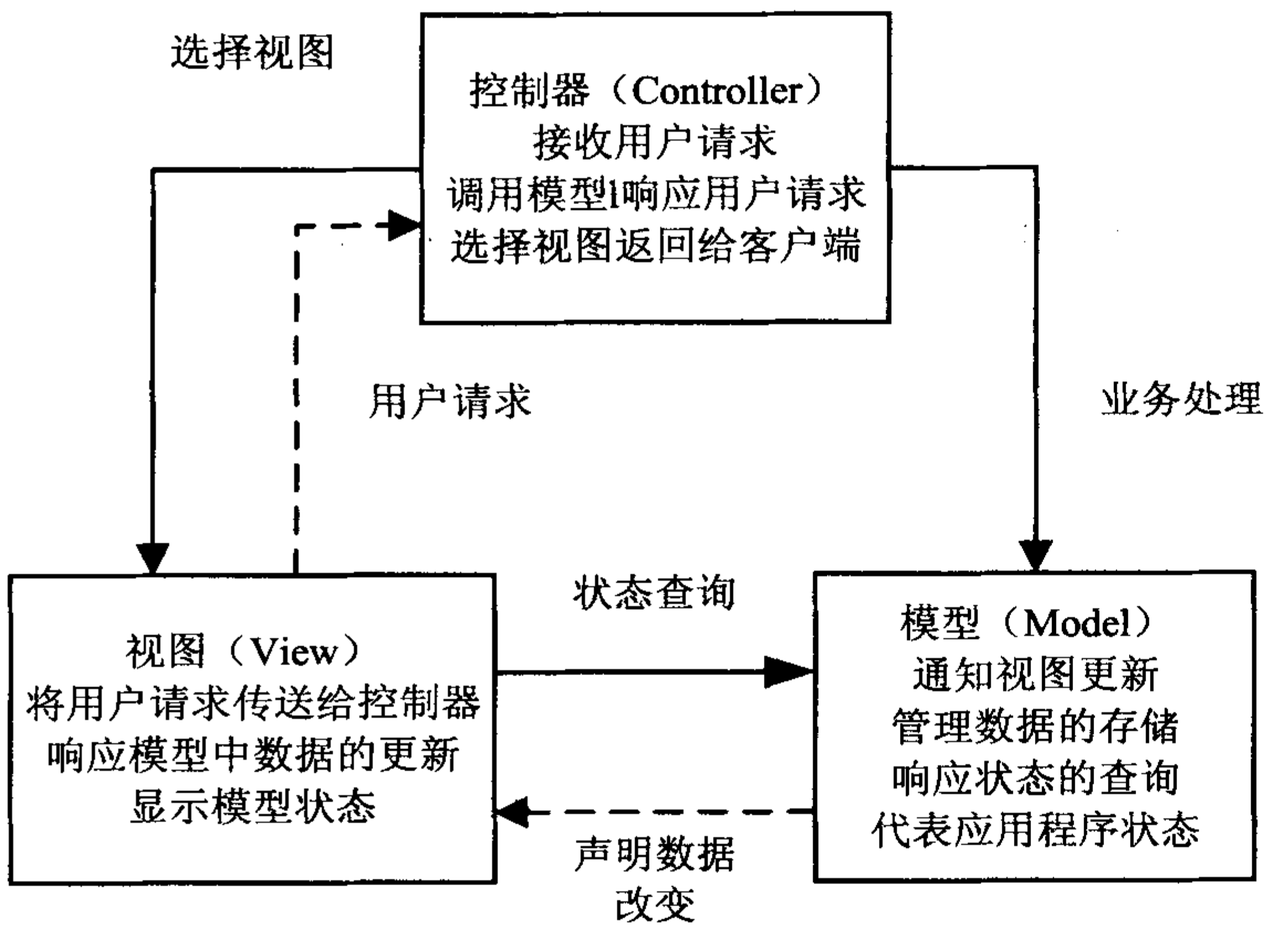

图 1 MVC 架构图

1.2 Zend Framework

Zend Framework 是一个基于 PHP5 技术的稳定的软件开发框架。它的内部主要有五大组件组成,分别是 MVC 组件、核心组件、数据操作组件、服务类组件、国际化组件[5]。其中每个组件都是相互独立的。这样的松耦合结构可以让开发者独立使用组件[6]。Zend Framework 提供的 Zend_Controller 组件是 MVC 组件的核心,同时也是实现 MVC 设计模式最核心最重要的部分。Zend_Controller 组件中的前端类(Zend_Controller_Front)实现了前端控制器(Front Controller)设计模式。在该种设计模式下所有的请求都通过 Front Controller 并分发(Dispatch)到不同的控制器来处理[7]。基于 Zend Framework 的响应 Request(请求)的大致流程[8]如下:首先路由器(Router)处理浏览器发出的 Request Object 并解析出需要调用的模块名、控制器名、控制器动作名、参数。分发事件被触发,在分发过程中从 Request Object 中提取并调用合适的控制器和控制器类文件中相应的 Action 方法。控制器接收请求并解析,然后将请求的参数传给 Model,Model 接收请求并通过 Zend_Db_Table 与数据库交互,判断数据是否和数据库中的数据一致。分发事件结束后前端控制器获取 Response Object 并将其返回给客户端。

