

文章编号: 2096-1472(2016)-03-45-03

# 基于ThinkPHP模型的敏捷WEB开发的应用与实践

张文胜, 陈宏

(西安欧亚学院, 陕西 西安 710065)

**摘要:**近年来,随着网络的普及和WEB应用需求的增加,敏捷的WEB开发的应用成为研究领域的热点之一。本文以ThinkPHP框架技术为基础,使用敏捷的WEB开发流程和方法,完成了新闻信息发布系统的整体设计和实现。最后,探讨了未来的研究趋势。

**关键词:** ThinkPHP; 敏捷开发; WEB开发

**中图分类号:** TP311 **文献标识码:** A

## The Research and Application of Agile WEB Development Based on ThinkPHP Mode

ZHANG Wensheng, CHEN Hong

(Xi'an Eurasia University, Xi'an 710065, China)

**Abstract:** In recent years, with the increasing popularity of the network and WEB application requirements, agile development of the WEB application has become one of the hot research field. This paper is based on the ThinkPHP framework, using the WEB development process and agile methods, completed the design and implementation of information publishing system. Finally, future research trends are discussed.

**Keywords:** ThinkPHP; agile development; WEB development

### 1 引言(Introduction)

目前,随着网络的普及,网站和Web应用程序开发的需求在不断扩大,用户体验要求也越来越高。如果大家都是需要自我开发一套Web应用系统,则存在成本高、没有技术力量,维护不方便等问题。所以,如果可以开发一套Web应用系统模型,将一次性源码交付客户,让客户拥有自己的Web应用平台,则操作相对简单,在功能上也可自主灵活修改,这样就能很好的解决用户所面临的问题<sup>[1]</sup>。

MVC模式(Model-View-Controller)是一种应用广泛的开发模式,能够解决团队开发之间分工协作、协同配合的问题,最优化系统各部件的开发效率。MVC模式将管理系统分为三大部件,分别为模型、视图和控制器。这三大部件各自分离,但又相互依存,最终形成了一个容易维护、容易扩展、高效运行的管理系统平台。对于后台程序员,借助于MVC模式就可以更加专注于业务流程和功能的实现,而不需要太多地涉及页面与前端。这种分工协作的最终目的是提高开发效率及项目质量<sup>[2]</sup>。

ThinkPHP框架是为了敏捷WEB应用开发和简化开发者应用而设计的,是基于MVC结构的开源PHP框架,ThinkPHP一直秉承实用简洁的设计思想,发挥开源和社区团

队的优势,不断优化和改进,在易用性、扩展性和性能方面均有较好表现,已经成长为国内领先和具有影响力的敏捷的WEB应用开发框架,已被大量应用于商业以及门户级网站的开发<sup>[3]</sup>。

本文以新闻信息发布系统为例,给出了基于ThinkPHP模型的敏捷WEB开发的设计思想及其开发流程,并使之在目录结构和代码上标准化和规范化,提高模块的复用性,使开发者专注于业务流程的设计,减轻在编写界面、组织代码等重复性的工作上的负担。

### 2 ThinkPHP MVC设计思想和开发流程(ThinkPHP MVC design and development process)

#### 2.1 ThinkPHP MVC设计思想

ThinkPHP是基于MVC设计模式的,且均支持多层(multi-Layer)设计。ThinkPHP中的MVC分层大致体现为:

**模型(Model)层:** 默认的模型层由Model类构成,但是随着项目的增大和业务体系的复杂化,单一的模型层很难解决要求,从3.1版本开始推出了多层Model的支持,设计思路很简单,不同的模型层仍然都继承自系统的Model类。

**控制器(Controller)层:** ThinkPHP的控制器层由核心控制器和业务控制器组成,核心控制器由系统内部的App类

完成，负责应用(包括模块和操作)的调度控制，包括HTTP请求拦截和转发、加载配置等，业务控制器则由用户定义的Action类完成。3.1版本开始增加了多层业务控制器的支持。

