

高校教师业务管理系统^①

乔向杰

(北京联合大学 旅游学院, 北京 100101)

摘要: 本文设计并实现了基于 J2EE 的高校教师业务管理系统. 文章详细讨论了系统实现的业务模型、功能模型等需求分析过程, 以及系统架构、通用查询框架、基于角色访问控制策略的“用户-角色-功能模块”的权限管理系统设计与实现等关键技术.

关键词: J2EE; 通用查询框架; 高校教师业务管理系统; 基于角色的访问控制策略

College Teachers' Professional Skills Management System

QIAO Xiang-Jie

(Tourism Institute of Beijing Union University, Beijing 100101, China)

Abstract: A college teachers' professional skills management system is designed and implemented based on J2EE. The requirement analysis process which involves business modeling and function modeling, the key technologies of system's technical architecture, general query frame and access control strategy based on RBAC are respectively discussed.

Key words: J2EE; general query frame; a college teachers' professional skills management system; role based access control

实现教师的专业化可持续发展, 职业生涯规划是一种有效的手段. 近年来, 许多高校为了适应国家高等教育改革的新任务和新要求, 都提出了教师的职业生涯规划项目来加强高校教师队伍的建设. 在建设的过程中, 需要评审和记录高校教师的教学、教研、教法及相关管理方面的信息; 认定教师承担科研方面的信息; 申报和评审继续教育情况的信息; 申报和评定参加有关社会服务, 如国内外访学、挂职锻炼等活动等等, 因此, 我们提出了要构建一个高校教师的业务管理系统来对这一过程的建设与实施进行管理与监控, 以便于学校能从整体把握教师职业成长的动态, 提高信息管理的效率. 职业生涯规划目标不仅要与个人的职业发展观和发展需求相联系, 还要与学校的总体发展目标保持一致. 由于每个学校的阶段定位与发展目标, 以及教师的素质和特点都不同, 因而对于教师职业生涯规划的内容和要求也不尽相同. 因此, 要想找到一个已经成型而且完全适用的现成系统是比较困难的, 而且此系统同一般教师业务档案管理系统不同,

它并不仅仅是对教师业务档案静态信息的管理与维护, 而更强调的是教师在提高自身业务水平, 开展职业生涯规划的发展路径的动态过程管理. 本文将以北联合大学旅游学院的教师业务管理系统的业务模型为例, 探讨系统的架构、功能与关键技术实现, 希望能对其他类似院校研发相关的系统提供借鉴.

1 系统分析

系统开发的实践表明, 用户需求很难在系统建立前被严格地定义和描述出来, 而且用户的需求是在不断的变化中的^[1,2]. 而原型法则弥补了这一缺陷, 通过快速开发一个含有最终系统主要特征的原型系统, 使用户在这个原型中得到启发并同开发人员一起修改和评估模型, 通过反复的修复来产生用户满意的原型, 最终形成系统. 因此, 为了保障系统需求分析的正确性、完整性及有效性, 我们先通过开展需求调研, 同相关的职能部门及系部进行沟通, 收集相关的文件资料, 然后进行系统的业务流程、数据流程、组织功能分析,

^① 基金项目:北京联合大学新起点计划项目(zk201220)

收稿时间:2013-08-07;收到修改稿时间:2013-09-02

形成系统的逻辑设计方案;既而开始构建原型,由于最终系统考虑为 B/S 模式,因此,为了表达系统的真实性,我们采用静态网页技术与客户端脚本技术迅速实现一个系统的原型,该原型充分体现出了真实系统的界面与操作逻辑,只是没有实际的数据库存储,因此很容易让用户接受大大提高了与用户沟通的效率,经反复确认与修改后,开始进行系统的概要设计与详细设计,最后才开始进行编程、测试与部署。

1.1 业务模型分析

在进行业务模型分析时,主要以学院发布的相关文档:《北京联合大学教师职业生涯规划实施方案(试行)》、《科研团队及类型》、《科研团队建设基本原则》、《北京联合大学旅游学院青年教师过教学关征求意见稿》等为依据。根据学院对教师职业生涯规划实施方案的要求,教师在制定职业生涯规划时,首先要全面了解本单位和学院的总体发展目标与方向、学科专业建设规划、师资队伍建设规划等基本信息,以使个人规划目标与本单位及学院发展目标更紧密地结合起来。教师个人的职业生涯规划设计职责包括:自我职业评估(包