2 MVC 模式下 Web OA 系统的设计

2.1 数据库设计

在 Web 办公自动化系统中,数据库采用 MS SQL Server 2008,MS SQL Server 2008 是一个可信任的、高效的、智能的数据库平台[9]。系统引入的 Zend Framework 开发框架是一个模块化和灵活的框架结构,具有丰富的组件,并且内部的各个组件独立。其中 Zend Framework 提供了 Zend_Db 数据库操作组件,组件中的 Zend_Db_Table 类通过 Zend_Db_Adapter 类连接到数据库[10],并对数据库中的表进行操作。在 Zend Framework 中构建 Web 应用程序时,需要创建配置文件。与数据库连接必须进行数据源配置,在 appplication. ini 配置文件中进行配置,与 MS SQL Server 2008 进行连接,配置代码如下:

```
resources. db. adapter = "sqlsrv"
resources. db. params. host = " . \sql2008"
resources. db. params. username = "jszx"
resources. db. params. password = " * * * "
resources. db. params. dbname = "yyoa"
resources. db. params. charset = "utf-8"
```

根据系统的 E-R 图来设计数据库表,同时对于数据库中的表结构和字段的设计,命名规则采用英文单词缩写的方法。对于每张数据库表的索引,建立一个聚集索引,部分操作比较频繁根据需要建立非聚集索引,但不超过三个。索引的命名规则统一使用:聚集索引用 PIX 加下划线加表名称,非聚集索引用 IX 加表名称加序号的方式。同时为了提高数据的执行速率和程序的安全性,系统采用存储过程,存储过程的使用增强了 SQL 语言的灵活性,使系统具有良好的安全机制。另外在数据库中创建了视图和触发器以提高数据的安全性和实现数据库表之间的级联更新。

2.2 系统设计

在系统的模块划分上,Web OA 系统设计主要包括我的公文包、工作审批等功能模块,每个模块下又细分了好几个子功能。基于 MVC 模式的 Web OA 系统的功能模块图如图 2 所示。

基于 MVC 设计模式,以 Web OA 系统中的新闻管理模块为例来阐述系统是如何设计的,系统中其他模块都采用同样的设计方案,以确保系统的层次分明、结构清晰、易于维护。新闻管理模块包括查询、添加、修

