

文章编号: 1007-757X(2012)7-0001-04

基于 Spring MVC 及 MyBatis 的 Web 应用框架研究

徐 雯, 高建华

摘 要: 基于 EJB 等重量级的 Web 应用框架存在很多问题, 如性能差、复杂度高、代码复用率低等, 提出了一种 B/S 结构与 C/S 结构相结合, 采用 Spring MVC 设计模式和 MyBatis 为基础的 Web 应用框架, 并对该框架的结构、组成等内容进行分析和研究。以 TOPCard 信用卡业务系统为应用实例, 说明 Spring MVC 和 MyBatis 在 Web 系统中的应用。通过实验结果分析, 基于 Spring MVC 及 MyBatis 的 Web 应用框架研究, 可以解决性能差、复杂度高、代码复用率低等问题。

关键词: Spring MVC; MyBatis; TOPCard 信用卡业务系统; 框架

中图分类号: TP311

文献标志码: A

Research of Web Application Framework Based on Spring MVC and MyBatis

Xu Wen, Gao Jianhua

(Department of Computer Science and Technology, Shanghai Normal University, Shanghai 200234)

Abstract: There are a lot of problems such as poor performance, high complexity, low reusability of code etc. in the Web application framework based on EJB. It is puts forward the Web application framework based on Spring MVC and MyBatis which combines B/S and C/S architecture, and analyzes and researches the frame structure and composition etc. in the paper. It is shows that Spring MVC and MyBatis application in Web system as an application example of TOPCard Credit Card Business System in the paper. Web application framework based on Spring MVC and MyBatis can solve problems such as poor performance, high complexity, and low reusability of code etc. through the analysis of experimental results.

Key Words: Spring MVC; MyBatis; TOPCard Credit Card Business System; Framework

0 引言

针对 Web 应用需求的日益增长, 快速开发高质量的 Web 应用系统, 成为业界研究的一大热点问题。一般, Web 应用表现层由 JSP、Servlet 实现, 主要完成与用户的交互; 业务层交给 EJB 实现, 完成系统核心业务逻辑的处理; 数据层由 JDBC 加 DAO 对象或是由 EJB 中的 EntityBean 实现, 负责与各种数据源的交互。由于 EJB 提供几乎所有的企业级服务^[1], 所以本身启动慢, 部署比较复杂, 而且性能比较差。本文采用 Spring MVC 设计模式和 MyBatis 框架, 提出了 B/S 结构与 C/S 结构相结合的 Web 应用框架, 应用于 TOPCard 信用卡业务系统, 说明 Spring MVC 和 MyBatis 在 Web 系统中的应用, 该系统在性能差、复杂度高等方面有很大的改进。

本文的章节安排如下: 第二部分对 Spring MVC 模式进行简述、深入研究 MyBatis 框架。第三部分以 TOPCard 信用卡业务系统为例, 对一种 B/S 结构与 C/S 结构相结合, 采用 Spring MVC 和 MyBatis 为基础的 Web 应用框架进行分析和研究。第四部分得到实验结果并进行分析。

1 Spring MVC 及 MyBatis 框架研究

1.1 Spring MVC 模式

MVC 模式 (Model-View-Controller) 是软件工程中的一种软件架构模式, 3 个基本部分为 Model、View、Controller^[2]。Spring 是以轻量级的开源框架, 它以控制反转原则和面向方面编程思想为基础, 提供管理业务对象服务^[3]。Spring 框架提供构建 Web 应用程序的全功能 MVC 模块^[4], 它是一种高度可配置的 MVC 框架。Spring MVC 实现了 MVC 的核心概念, 它为 Controller 和处理程序提供了大量与此模式相关的功能, 当向 MVC 添加反转控制时, 它使应用程序高度解耦, 提供简单的配置更改就可以动态地更改组件的灵活性, 图为 Spring MVC 设计模式结构, 如图 1 所示:

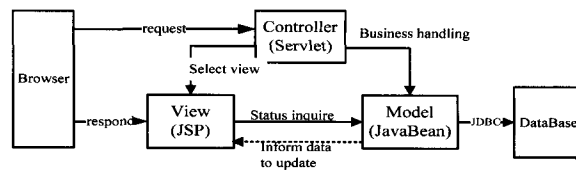


图1 Spring MVC 设计模式结构

基金项目: 国家自然科学基金资助项目(编号: 60673067), 上海市科委项目(编号: 09220503000)和上海市引进技术的吸收与创新计划(编号: 2010CH-014)。

作者简介: 徐 雯 (1989-), 女, 上海师范大学计算机科学与技术系, 硕士研究生, 研究方向: 软件可靠性设计理论与方法、软件测试技术, 上海, 200234

高建华 (1963-), 男, 工学博士, 上海师范大学计算机科学与技术系, 教授, 研究方向: 软件可靠性理论与设计、软件开发环境与开发技术、数据安全与计算机安全、网络测试、LSI/VLSI 测试等领域, 上海, 200234

Spring MVC的Web框架是一个请求驱动的Web框架,整个框架由Model, View, Controller 3个部件组成。Spring MVC的重要组成部分是DispatcherServlet,它的主要任务是将请求分发到相应的处理程序,让这些业务程序来处理这些请求。以下是TOPCard信用卡业务系统中Dispatcher Servlet的声明和映射:

```
<servlet>
  <servlet-name>topcardweb</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>topcardweb</servlet-name>
  <url-pattern>*.ctl</url-pattern>
</servlet-mapping>
```

Dispatcher Servlet被命名为topcardweb拦截web请求,上述配置将随web应用启动而自动加载,采用通配符与URL对应来进行配置,所有以.ctl作为后缀的请求将都会被topcardweb拦截,并将拦截的请求通过控制器映射(Handler Mapping)发送给相应的Bean进行处理。

1.2 MyBatis框架

MyBatis框架集合多种操作型关系数据的概念和方法,它是一个强大的数据访问工具和解决的方法^[5]。MyBatis框架主要包含DAO组件与SQL Map组件两大类,MyBatis常用工具有Log管理等,其架构图,如图2所示:

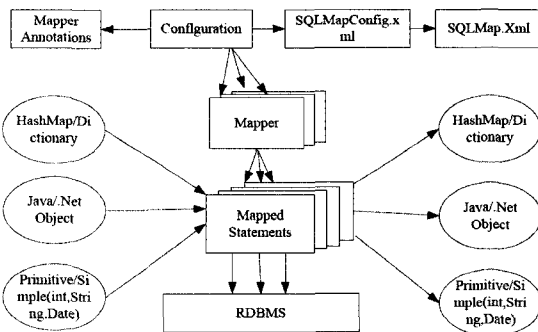


图2 MyBatis架构图

MyBatis DAO组件的主要目标是抽象化应用程序的数据访问层和持久层的表示方式和位置,使它远离应用程序的业务逻辑,其主要功能是帮助开发人员进行基于DAO设计模式设计和开发J2EE应用程序。SQL Map组件是MyBatis Database Layer框架的重要组成部分,它使用简单的XML配置文件将Java Bean、XML、Map映射成SQL语句,通过SQL语句的执行获得Java Bean、XML、Map等对象。MyBatis SQL Map能大大减少访问数据库的代码,并且提供了一个简洁的框架,MyBatis DAO实现图和SQL Map实现图见参考文献^[9]。

Log管理是MyBatis不可缺少的部分,它管理4种模式的日志:①Jakarta Commons Logging (JCL) 记录日志信息;②Log4J;③JDK1.4 Logging API;④无日志模式。MyBatis

框架使用JCL记录日志信息,JCL使用独立于具体实现的方式提供日志服务,可以插入包括Log4J和JDK1.4 Logging API等不同的日志服务实现。为了对日志做一定的封装,MyBatis采用简单的工厂模式,其Log类结构,如图3所示:

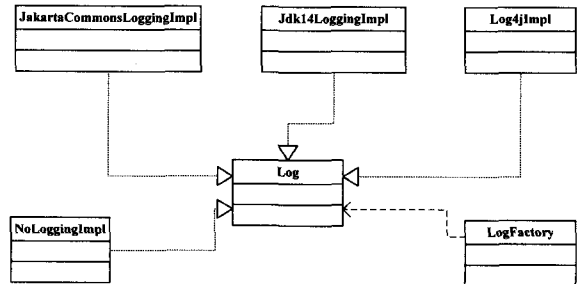


图3 MyBatis平台的Log类结构图

LogFactory依赖Log接口,它是日志的工厂类,Log接口能通过NoLoggingImpl、JakartaCommonsLoggingImpl、Jdk14LoggingImpl、Log4jImpl来实现。

2 Spring MVC 和 MyBatis 在 TOPCard 信用卡业务系统中的应用

2.1 TOPCard信用卡业务系统的拓扑结构

TOPCard 信用卡业务系统的拓扑结构,包括信用卡的申请、建账制卡、授权、清算等核心业务,提供信用卡业务的核心支持。辅助系统通过与核心信用卡系统的协作来完成信用卡的其他业务功能,如图4所示^[8]。

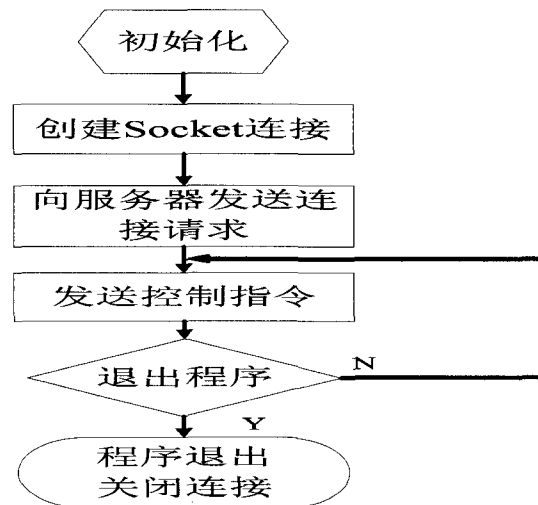


图4 TOPCard信用卡业务系统的拓扑结构

各个系统的功能如下:

(1) TOPCard 信用卡业务系统进行信用卡核心业务的处理,包括申请、建账制卡、授权、清算等。

(2) TOPCard 信用卡业务系统与 TOPCard 核心业务系统之间通过在线和批量的方式进行交互。在线方式包括ATM、柜面取现等交易。批量方式包括清算、总帐等文件。

(3) 卡中心员工、分行柜员, 通过直接登录 TOPCard 信用卡业务系统的方式进行业务处理。

(4) 行内核心系统向 TOPCard 信用卡业务系统发出的文件主要有: 行内消费预借现金请款文件、柜面还款清算文件、约定还款应答文件。TOPCard 信用卡业务系统向行内核心系统返还的文件主要有: 会计总账文件、约定还款请求文件。

(5) 国际组织将收到的请款文件, 给 TOPCard 信用卡业务系统进行入账, TOPCard 信用卡业务系统批量结束后, 将当日的争议退单等信息生成返还文件, 传回给各个国际组织前置机, 以便传回给国际组织。