括职业能力、职业兴趣、个性、职业价值观、工作业绩等方面的职业评估),与本单位负责人交换意见,设定个人的职业生涯发展目标,执行职业生涯发展策略的行动计划,进行自我职业发展的绩效评估。对参与职业生涯规划的教师,要求完成教学能力提升、科研能力提升、企业(行业)实践和访问学者等方面的任务。因此,在教师的职业生涯规划信息系统中可以设计为两个阶段:一个为职业生涯的规划阶段,一为职业生涯规划的实施阶段。规划阶段主要为教师对自我进行评估,然后由系部党政负责人和专业负责人进行评估,最后由教师本人进行确认后形成教师职业生涯规划;实施阶段主要为教师根据每项规划要求至考核期后进行项目申报,并由相关职能部门(人事处、教务处、科研处)进行审批,直至教师完成所有规划项目。教师可在此期间随时查询自己的规划执行进度与状态。同时,院领导、系部领导也可对教师的职业生涯规划及实施情况进行实时的查询与统计分析,以便即时调整实施方案,提高实施效率与教师满意度。系统的业务模型描述如图 1 所示。

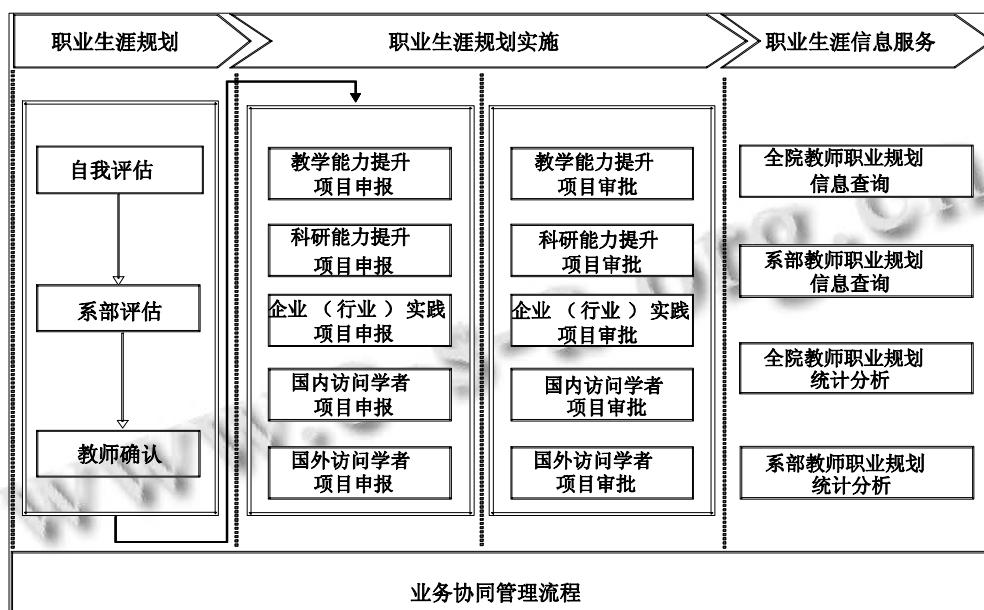


图 1 系统业务模型

1.2 功能模型分析

明确了系统的业务模型,就可以对系统进行信息模型的分析,以完成系统的逻辑设计,最终进行物理设计,并开始系统实施。系统的功能模型如图 2 所示。系统的功能架构主要有四大部分组成:职业生涯规划

划、职业生涯规划评估认定、职业生涯规划信息服务及系统管理。其中职业生涯规划模块主要为教师本人进行职业生涯规划的业务模块,通过此模块教师可以进行自我评估及确认、对各项规划项目的申报以及对职业生涯规划总体信息的查询;职业生涯规划评估认定模块主要为

系部对教师职业规划的评估以及各职能部门对教师职业规划申报项目的审批和认定；职业生涯信息服务是院领导及系部领导对全院及各系部教师职业规划信息的查询与统计分析，通过此模块可以使院系领导对全

