基于 ZK 的 MVVM 与 MVP 设计模式应用研究

林亚明

(闽江学院计算机科学系,福建 福州 350108)

[摘 要]简要介绍基于 ZK 框架的 RIA 系统架构,讨论 MVVM 设计模式和 MVP 设计模式在 ZK 框架中实现的技术特点和应用场合,综合两种设计模式的优点提出了复杂业务系统界面开发的最佳实践,并通过实例给予论证.实例表明,该最佳实践能够应付复杂的界面开发需求,同时保持代码的结构简洁,从而提高系统的维护性.

[关键词]ZK; RIA;设计模式; MVVM; MVP

[中图分类号]TP311 [文献标志码]A [文章编号]1673-8012(2012)06-0072-03

软件项目开发生命周期中,系统架构设计和 设计模式选择是否合理将直接影响到软件系统 最终是否成功. 系统架构设计离不开技术选型, 合适的开发技术将更容易实现系统架构设计目 标. RIA(rich internet application)技术作为实现 WEB 2.0 应用的关键技术,由于更加追求实现丰 富用户体验的用户界面而倍受开发者欢迎. 目前 RIA 主要有两类的实现方式:一是通过第三方商 业插件形式实现,比如 Adobe 公司的 Flex,微软 公司的 Silverlight, Oracle 公司的 Javafx 等; 二是 用 W3C 标准技术, 比如 CSS, Javascript 框架, html 5等. 前一种实现方式使用非标准化技术,技 术投人具有一定风险;后一种方式采用标准化技 术,具有平台无关性等优点,然而也存在问题,即 项目 B/S架构前后台需要使用不同的语言,比如 前台使用 Jquery 和 Javascript 语言,后台使用 JEE 技术和 Java 语言. 因此,项目需要不同语言专长 的人才,增加了项目成功实施的成本和难度;而 ZK 框架将 Windows 桌面式开发模式引入到 RIA 应用开发中,很好地解决了上述问题.

1 ZK 框架

ZK 是一个基于事件驱动和组件的框架,包括一个基于 AJAX 的事件驱动引擎,一整套丰富的 XML 用户接口语言(XML User Interface Lan-

guage)和 XHTML 组件,以及 ZUML(ZK User - interfaces Markup Language)的界面标识语言[1]. ZK 框架的开发语言主要使用 JAVA,也支持 JAVASCRIPT, RUBY, GROOVY, PHYTHON 等,多数情况下开发 WEB 应用可以 100%纯 JAVA 实现. 它通过封装屏蔽 HTTP 网络通讯的细节,使 API 模型与本地 JAVA 桌面开发类似. 因此,JAVA 程序员很容易将 JAVA 桌面开发的经验带到 WEB 开发中来,同时程序员不需要了解 JAVASCRIPT 编程,跨浏览器问题,复杂的 AJAX 通讯,商业逻辑接口的曝露(通常使用 WEB SERVICE)等一系列企业级 WEB 开发难题,节约了成本,节省了开发时间.

2 ZK 支持的设计模式

ZK 框架对 MVVM 设计模式^[2-3]和 MVP 设计模式^[4]的实现都提供相关 API 支持. ZK 实现的 MVVM 设计模式 API 简洁、直观; ZK 实现的 MVP 设计模式 API 控制灵活,但需要更多的编程. 两套 API 各有特点,因此有必要研究界面类型与设计模式选择之间的关系,从而总结出基于 ZK 的界面开发最佳实践.

- 2.1 MVVM 设计模式及其 ZK 实现
- 2.1.1 MVVM 设计模式

MVVM 设计模式代表了模型(Model)、视图

[[]收稿日期]2012-06-30

[[]作者简介]林亚明(1977-),男,福建福清人,讲师,硕士,主要从事软件工程、模式识别和面向对象技术方面的研究.

(View)和视图模型(ViewModel)^[5],是 MVC 设计模式的一个变型,用于帮助将业务对象逻辑与控制器逻辑从界面脱离. MVVM 隔离了模型层和视图层,同时避免了视图层与控制器层的紧耦合. 这样用户界面设计者与软件编程人员可以并行独立地开展项目工作,从而使项目开发效率得到极大提升,项目的开发进度也能够更好地跟踪和控制^[6-7].

ZK 遵循 MVVM 模式. 这里的视图模型 (ViewModel)层负责将模型(Model)中的值转化为视图(View),并且当视图被用户改变时,将其更新回模型. 图形用户界面由标记语言声明,该语言完全和模型(Model)中包含的数据分离. 在 ZK 框架中使用 ZK 用户界面标记语言(ZUML),而真正的业务逻辑驻留在 Java 代码中. 这里的一大优点是,用 ZUML 编写 ZUL 文件简单易学,能够被非编程人员编辑. 由此可以并行用户界面 (UI)和 Java 代码的开发,提高开发效率. 具体原理如图 1 所示.