(6) 辅助业务系统, 包括影像征审系统、CallCenter、网银系统、积分系统、防欺诈系统、催收系统、数据集市。

影像征审系统进行申请件的处理、影像的管理以及对持卡人的征审和授信处理, 为信用卡系统提供申请信息来源。

CallCenter、网银系统用 TOPCard 信用卡业务系统提供的客户信息、账户信息等, 为客户提供全方位的服务。

积分系统, 利用 TOPCard 信用卡业务系统提供的交易数据, 为持卡人累计奖励积分并实现相应的兑换等功能。

防欺诈系统, 利用 TOPCard 信用卡业务系统提供的交易信息, 按照预定的规则给卡中心提供预警信息。

催收系统, 处理 TOPCard 信用卡业务系统中处于拖欠等状态的客户, 跟踪对应的客户的行为。

数据集市, 对 TOPCard 信用卡业务系统的业务数据进行整合, 提供各种分析结果用于产品的风险、效益评估, 同时作为市场推广和营销的依据。

TOPCard信用卡业务系统, 在前台连接模式和客服和卡中心运营渠道中, 客服、卡中心通过浏览器访问Web服务器, Web服务器采用JDBC和TCP/IP两种方式与信用卡主机通讯。WEB应用采用Spring MVC框架, 应用逻辑采用面向服务的设计思想, 资源管理层采用MyBatis的工具对数据库进行访问。

2.2 TOPCard信用卡业务系统的三层结构

TOPCard 信用卡业务系统采用 B/S 与 C/S 混合模式。在客服和卡中心运营渠道以及前台连接模式中采用 B/S 模式, 分为 3 层结构, 具体如下: 客服、卡中心、Web 服务器、信用卡主机。在联机渠道中, 采用 C/S 模式, 分为三层结构, 各层之间采用 TCP/IP 通讯。具体如下: 国际组织、各组织接口机、TOPCard 系统; 银联系统、银联前置、TOPCard 系统; 行内渠道、行内前置、TOPCard 系统。

TOPCard信用卡业务系统建立在B/S与C/S三层结构基础上, 这样的设计方法有利于设计大型的分布式程序。通过把系统的功能分离到多个逻辑层面, 将一些敏感的数据放在最下层, 即在防火墙后面处于一个非常安全的地方。应用服务技术, 可以使应用程序基本上不直接访问系统和操作系统, 这样就减少了像大多数两层结构的弱点。TOPCard信用卡业务系统的三层结构。如图5所示:

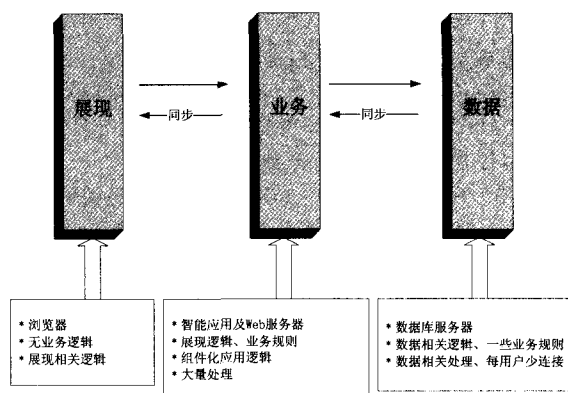


图5 TOPCard信用卡业务系统的三层结构

2.3 TOPCard信用卡业务系统的逻辑结构

Spring MVC及MyBatis架构的整合, 能将Web系统中的表示层、业务层和逻辑层有效地分开, 从而有利于web系统的整体维护和升级^[6]。Controller实现了org.springframework.web.servlet.mvc接口, Model主要用来完成对MyBatis数据库的操作, 由Dao, DaoImpl, Service和ServiceImpl相关的接口和类在程序中来实现, View用JSTL和Xml来完成^[7], 如图6所示:

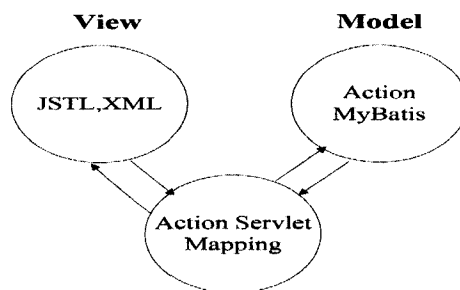


图6 Spring MVC和iBATIS的整合图

TOPCard信用卡业务系统逻辑结构图, 如图7所示:

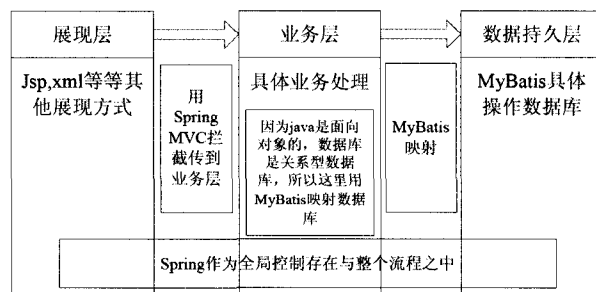


图7 TOPCard信用卡业务系统逻辑结构图

Spring作为全局控制存在与整个流程之中。

(1) 在展现层中, 为用户接口(展示页面)提供信息供应, 本层采用Spring MVC框架做表现层平台。首先通过JSP页面实现前台交互, 负责传送Request和接收Response, 然后Spring MVC根据配置文件将DispatcherServlet接收到的Request委派给相应的Controller处理。

(2) 在业务层中, 实现具体业务逻辑, 为应用展现层提供服务支撑, 本层采用Spring框架技术实现。管理服务组件的Spring IOC容器, 负责向控制器提供Model组件和该组件的DAO组件完成业务, 并提供事务处理、缓冲池等容器组件以提升系统性能和保证数据的完整性。

(3) 在数据持久层中, 为上层业务服务提供数据访问支持, 本层采用MyBatis技术实现。依赖于MyBatis的对象化映射和数据库交互, 处理DAO组件请求的数据并返回处理结果。

2.4 TOPCard信用卡业务系统运行流程图

TOPCard信用卡业务系统, 用Spring MVC作为MVC开发模式做展现层平台, 利用MyBatis作为O/R Mapping数据持久层框架, Spring作为全局控制存在与整个流程之中, 其运行流程, 如图8所示:

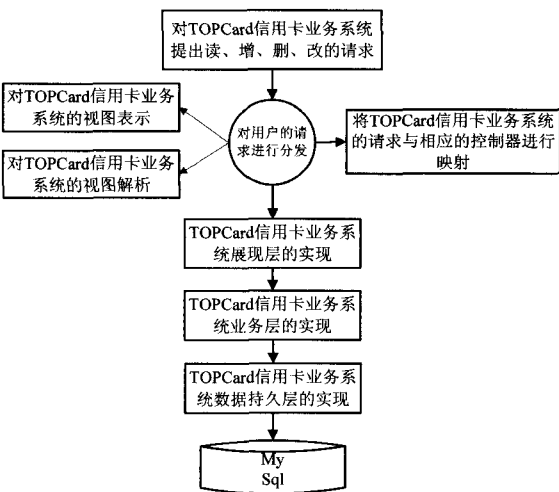


图8 TOPCard信用卡业务系统运行流程图

3 实验结果与分析

为了进行实验对比分析, 本文进行比较的基于EJB的信用卡业务系统其业务层交给EJB实现, 完成系统核心业务逻辑的处理; 数据层由EJB中的Entity Bean实现, 提供原始数据面向对象的封装, 借助CMP(容器管理持久性)获得O/R映射能力, 完成与各种数据源的交互。实验软件环境参数如下:

- 信用卡主机、备机: AIX 或 HP-UX 或 Solaris
- 数据库: DB2 或 Oracle
- Web 服务器: linux(Redhat ES)、WebSphere
- 银联前置: linux(Redhat ES)或 solaris for X86
- 客服工作站: Windows XP
- 卡中心工作站: Windows XP

3.1 性能分析

系统性能是一个总体概念, 包括了硬件、网络、数据库、应用各个方面的配置和数量, 同时还必须考虑配置成本, 即不能为了保证系统性能而通过任意扩大系统配置的办法来满足系统性能。因此构建优秀的应用软件体系, 对发挥系统整体性能就显得尤为关键。