院及各系部的教师职业规划的进度及整体情况进行把握和跟踪，以便进行适时和适当的政策与实施调整等。系统管理主要为系统管理员设置用户、权限、角色，以及进行模块的管理、公共参数的设置等。

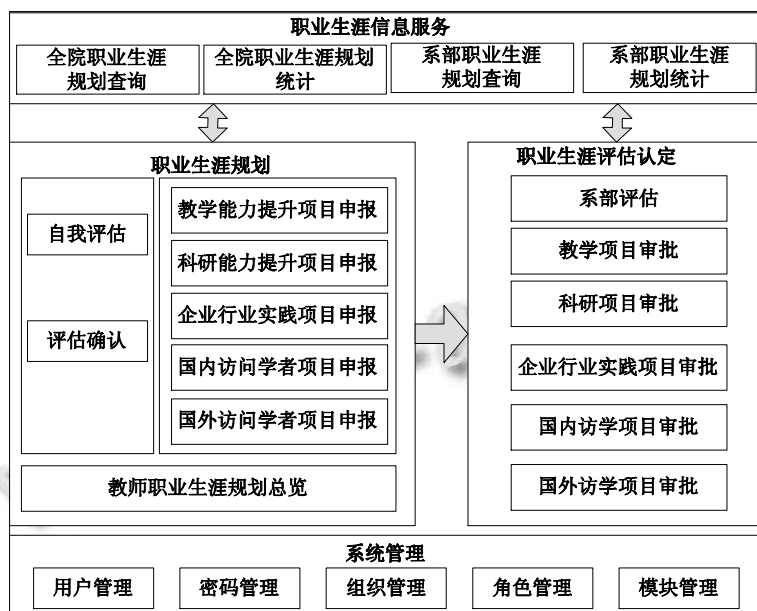


图 2 系统功能模型

1.3 用户角色功能分析

使用该系统的用户可以分为教师、系部领导、教务处、科研处、人事处、院领导、系统管理员等，具体的描述及职责如表 1 所示。

表 1 系统用户角色职能列表

用户	描述	职责
教师	参与教师职业生涯规划的教师职工	进行个人评估与确认、规划项目的申报
系部领导	各系部的教师职业规划负责人	进行系部对教师职业规划的评估意见与指导、查看和跟踪本系部教师的职业生涯规划及实施情况
教务处	负责教师职业生涯规划教学项目相关的工作人员	进行教师职业生涯规划教学能力提升项目申报的审批
科研处	负责教师职业生涯规划科研项目相关的工作人员	进行教师职业生涯规划科研能力提升项目申报的审批
人事处	负责教师职业生涯规划企业（行业）实践、国内外访学项目相关的工作人员	进行教师职业生涯规划社会服务能力提升项目申报的审批
院领导	分管教师职业能力提升项目的相关领导	查看和跟踪全院教师的职业生涯规划及实施情况
系统管理员	负责系统设置与维护的管理人员	进行系统用户管理、角色管理、用户权限分配、字典管理等

2 系统设计

2.1 总体框架与思路

系统采用基于 J2EE 标准的 B/S 架构实现，总体框架与思路如图 3 所示。在基础层构建系统的基本软硬件环境，包括网络环境、服务器、系统软件、中间件，以及进行统计分析的图表展现工具。其中，网络环境直接通过校园网接入，教师在校外登录系统时采用 VPN 方式即可；服务器实际应包括应用服务器与数据库服务器，在考虑系统用户数与并发量后，可将应用与数据库服务放在同一台性能较高的服务器上运行；系统软件及中间件等主要为操作系统（主要采用网络操作系统如 Windows Server 系列、Linux 等）、WEB 服务器软件与应用服务器软件（如 Weblogic、Tomcat 等）。通过教师的职业生涯规划与实施，系统将记录整个职业生涯规划的生命周期，形成教师职业生涯规划的基本信息数据库，主要包括教师的申报信息、审批信息等。整个系统基于 J2EE 的体系结构，用户最终通过客户端的浏览器来使用系统。

2.2 系统架构设计

系统采用基于 J2EE 的 struts+hibernate 分层结构实

现, 即 MVC 体系结构(Model-View-Control)将显示逻辑与事务逻辑分开处理, 便于系统的维护和升级^[3-5]. 系统实现架构如图 4 所示.