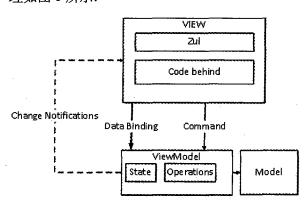


图 1 MVVM 设计模式

2.1.2 特点分析

从图 1 可以看出,视图模型代码没有依赖任何视图组件类和接口,只关心业务的抽象级别的处理,不依赖于具体的展现细节. MVVM 模式给 ZK 应用开发带来的好处有:

- 1) 代码仅仅是一个 POJO (Plain Old Java Objects),可以脱离 WEB 容器环境用 JUNIT 等单元测试工具对视图模型代码进行单元测试,提高了测试效率.
- 2) 当界面展现方式改变,视图模型代码几乎不需要做任何变化就可以直接使用,方便应对界面变化需求. 例如,如果把整数加法器的结果改为 Label 标签显示,只要修改 ZUL 文件.
 - 3) 界面甚至可以使用另外一套支持移动平

台的组件进行开发,从而实现移动平台开发与企业级平台开发的统一. ZK 的子项目 ZK MO-BILE^[8]进行了这方面的探索,目前移动平台市场竞争激烈,ZK MOBILE 将会有很大的发展空间.

4) 界面控制代码非常少,许多异步编程的 繁琐细节都由 ZK 绑定器帮助实现. 提升代码可 读性,降低开发和维护成本.

2.2 MVP 设计模式及其 ZK 实现

2.2.1 MVP 设计模式

MVP 是 MVC 设计模式的另外一种变型. MVP 中的 Presenter 替代了 MVC 中的 Controller, Presenter 负责处理与 View 相关的状态变化和视图设置,实现 View 和 Model 的解耦. View 接受用户请求并传递给 Presenter, Presenter 操作 Model, 当 Model 更新时,数据通过 Presenter 传递到 View 触发界面更新. 具体模式如图 2 所示.

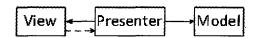


图 2 ZK MVP 设计模式

2.2.2 特点分析

从图 2 可以看出,基于 MVP 模式的 ZK 实现有如下特点:

- 1)通过注人方式,Presenter 能够访问到界面上的任何组件,编程手段很灵活,因此能实现很复杂的界面需求.但由于与界面组件 API 耦合,导致无法独立进行单元测试.
- 2)注解采用与 Jquery 类似的选择器语法,不但能够注入单个组件,甚至能够注入组件集合,方便 Presenter 的界面逻辑实现. 例如,若要实现注册页面的所有输入框清空,只需声明将所有输入框组件注入到一个组件 List 中,于是只需要一句迭代 List 的清空代码就可以实现清空功能.
- 3) Presenter 既要与业务层交互,又要与 View 层交互,导致项目中 Presenter 层承担较多 职责,代码集中于该层,导致代码分布不合理
- 4)由于与具体界面组件耦合,当需要迁移到 其他界面组件(比如移动平台),Presenter 往往需 要重写,导致重用率低.

3 两种设计模式的最佳实践

应用系统开发进入 WEB 2.0 时代后,复杂业务系统界面开发成为项目实施中很繁重的一项工作.本人在参与实际项目的研发中总结出以

下的 ZK 界面开发最佳实践:

- 1)可能情况下,尽量使用 MVVM 模式实现. 因为数据绑定器减轻了许多常规代码工作量, MVVM 模式开发效率高.
- 2) 当界面需求复杂, MVVM 设计模式无法 胜任或者实现困难时(例如界面的外观不是固定 的,随着数据状态的变化而变化), 采用 MVP 模 式, 控制实现更灵活.
- 3)若有可能,将页面中不变的部分剥离出来,形成子页面,尽可能利用 MVVM 设计模式,然后通过两种设计模式的组合方式来组装页面.

3.1 用例分析

最佳实践的第1点和第2点好理解,下面通过一个闽江学院在线考试系统的考试子模块来 展现第3点最佳实践的用法.

在线考试系统中,每份试卷的题型和数量可变,组卷时需要动态设定.考试子模块页面除了展示考生信息、计时、提交试卷等一系列功能外,还需要根据数据库中试卷信息动态生成试卷和答题卡,页面状态控制功能复杂,因此单独采用ZK MVVM模式实现困难.若采用ZK MVP模式实现,考试页面的Presenter需要包含以下功能:

- 1)加载试卷数据,生成相应的数据结构.
- 2)根据题型和数量动态构造考生界面.
- 3)处理考生答案上传,学生考试时间更新数 据库等一系列功能.