影响信用卡应用系统总体性能的主要是联机授权模块、批处理及报表模块等, 这些模块中的性能薄弱都将导致整个系统的业务无法充分展开, 影响到银行的业务运作。

各个模块性能瓶颈可能出现的位置是: 联机授权模块主要存在于通讯能力、数据库操作以及应用服务器的信息处理能力; 批处理及报表模块的性能瓶颈主要存在于对数据库的读写、文件的写操作以及服务器的计算能力上。由于联机授权模块、批处理及报表模块是系统中对性能要求最高的部分, 如表1所示:

表1 TOPCard与基于EJB信用卡业务系统的性能测试结果

模块	TOPCard 信用卡业务系统	基于 EJB 的信用卡业务系统
联机授权方面	平均每秒处理交易笔数: 20	平均每秒处理交易笔数: 10
	峰值处理交易笔数: 100	峰值处理交易笔数: 60
	主机单笔交易处理时间: <1 秒	主机单笔交易处理时间: <2 秒
批处理及报表	核心批处理时间: <4 小时	核心批处理时间: <5 小时

TOPCard 信用卡业务系统在预测银行 10 年的业务量下的性能测试结果与基于 EJB 的信用卡业务系统的比较。假设信用卡帐户 500 万张卡, 日均交易 20 万笔, 200 人并发访问 WEB 服务。

在性能方面, 可以看出基于 Spring MVC 及 IBATIS 的 TOPCard 信用卡业务系统大大优越于基于 EJB 的信用卡业务系统。

3.2 复杂度分析

基于 EJB 的信用卡业务系统为了实现全面且完整的系统应用时, 重量级容器 EJB 会带来诸如部署复杂, 运行缓慢, 内在服务多等负面作用。EJB 要求创建多个组件接口并实现多个不必要的回调方法组件接口要求实现 EJBObject 或 EJBLocalObject 以及处理许多不必要的异常。基于 XML 的 EJB 部署描述符比较复杂并容易出错。EJB 组件并非是真面向对象的, 因为它们在继承和多态性方面的有使用限制开发人员无法在 EJB 容器外部测试 EJB 模块, 而在容器内部调试 EJB 非常复杂和耗时。调用 EJB 是一项复杂的任务。即使是在应用程序中使用最基本的 EJB 也需要对 JNDI 有一个详细的了解对容器的依赖使得 EJB 只能用于服务器组件的开发, 无法实现一次编写。

在复杂度方面, 基于 EJB 的信用卡业务系统的复杂度大于基于 Spring MVC 及 IBATIS 的 TOPCard 信用卡业务系统。

3.3 代码复用率分析

Spring 是轻量级的框架, 对其他对象没有侵入性, 代码有很高的复用性, 并且是一个免费的开源框架; 而 EJB 是重量级的框架, 难以维护和复用。所以显而易见, 基于 Spring MVC 及 IBATIS 的 TOPCard 信用卡业务系统, 能解决基于 EJB 的信用卡业务系统的代码复用率低的问题。

(下转第 10 页)

3.4 中的函数说明可知测试用卡为 Mifare1 S50 卡。

4 结论

RFID 与 WSN 融合技术在实际中应用越来越广泛, 无时无刻不改变着人们的生产、生活方式。相信 MFRC522 和 XBee 系列模块将会发挥出更大的作用, 成为信息高速通道的中流砥柱, 给人类的生活面貌带来革命性的改变。

但是我国目前 RFID 的发展仍处于研究阶段, 优秀的技术和产品依然不是很多。而利用 Keil 软件和 Atium Designer 软件进行 RFID 软硬设计不仅大大减少了研发成本, 而且为开发者提供了一个完整的平台。开发者只要利用这两个软件和相应的硬件即可进行产品开发。本文以最低花费的软硬件资源探索了 RFID 与 WSN 融合技术的实现过程, 提出了一套完整、可行的软硬件开发方案。