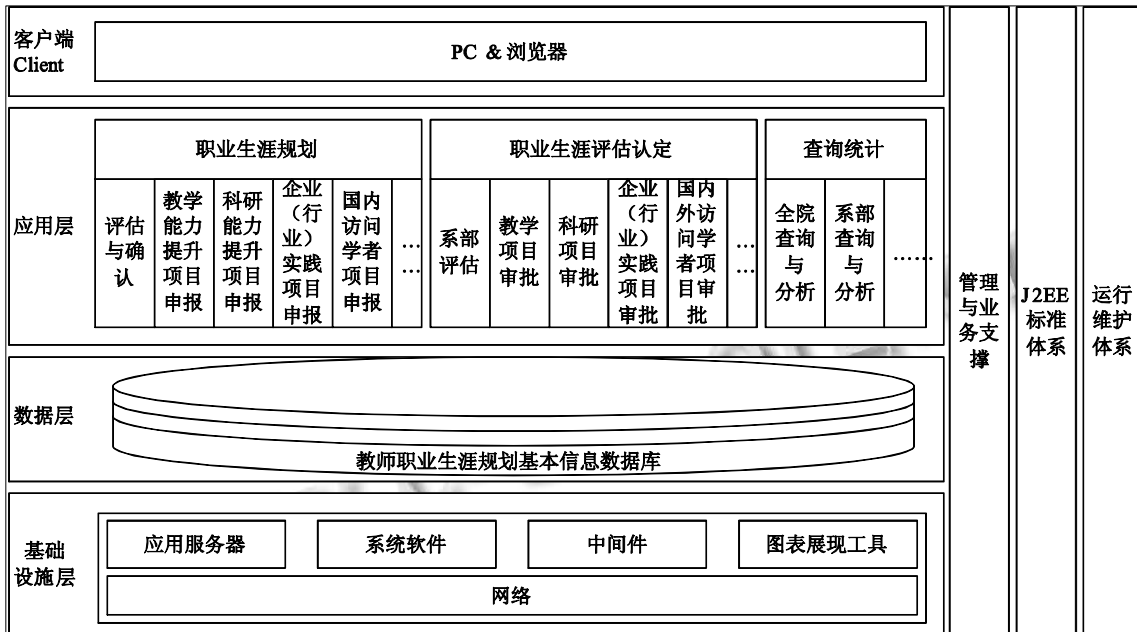


图 3 系统总体框架

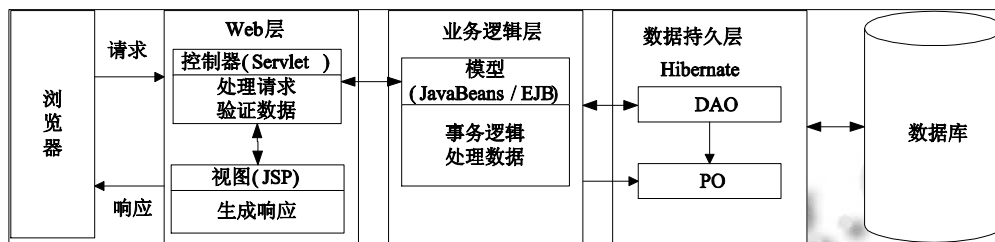


图 4 系统实现的架构

浏览器层: 用户通过浏览器与系统进行交互, 浏览器向系统提交请求后, 经过系统处理仍返回到客户端浏览器界面.

Web 层: 从客户浏览器获得客户输入, 对数据进行验证后传递给逻辑层的组件, 再将从业务逻辑层获得的处理结果返回给浏览器. Web 层由控制器 Servlet 和视图 JSP 组成, 封装在 Web 容器中. 业务流程控制一般均由控制器 Servlet 开发, 响应用户的查询等请求并调用业务逻辑层的 JavaBean 来实现复杂的业务逻辑^[3].

业务逻辑层: 处理控制器传递过来的用户数据, 经过业务及数据处理后返回给控制器, 再由控制器返回给视图层显示结果. 业务逻辑层封装了系统提供给

用户的接口, 主要由 JavaBean 或 EJB 来实现.