由此可看出,考试页面的 Presenter 层由于承担许多的职责,导致代码臃肿、可读性差等问题,增加了实现和维护的难度.于是,我们将每种题型对应到各自的题型子页面模块,每个题目的展示和学生答案提交由题型子页面模块负责,考试页面负责组合各题型子页面模块,具体伪代码实现流程如下:

- 1) 初始化页面时,使用 ZK 中 Selectors. wire-Variable 方法进行页面组件缠绕,使得在 Presenter 层方便访问页面组件对象.
- 2) Presenter 层从数据库读取试卷信息,根据每个题目的题型,使用 ZK 中 Executions. create-Components(path,component,map)方法让题型子页面负责展示,其中 path 是子页面的路径,componet 是父页面组件,用于控制子页面的位置,map 是传递给子页面的参数,试题信息通过该参数传递给子页面模块.
- 3)子页面 Presenter 层与主页面 Presenter 层 74

之间使用 ZK 中 BindUtils. postNotifyChange()方法,通过消息响应进行消息传递和处理,进一步解耦子页面与主页面模块代码.

3.2 特点分析

每道题目由子页面负责展示,减轻了考试页面控制器的负担.每个子页面由于业务逻辑单一,使用 ZK MVVM 设计模式方便实现,便于系统实现和测试;考试主页面使用 ZK MVVM 和 ZK MVP设计模式组合方式实现,在功能方便实现与代码可读性之间达到理想的平衡.

4 结语

MVVM设计模式实现UI设计的分离,使得软件设计流程和分工更加合理和规范,但是复杂界面控制用MVVM设计模式实现困难.MVP设计模式直接操作界面组件,易于实现复杂的界面展现逻辑,同时MVP设计模式中Presenter多种职责的重合导致模块内聚性差,不易维护.两者各有优缺点,只有在具体需求应用场合,具体分析,灵活应用,才能将两种设计模式的优势充分发挥出来.

[参考文献]

- [1] Henri Chen, Robbie Cheng. ZK 框架——Ajax 开发实战[M]. 蔡毅,译,北京:电子工业出版社, 2009.
- [2] Henri Chen. Hello ZK MVVM [EB/OL]. (2011 10 17) [2012 05 20]. http://books.zkoss.org/wiki/Small% 20Talks/2011/November/Hello% 20ZK% 20MVVM.
- [3] Hawk Chen. MVVM in ZK6; in Contrast to MVC[EB/OL]. (2011 12 16) [2012 05 20]. http://books. zkoss. org/wiki/Small _ Talks/2011/December/MVVM_in_ZK6;in_Contrast_to_MVC.
- [4] Wikipedia. Model view presenter [EB/OL]. (2011 08 21) [2012 10 26]. http://en. wikipedia. org/wiki/Model view presenter.
- [5]程国雄,胡世清. 基于 Silverlight 的 RIA 系统架构与设计模式研究[J]. 计算机工程与设计,2010,31(8): 1706-1713.
- [6]黎永良, 崔杜武. MVC 设计模式的改进与应用[J]. 计算机工程, 2005,31(9):96-97.
- [7]许鑫,费翔林. 基于 MVC 模式的应用软件开发框架 研究[J]. 计算机工程与应用, 2005, 30:102-104.
- [8] ZK partners. ZK Mobile [EB/OL] (2011 04 18) [2012 - 10 - 26]. http://www.zkoss.org/product/zk-mobile. (下转第 78 页)

华大学出版社,2001:180-188.

- [4] Sham Bhangal, Igor Choromanski. Flash 游戏设计宝典 [M]. 北京:清华大学出版社,2002:81-87.
- [5] Greif I, Sarin S. Data Sharing in Group Work[M]. Greif

I. Computer Supported Cooperative Work: A Book of Readings. Palo Alto, CA: Morgan Kaufmann Pub. Co., 1988:35-67.

Design and development of computer assembly virtual experiment

TANG Yong - fen

(College of Information and Engineering, Lianyungang Technical College, Lianyungang Jiangsu 222006, China)

Abstract: In the course of computer assembly experiment, it is easy to damage the computer because of the frequent disassembly and assembly. The article mainly introduces the building of a new opening experiment teaching mode with virtual computer assembly experiment platform on the basis of ActionScript. Compared with the traditional experiment lab, this experiment platform runs in the Web environment. And with browser and explorer, the students can use the platform any time and any place, which improves greatly the efficiency and decreases the experiment costs, enhances the students' enthusiasm of learning, creative and initiative spirit.

Key words: virtual experiment; ActionScript; Flash; interaction design

(责任编辑 吴朝平)

(上接第74页)

Application of MVVM design pattern and MVP design pattern based on ZK

LIN Ya - ming

(Department of Computer Science, Minjiang University, Fuzhou Fujian 350108, China)

Abstract: The ZK - based RIA architecture was briefly described. The technical features and applications about MVVM design pattern and MVP design pattern used in ZK were discussed. The best practices about the complex business systems interface development were proposed with the advantages of the integrated two design patterns, which is given to demonstrate by examples. The examples show that the best practices in maximizing the use of the MVVM design pattern are able to cope with the complexity of the interface development needs, thereby improving the maintainability of the system.

Key words: ZK; RIA; design pattern; MVVM; MVP

(责任编辑 吴 强)