视图(View)层：由模板和模板引擎组成，在模板中可以直接使用PHP代码，模板引擎的设计会在后面讲述，通过驱动也可以支持其他第三方的模板引擎。

在MVC三层中，ThinkPHP并不依赖M或者V，甚至可以只有C或者只有V，这个在ThinkPHP的设计里面是一个很重要的用户体验设计，用户只需要定义视图，在没有C的情况下也能自动识别<sup>[4]</sup>。

### 2.2 ThinkPHP开发流程

使用ThinkPHP创建应用的一般开发流程是<sup>[5]</sup>：

- (1)创建数据库和数据表(没有数据库操作可略过)。
- (2)项目命名并创建项目入口文件。
- (3)完成项目配置。
- (4)创建控制器类。
- (5)创建模型类，如果只是简单的模型类可以不必创建。
- (6)创建模板文件。
- (7)运行和调试。

### 2.3 ThinkPHP目录结构各个文件夹的含义<sup>[6]</sup>

ThinkPHP的目录结构非常清晰和容易部署，目录结构的内容和说明如表1所示。

表1 目录结构的内容和说明

Tab.1 The directory structure of the content and description

目录/文件	说明
ThinkPHP.php	框架入口文件
Common	公共文件目录
Conf	项目配置目录
Lang	语言包目录(非必须)
Lib	应用类库目录，通常包括Action和Model子目录
Tpl	模板文件目录，支持模板主题
Extend	框架扩展目录

## 3 基于ThinkPHP模型的敏捷WEB开发的应用(The application of ThinkPHP model of agile development based on WEB)

### 3.1 应用程序的模型描述

首先，设计一个模型(M)，模型的定义由Model类完成，该类完成了基本的CURD、ActiveRecord模式、连贯操作和统计查询等。Model类位于项目目录下面的/ Lib/ Model目录。

其次，设计控制器(C)，应用控制器(核心控制器)和Action控制器都承担了控制器的角色。核心控制器由系统内部的App类完成，负责应用(包括模块和操作)的调度控制，包括HTTP请求拦截和转发、加载配置等，业务控制器则由用户定义的Action类或者其他控制器类完成。Action控制器位于项目目录下面的/ Lib/ Action目录。

最后，完成视图设计(V)，视图主要是指模板文件和模板引擎，模板的实现和框架无关，可以独立制作和预览。模板位于项目目录下面的/ Tpl目录。

模型调用关系描述如图1所示。

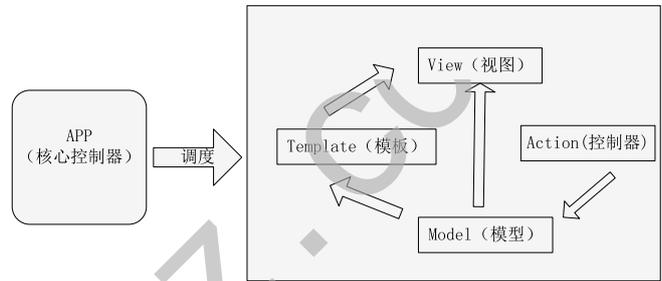


图1 模型调用关系描述

Fig.1 Call relation description model

### 3.2 构建应用程序的结构

在使用应用架构创建应用时，使用统一的命名规范和目录结构，规定所有类文件都放至Lib目录下。在该目录下，可根据分别建立Action、Model、Behavior和Widget等四个子目录。项目文件夹的命名约定见2.3节，这样就将对视图界面、交互逻辑和数据的处理清晰地分开，便于对应用进行管理和优化。目录结构如图2所示。