数据持久层: 完成持久对象到关系数据库的映射, 并对持久对象进行操作. 业务逻辑模块由 Hibernate 通过 O/R 映射文件实现对具体数据源的操作.

数据库层: 对象持久性的具体实现, 本系统分别采用了 Mysql 和 oracle 进行了测试实现.

2.3 通用查询框架设计

对一个业务对象的管理所涉及的业务操作往往都包括新增、删除、修改、查询、浏览等操作, 比如在设置查询条件后显示查询列表, 进而可以进行新增一个项目或对查询结果列表中的记录进行修改、删除和浏览等. 因为这些操作的原理基本一致, 为了提高开发效率, 系统进行了通用查询框架的设计, 以简化对

每一个查询实例的编码。通过设计通用查询的基类，将常用的数据库表的查询、新增、删除、更新等操作进行类的方法实现，相应模块在实现时则只须根据本模块的需求进行界面内容的设计，在业务逻辑上则直接调用基类相应的方法即可，而无须再在每一个模块的 Action 类中编写重复的代码，从而能大大提高开发的整体效率。

2.3.1 通用查询类设计

系统主要设计了四个通用类：Action、Manger、Form 及 Hibernate Pojo 类。使用 DispatchAction 类型的 Action 作为业务流程的控制类，使用 Manager 类处理业务逻辑。Form 为 Struts 的 ActionForm 类，负责视图与控制层之间传送数据。还有一个是对应数据库表的 Hibernate Pojo 类。这些对象总是一一对应的，其关系如图 5 所示。

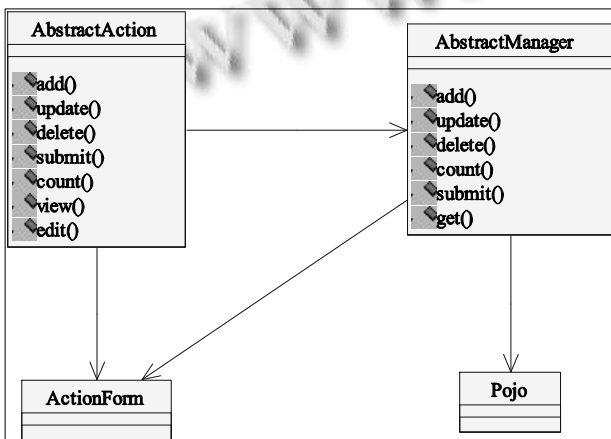


图 5 通用查询类设计

架构中提供 Action 和 Manager 的一个通用实现作为所有业务 Action 和 Manager 的基类，通用基类里主要实现对数据库表进行增删改查的操作流程和业务逻辑，因此子类中不必再处理这些通用业务操作。

Action 实现一个 getManager 方法，返回与该 Action 相对应的 Manager，用于基类进行通用增删改查功能，这些流程子类不用实现，只需另外实现各业务特定的流程即可。

Manager 实现一个 initManger 方法，在这个方法里设定该 Manager 处理的数据库 pojo 对象的类、处理的数据库表和表的主键，这个方法在基类的构造函数里调用。同样，Manager 的子类也不需要实现通用的增删改查数据操作，只需另外实现各业务特定

的逻辑即可。

2.3.2 Struts 配置

对于一个完整的功能模块来说，主要包括查询、新增、编辑、浏览、删除等操作，对每一个功能模块将设置一个该模块的 struts 配置文件 struts-config-xxx.xml(其中 xxx 代表一个模块的包名)，然后把该配置文件加到 Struts 的主配置参数上：

在 WEB-INF\web.xml 找到 Struts 的控制层 action sevlet，增加一个初始参数：

```

<init-param>
    <param-name>config/xxx </param-name>
    <param-value>/WEB-INF/config/xxx/
        struts-config-xxx.xml</param-value>
</init-param>
    
```