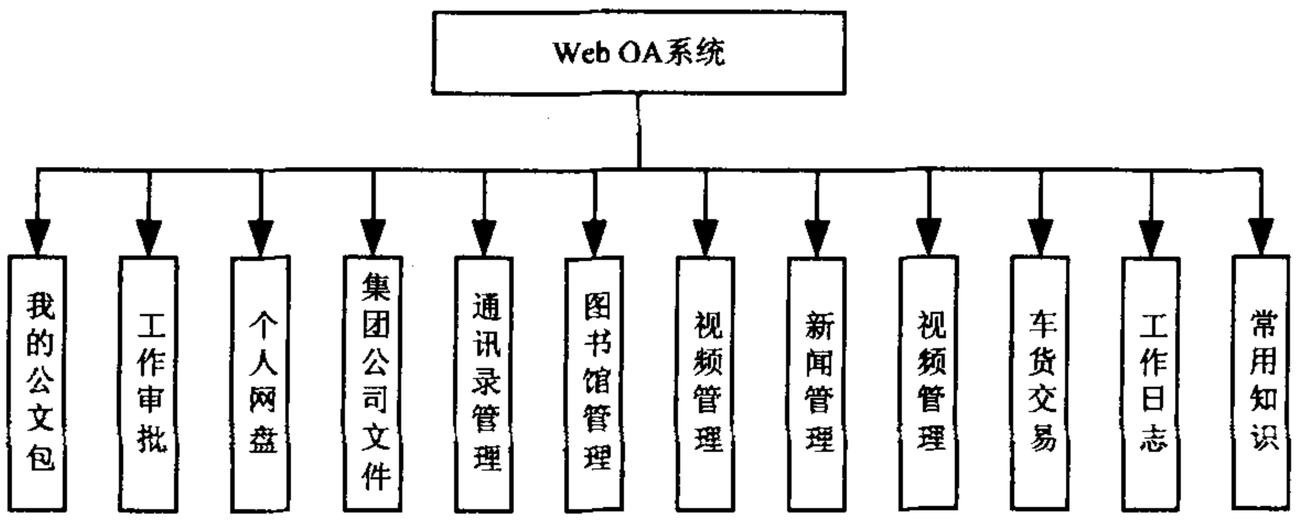

图 2　Web OA 系统功能模块图

改、删除新闻等功能。基于 MVC 设计模式的程序结构设计如表 1 所示。在新闻管理模块中定义了两个视图文件分别为添加(addnews. phtml)和显示新闻列表(shownews. phtml)视图界面。在控制器类文件 NewsController. php 中定义了添加 addnewsAction()、修改 updatenewsAction()、删除 delnewsAction()、显示新闻列表 shownewsAction()等 Action()方法。模型文件 NewsManage. php 中定义了与数据库交互的添加 add($params)、修改 update($params)、删除 del($id)、查询 getnews($params)等方法。以发布新闻为例,当用户发出一个发布新闻请求时即浏览器显示 addnews. phtml 页面,用户提交表单后会触发 NewsController. php 里的 addnewsAction()方法,而后调用 Model 里相应的 add($params)方法,Model 调用数据库中的存储过程,并进行新闻的添加操作。修改、删除新闻的操作同添加操作。

表 1　基于 MVC 模式的程序结构设计

视图(View)	控制器(Controller)	模型(Model)
addnews. phtml shownews. phtml	NewsController. php	NewsManage. php

3　MVC 模式在 Web OA 系统中的实现

3.1　系统开发平台及环境

系统引入基于 PHP 语言的集成开发环境-Zend Studio,它具有良好的数据库连通性并加速开发周期。采用 PHP5+Apache 服务器+ MS SQL Server 2008 组合对系统进行开发。其中 PHP5 是一种功能强大且丰富的完全面向对象的编程语言,目前广泛应用于 Web 程序的开发。Apache 服务器是当今热门的 Web 服务器之一,尽管它的配置比较复杂,但因其具有良好的跨平台性、可移植性和稳定性,受到许多开发人员的青睐。

采用 Ajax 和 jQuery 等相关技术轻松实现了网页的局部刷新,并且提高了代码的简洁性和可重用性[11]。

3.2　系统实现

下面以 OA 系统中用户登录的功能实现为例来说明系统是如何实现 MVC 设计模式的。当用户向服务器端发送一个登录请求,将请求的参数即用户名和密码传送给控制器,控制器接收请求后,将参数传送给模型 Model,Model 查询数据库中的数据,若数据库中的数据和参数一致,则登录成功并转向 main. phtml 页面。

3.2.1　视图的实现

视图文件存放 \ application \ views \ scripts 目录下,用户登录的功能实现需要两个视图文件,分别为登录页面 login. phtml 和登录成功后跳转到的页面 main. phtml。

同时在登录界面 login. phtml 中使用 JavaScript 进行输入数据的检查。在 js 里数据编码统一采用 UTF-8[12],所有的 PHP 文件以及视图文件也要采用此类编码以防止乱码问题带来的不便。用户进入登录界面 login. phtml,输入用户名和密码,并将用户名和密码传送给控制器,若用户名和密码输入正确则跳转到 main. phtml 页面。否则提示错误,并重新输入。在视图文件登录页面 login. phtml 中设置表单:

```
<form id = " form1 "  name = " form1 "  method = " post "  action
= "" >
<tr><td width = "49"  height = "24"  align = " left" >用户名:</
td>
<tdwidth = "157" align = " left" >
<input name = " user"  type = " text"  id = " user" >
</td></tr>
<tr><td valign = " top"  align = " right"  height = "60" >密码:</
td>
<tdalign = " left" ><inputname = " pwd" type = " password" id = "
pwd" >
</td></tr>
<tr><td>
<input type = " submit"  name = " submit"  id = " submit"  value
= "登录"  class = " but"  />
</td></tr>
</form>
```

3.2.2　控制器的实现

在 Zend Framework 框架中添加控制器 loginController,这个类继承抽象类 Zend_Controller_Action。

打开文件 \ application \ controllers \ loginController. php,输入以下代码:

```
<? php
require_once 'Zend/Controller/Action. php';
classloginController extends Zend_Controller_Action
{
public functionloginAction()//登陆
{ $this->_helper->viewRenderer>
setNoRender(true);//不显示 View
$username = $_POST['user']; //用户名 $password = md5
```

```
($_POST['pwd']); //密码
    }
    }
    ?>
```