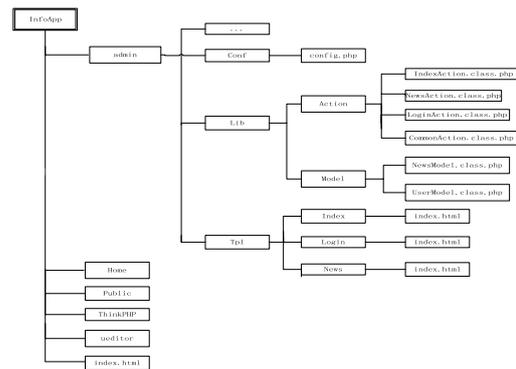


图2 应用程序目录结构

Fig.2 The directory structure of the application

### 3.3 MVC各个模块的实现

视图由View类和模板文件两部分组成。Action控制器直接和View视图类交互，View类把控制器传来的变量展现出来，而模板文件完成呈现的形式。

为了对模板文件更加有效地管理，Think PHP对模板

文件进行目录划分，默认模板文件定义规则是：模板目录\分组名\模板主题\分组名\模块名\操作名+模板后缀。项目模板目录是admin\Tpl，对于后台管理模块来说，首先需要新建Admin文件夹(模块名)，然后在该目录下新建三个文件夹Index、Login和News，分别对应显示查看管理员主页面、登录页面和添加新闻页面的内容。当用户调用Action文件夹各个类的对应的相关方法时，系统将会调用相应模板文件夹下的静态网页index.html，并将动态内容输出到模板中指定的标记处，最后使用display方法输出相应的模板。

Model模型的定义是由Model类来完成，ThinkPHP对数据库的表名和模型类的命名遵循一定的规范。首先数据库的表名和字段全部采用小写形式，模型类的命名规则是除去表前缀的数据表名称，并且首字母大写，然后加上模型类的后缀定义。

系统在后台登录和添加新闻时，填写完表单后，需要校验表单各项内容，为实现上述功能需求，就需要自定义对应的模型类UserModel.class.php和NewsModel.class.php。并使用D方法实现自动验证与自动填充。

Action控制器对应的文件位于项目目录admin\Lib\Action下。框架会根据当前的URL来分析要执行的模块和操作，即模型和视图只有在定义了相应的控制器后才能真正被使用。

在管理员管理模块中，需要自定义IndexAction、LoginAction和NewsAction控制器类，定义规则和模型定义相同，需要注意的是控制器名称要与视图模板文件夹名称保持一致。

根据项目的功能需求控制器类IndexAction中需要定义news、edit、quit和filter等方法，NewsAction中需要定义add、delete、edit、quit和update等方法，LoginAction中需要定义index、login和checklen等方法。

至此，基于ThinkPHP模型框架的新闻发布管理系统中后台管理模块的基本功能就搭建完成，前台功能的搭建过程也大致类似。纵观整个开发过程中，各个模块定位清晰，分工明确，调度自然流畅，复用性和扩展好，特别对于美工基础比较薄弱的开发者，可以借鉴现成的前端模板资源，并集成开源的百度UEditor编辑器插件，完全可以满足用户要求。

#### 4 运行结果及分析(The operation results and analysis)

本文以新闻信息管理系统为例进行分析说明，基于ThinkPHP模型框架实现了新闻发布管理信息Web前端展示和后台管理操作，系统前端展示如图3所示。



图3 应用程序前端主页

Fig.3 The tip home page of the application

开发人员可套用合适模板，使用Volist标签主在模板中循环输出新闻标题、内容和发布时间，主要代码如下：

```
<volist name="news_list" id="vo">
  <div class="col last">
    <h2>{$vo['subject']}</h2>
    <p>{$vo['message']}stripslashes</p>
    <a class="more" href="#">{$vo['createti
me']}</a>
  </div>
</volist>
```

后台管理登录如图4所示，后台管理主页如图5所示，新闻添加页面如图6所示。

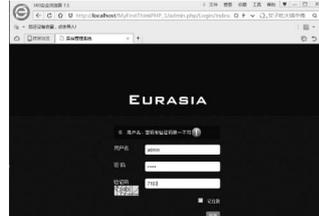


图4 后台管理登录

Fig.4 Admin login



图5 后台管理主页

Fig.5 Admin home page



图6 新闻添加页面

Fig.6 Add news page

在后应用中,登录信息需要格式校验,校验完成后,调用后台数据库验证用户名和密码。成功后,进入后台管理主页,后台会罗列每条已经发布新闻的基本信息,管理员可以对每条信息进行修改和删除。处理事件被绑定到每行新闻的操作图标上,然后通过单击进行修改和删除。新闻添加时,进入新闻添加页面,添加新闻标题和内容即可,本系统集成开源的百度UEditor编辑器插件,使用在线编辑功能,方便对新闻内容的添加和修改。