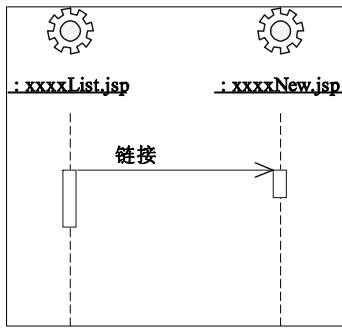
在 struts-config-xxx.xml 文件中还需再分别配置相应的 ActionForm 及处理页面的 Action 类和结果页面的 forward 信息：

```

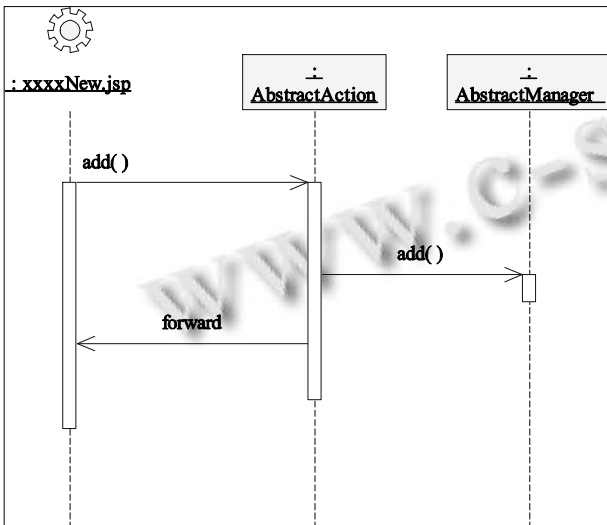
<form-bean name="xxxForm"
    type="edu.tibuu.xxx.form.xxxForm" />
<action
    attribute="xxxForm"
    name="xxxForm"
    path="/xxx "
    parameter="method"
    scope="request"
    type="edu.tibuu.xxx.action.xxxAction">
    <forward name="add" path="/xxx_edit.jsp"></forward>
    <forward name="edit" path="/xxx_edit.jsp"></forward>
    <forward name="view"
    path="/xxx_view.jsp"></forward>
    <forward name="list"
    path="/../common/general_query.do?query_id=xxx_query"></forward>
</action>
    
```

2.3.3 实现流程

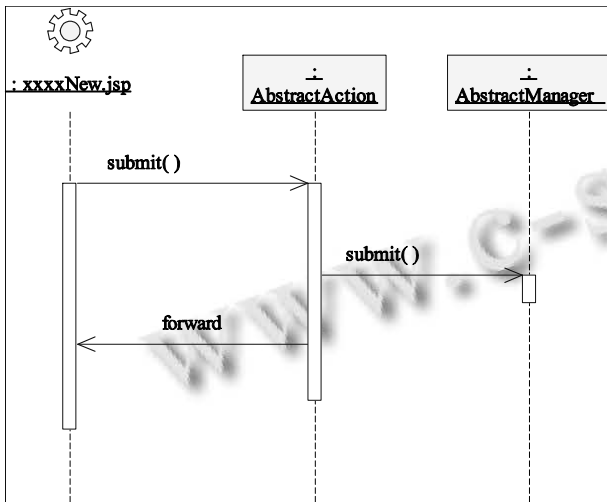
以系统的新增业务操作为例，增加页面一般从列表页面的按钮链接进入，增加页面不需要 Action，直接 forward 到相应的 jsp 页面。用户在增加页面填写提交后，由 Action 的 add 方法或 submit 方法处理添加操作，并根据具体的配置，返回到列表页面或增加页面。具体流程如图 6 所示。