在loginController类里定义一个Action方法loginAction(),那么控制器会指定这个Action方法,选择同名的View文件(login.phtml)输出。在loginAction()里采用$_POST[]方法获取来自method="post"的表单中提交的用户名和密码,并且调用Model里的方法loginManage($params),若数据正确,则通过$this->_redirect('/Index/main')这句代码转向main.phtml页面,否则提示登录失败。

3.2.3 模型的实现

在MVC设计模式中,Model负责对数据库的操作,并处理控制器Controller传来的数据请求以及当数据发生改变时将这些变化告知视图,然后视图会做出相应的调整。在基于Zend Framework框架的Web OA系统中,模型类文件存放在\application\models目录下。Model获取从控制器Controller传来的用户名和密码等数据,在模型文件中定义一个方法,此方法调用存储过程P_System_login,获取数据库中的数据经控制器的调用将结果返回给相关的视图(View)输出给客户端。主要实现代码如下:

```
$db=get_db();
$params = array(
array(&$username, SQLSRV_PARAM_IN),
array(&$password, SQLSRV_PARAM_IN),
)
$results=$db->queryporc('P_System_login',
$params); //调用存储过程
return $results;
```

4 结束语

文中对于MVC设计模式和Zend Framework以及系统的设计与实现进行了详细的阐述。MVC设计模式不仅能够适宜的把整体分成局部,同时利用三层架构的思想以及低耦合性的结构使得开发人员各司其职,而且简化了应用程序的复杂性,提高了系统的可扩展性。Zend Framework的模块化的结构设计和丰富的组件使得程序更易扩展和灵活,系统也更加稳定。实践表明,采用MVC模式开发出来的系统结构更加清晰,性能更加稳定,并且易于管理。

参考文献:

[1] 王映辉,王英杰,王彦君,等.基于MVC的软件界面体系结构研究与实现[J].计算机应用研究,2004(9):188-199.

[2] 孙卫琴.精通Struts:基于MVC的JavaWeb设计与开发[M].北京:电子工业出版社,2004.

[3] 刘春花,王忠民.基于MVC模式的远程评议系统的设计与实现[J].计算机工程与设计,2008,29(13):3648.

[4] 王晓楠. MVC的设计和实现[J].计算机系统应用,2004(3):56-58.

[5] 陈名箴.基于Zend框架的项目管理系统设计与实现[D].杭州:浙江大学,2010.

[6] Evans C. PHP Architects Guide to Programming with Zend Framework[M].[s.L.]:Marco Tabini & Associates,2008.

[7] 陈营辉,赵 伟,赵海波,等. Zend Framework技术大全[M].北京:化学工业出版社,2010.

[8] Rob A,Nick L,Steven B. Zend Framework in Action[M]. United States of America: Manning Publications,2008.

[9] 赵新燕.山东省青干院学工信息管理系统的设计与实现[D].济南:山东大学,2010.

[10] 张朝阳,熊淑华,衡 丽.基于Zend Framework的网站设计与实现[J].计算机技术与发展,2011,21(11):199-200.

[11] Flanagan D. jQuery Pocket Reference[M].[s.L.]:O'Reilly,2010.

[12] 王 锋,魏晓丽,江开耀.基于XML的C#多语言界面实现[J].计算机工程与设计,2008,29(15):4073-4074.

(上接第62页)

edge and Data Engineering,2007,19(1):1-16.

[7] 戴 颖,李兴国,赵启飞.一种相似重复记录检测算法的改进研究[J].计算机技术与发展,2010,20(7):13-16.

[8] 韩京宇,徐立臻,董逸生.一种大数据量的相似记录检测方法[J].计算机研究与发展,2005,42(12):206-212.

[9] 庞雄文,姚占林,李拥军.大数据量的高效重复记录检测方法[J].华中科技大学学报,2010,38(2):8-11.

[10] 鲁均云,李星毅,施化吉,等.基于内码序值聚类的相似重复记录检测方法[J].计算机应用研究,2010,27(3):874-878.

[11] 孟祥逢,鲁汉榕,郭 玲.基于遗传神经网络的相似重复记录检测方法[J].计算机工程与设计,2010,31(7):1550-1553.

[12] 陈锦禾,沈 洁.基于信息熵的主动学习半监督分类研究[J].计算机技术与发展,2010,20(2):110-113.

[13] 陈 伟,丁秋林.一种XML相似重复数据的清理方法研究[J].北京航空航天大学学报,2004,30(9):835-838.

[14] Keulen M, Keijzer A. Qualitative effects of knowledge rules and user feedback in probabilistic data integration[J]. VLDB journal,2009,18(5):1191-1217.

[15] Data Quality:Concepts,Methodologies and Techniques(Data-centric Systems and Applications)[M].[s.l.]:[s.n.],2006.