ThinkPHP框架以标准的敏捷的方式开发Web项目,提高了开发速度,降低了开发难度,使整个项目的模块划分和代码编写更加规范,前端页面的展示借用模板更加容易实现和扩展,更适合多人协同工作,进行分工编写复杂的界面和后台。

本系统使用ThinkPHP开发框架来部署Web应用,快速、简单、门槛低,能够短时间内开发出不错的有效率,安全性能高的网站。特别是初学者,能够在此基础上进一步的扩展,添加诸如自动填充实现无限级分类功能,这样对于了解PHP语法并且要快速开发的人来说,是一个十分便捷的工具。

## 5 结论(Conclusion)

本文在ThinkPHP开发框架的基础上,通过敏捷的开发手段,对其中所使用的关键技术均做了详细介绍,并通过实际应用给出了实现方法,系统经过测试,性能稳定、交互界面友好,安全性能良好,基本达到了预期的设计要求,具备较强的实用和实践参考价值。

随着Web技术的发展,工程化的特点日趋明显,本文在

ThinkPHP模型的敏捷Web技术的规范化、模块化、流程化上做了一些探索,在今后的工作中,会在提供标准服务的基础上,让开发者可以根据自己的产品和业务特点,通过在线配置和上传代码的功能来扩展自定义的功能,满足个性化需求。

## 参考文献(References)

- [1] 张文胜,陈宏.基于MVCS模式的组件化手机Web前端的研究与应用[J].电子设计工程,2015,11(23):19-22.
- [2] 张文胜,陈宏.企业级组态式Web前端设计方法研究与应用[J].电子设计工程,2014,22(17):162-164.
- [3] ThinkPHP3.2.3完全开发手册[Online]Available: <http://www.kancloud.cn/manual/thinkphp/1679>,基础部分,2015.
- [4] ThinkPHP完全开发手册[Online]Available: [http://doc.thinkphp.cn/manual/development\\_process.html](http://doc.thinkphp.cn/manual/development_process.html),MVC分层,2014.
- [5] ThinkPHP完全开发手册[Online]Available: [http://doc.thinkphp.cn/manual/development\\_process.html](http://doc.thinkphp.cn/manual/development_process.html),开发流程,2014.
- [6] 王俊芳,李隐峰,王池.基于MVC模式的ThinkPHP框架研究[J].电子科技,2014,27(4):151-53.

## 作者简介:

张文胜(1967-),男,硕士,工程师,讲师.研究领域:Web前端架构,软件架构设计,软件工程.

陈宏(1977-),女,硕士,讲师.研究领域:软件测试,软件架构设计,软件工程.

(上接第57页)

## 6 结论(Conclusion)

智能家居系统是嵌入式远程控制系统的-一个应用例子,由于控制对象多,单个智能控制系统无法完成控制任务,需要构成一个由多种设备、多种平台的综合系统。本方案内部采用总线协议,具有组网方便、性能稳定、成本低廉的特点;外部采用电话与TCP/IP协议<sup>[6]</sup>,具有远程控制能力、扩展性强的特点。嵌入式与非嵌入式的连接再加载数据库又能组成更大的智能系统。

## 参考文献(References)

- [1] 王平,等.家用网络化智控系统的研究开发[J].计算机应用研究,2004,21:221-222.
- [2] 王平,等.家居网络智控系统中PSTN远程通信模块的设计[J].福建师大学报,2007,23(4):61-63.

[3] 张培仁.基于C语言编程 MCS-51单片机原理与应用[M].北京:清华大学出版社,2003.

[4] W.Richard Stevens.陆雪莹,译.TCP/IP详解,卷2:实现[M].北京:机械工业出版社,2000.

[5] 夏涛,陶洋.基于四大模块的智能家居的研究与设计[J].工程地质计算机应用,2007,48(4):23-26.

[6] 郑楠,吕幼新.Winsock技术在校园消防远程监控系统中的应用[J].信息与电子工程,2007,5(3):203-205.

## 作者简介:

王平(1956-),男,本科,教授.研究领域:嵌入式智能终端,通信技术.

叶福兰(1981-),女,硕士,副教授.研究领域:数据挖掘.

陈章斌(1979-),女,硕士,讲师.研究领域:网络通信.