(a) 进入增加页面



(b) 保存流程



(c) 提交流程

图 6 新增功能实现流程

2.4 权限管理设计

2.4.1 访问控制模型与策略

通过对系统的业务模型及功能模型尤其是用户角色

功能的分析,我们发现不同的用户所需使用系统的功能,或者说其应在系统中执行的业务操作和类型是不同的.比如,参与的教师应具有的是对自己职业生涯的规划、项目申报以及进度查询等功能;职能部门的相关人员则是对教师申报的相关项目按规则进行审批(参见表 1),而这些内容是不能由参与教师进行操作的.同时,一个用户可以拥有多个角色,亦可拥有其除所拥有角色之外而特有的功能.因此,系统需要对用户及角色进行分类管理并进行权限的分配.针对以上的系统需求,我们设计了一个基于角色的访问控制策略(Role Based Access Control, RBAC)^[6-10],即用户-角色-功能模块的权限系统.在该系统中,用户和角色之间、角色和功能模块之间、用户和功能模块之间均为多对多的关系,即:同一用户可以有多种角色,同一角色可以赋予多个用户;同一角色可对多个功能模块有访问权限,同一功能模块可赋权给多个角色;同一用户可以访问多个功能模块,同一功能模块可赋权给多个用户.用户和角色的权限具有继承关系,当用户隶属于某个角色时,其将具有该角色所拥有的所有权限,因此,一个用户所拥有的权限是其自身所赋予的权限与其所隶属角色的权限的并集.

2.4.2 数据库设计

要实现用户、权限与功能模块之间多对多的逻辑关系,同时考虑对用户、组织及功能模块本身的管理,与权限系统相关的数据库设计如表 2 及图 7 所示.

表 2 数据库表及功能说明

表名	功能与说明
USER_INFO	记录用户信息
ORG_INFO	记录机构信息以及机构关联关系
ROLE_INFO	记录角色信息
USER_PERMISSION	记录用户权限信息
USER_ROLE	记录用户角色关联关系
USER_ORG	记录用户机构关联关系
MODEL_PERMISSION	记录模块与权限对应关系
ROLE_PERMISSION	记录角色权限关联关系

2.4.3 授权管理与控制流程

在一个完整的授权系统中,还要有对关于系统所涉及对象的基本管理与授权操作.主要包括用户管理、组织机构管理、角色管理与模块管理等,它们的主要功能如下:

用户管理:增加、修改、删除用户,可以为用户指定部门,还可以对用户授于某种角色,还可以增加、修改、删除用户对模块的权限.同时还可以根据条件查询用户.

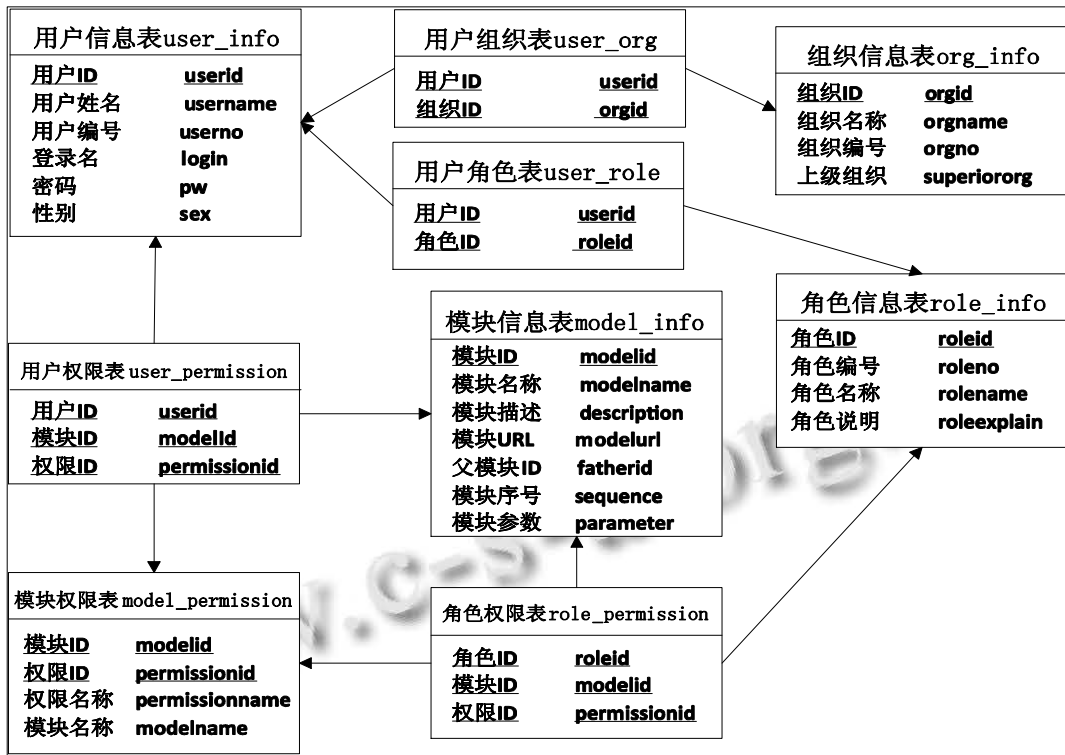


图 7 权限管理数据库设计

组织机构管理: 增加新组织、修改和删除已经存在的组织. 系统采用组织机构树的形式对组织进行维护和管理. 组织机构采用的是层次结构进行的存储, 每个结点都将设置一个字段来专门存储其所在的层次及上层结点编号, 这样便于将组织机构组织为一个树型进行维护.