实现了 RFID 与 WSN 融合技术中软硬件接口, 并且在这一基础上进行了 RFID 软件比较深入的研究, 分析了一项技术的核心应用方法, 完成了软件与硬件的联合调试, 并实现了读写 Mifare1 S50 卡数据块的功能, 具有高实用性和参考价值。

在软件方面, 分析并编写了相关的系统监控程序和接口

程序, 研究了一种使用串口调试助手进行软硬件联合调试的实用方法; 研究了通讯函数的详细使用方法; 为嵌入式软件开发提供了一种实用设计思路; 针对对嵌入式软件开发提出了底层功能函数和基本功能函数两种新的分类, 更是完成了接口软件的设计目标, 值得推广借鉴。

参考文献:

- [1] 物联网世界[OL]. <http://www.rfidworld.com.cn/>
- [2] 黄玉兰. 射频识别 (RFID) 核心技术详解[M]. 人民邮电出版社, 2010
- [3] 张迎新. C8051F 系列 SOC 单片机原理及应用[M]. 国防工业出版社, 2009
- [4] 李斌, 李文峰. WSN 与 RFID 技术的融合研究[J]. 计算机工程, 2008, 34 (9), 127—129
- [5] 李昊. 基于三层架构的 RFID 接口模型的研究与应用[D]. 西安: 西安电子科技大学, 2008

(收稿日期: 2012.04.28)

(上接第4页)

4 总结

本文通过对采用 Spring MVC 和 MyBatis 框架为基础的 Web 应用框架进行分析和研究, 可以发现将 Spring MVC 应用于 TOPCard 信用卡业务系统的架构中, 可以有效的将系统的表示层、业务层、逻辑层分开来, 提升系统性能, 提高了系统的可维护性和可扩展性; 将 MyBatis 应用到 TOPCard 信用卡业务系统中, 不仅可以大大提高代码的复用性, 而且能有效隔离数据库设计 and 应用中使用的对象模型, 这就使得它们两者之间的相关性能降至最低。将 Spring 作为全局控制存在与整个流程之中, 使得系统复杂度大大降低, 提高了代码复用率。Web 应用框架的研究是一个不断深入、逐步完善的过程, 在对 Web 应用框架不断推广试用的基础上, 还将进一步扩展其功能, 使其应用领域更加广泛。

参考文献:

- [1] Johnson, Juergen Hoeller. R. Expert One-on-One J2EE Development without EJB[M]. Indiana: Wiley Publishing Inc, 2004.
- [2] Ren Yongchang, Xing Tao, Xing Zhaofeng, Zheng Jianchun. Application Research for Integrated SSH Combination Framework to Achieve MVC Mode[C]. International

Conference on Computational and Information Sciences, pp.449-502, Sept. 2011.

- [3] Shi GAO, XingHeng WANG, Xue DENG. Public Computer Education Management System
- [4] based on Java EE Lightweight framework[C]. International Conference on Computer Science and Service System, pp. 1486 - 1489, Jun. 2011.
- [5] 张宇, 王映辉, 张翔南. 基于 Spring 的 MVC 框架设计与实现[J]. 计算机工程, Vol.36, No.4, pp.59-62, Feb. 2010.
- [6] Clinton Begin, Brandon Goodin and Larry Meadors. iBatis in Action. Manning publications Co. Chero Station, 2007.
- [7] Harvey M.Deitel, Paul J.Deitel 著, [M]刘宗田, 苗沛荣等译, Java 程序设计教程, 机械工业出版社, 2002.
- [8] Kumar, B V Sangeetha, S Subrahmanya S V 著[M], 宋梅, 张学平译, 深入浅出J2EE架构, 清华大学出版社, 2006.
- [9] TOPCARD 信用卡核心系统产品说明书, [K]上海华腾软件系统有限公司, 2009.
- [10] 封小钰, 王飞. 基于 iBATIS 数据库访问技术的研究与应用. [J] 电脑开发与应用, Vol.21, No.6, pp.51-53, Apr.2008.

(收稿日期: 2012.05.10)