角色管理: 增加角色、修改和删除原有的角色, 并给角色授权.

模块管理: 增加、修改和删除系统中所包含的功能模块及其所对应的功能权限点.

当用户登录系统时, 首先对用户身份进行验证及

取得用户权限. 系统取得用户所拥有的权限点集合, 并根据这些权限点决定用户是否有权访问特定的资源. 例如: 如果非管理人员访问管理模块, 会被拒绝访问 (或者不允许使用特定功能).

用户登录系统完成验证以后, 取得两部分权限的集合, 一是来自于用户自己的权限点, 二是来自于用户所属角色所拥有的权限点. 根据用户的功能权限集合动态生成菜单, 这样不同的用户授权后将会看到不同的系统界面. 图 8 所示为系统的权限设置界面; 图 9 所示分别为超级用户和普通教师角色登陆成功后所能看到的系统菜单与功能模块.



图 8 角色模块权限分配

另外, 为防止非法用户直接以 HTTP 方式在浏览器请求执行某页面, 在每个应用系统程序文件中还需

判断该用户是否登录并建立会话, 会话是否过期; 如 已建立会话, 就检查该用户是否有该页面的访问权限。



(a) 超级用户登录成功界面



(b) 普通教师登录成功界面

图 9 不同角色用户登录成功界面

3 结语

本文引入快速原型法设计并实现了基于 J2EE 的高校教师职业生涯规划系统, 保障了系统需求分析的正确性与有效性, 并大大提高了系统开发的整体质量与效率. 在系统实施上, 采用了集成 struts 框架与 hibernate 框架来实现 MVC 模式, 使得系统的可维护性、可扩展性强; 为了提高系统开发效率, 系统设计了一个通用框架模型, 将一些操作通过基类中实现, 这样在具体的模块功能页面中, 只须修改配置文件及简单的与模块相关的 jsp 页面即可; 在权限管理中, 通过基于 RBAC 的用户-角色-功能模块的权限策略使得系统具有灵活的资源访问权限配置, 亦保证了系统的安全性。

参考文献

- 1 罗新星, 朱名勋, 唐中君. 可信软件需求获取与分析研究综述及展望. 计算机应用研究, 2010, 27(10): 3617-3620.
- 2 常建军等. 计算机信息系统开发策略与实践. 微机发展,

1994, (5): 11-14.

- 3 妙旭华, 包理群, 李颖. 基于 J2EE 多层架构的重金属污染监管设计. 计算机应用与软件, 2013, 30(4): 128-130.
- 4 李小平等. 基于 J2EE 多层架构的 Web 开发框架研究. 计算机应用研究, 2008, 25(5): 1429-1431.
- 5 Holmes J. 孙燕等, 译. struts 程序员查询辞典. 北京: 中国铁道出版社, 2005.
- 6 于春生, 聂晶. 基于组和角色的工作流权限访问控制模型. 计算机应用, 2011, 31(3): 778-780.
- 7 陈琛等. 一种基于改进 RBAC 模型的 EIS 权限管理框架的研究与实现. 计算机应用研究, 2010, 27(10): 3855-3858.
- 8 呼和等. 基于 Web 服务的企业统一认证与授权系统. 计算机应用, 2011, 31(2): 576-580.
- 9 严骏等. MIS 中基于部门和角色的细粒度访问控制模型. 计算机应用, 2011, 31(2): 523-526.
- 10 杨柳, 危韧, 陈传波. 一种扩展型基于角色权限管理模型 (E-RBAC) 的研究. 计算机工程与科学, 2006, 28(9): 126-128.