

摘要

中小企业是我国国民经济的重要组成部分，创造了大量的就业机会和创新成果，是最活跃的创新基因。一直以来，中小企业的创新活动都有相当大的盲目性，政府作为公共资源的最大拥有者，需要引导中小企业的创新活动，为中小企业提供各种创新服务。随着计算机技术和信息技术的迅速发展并快速地渗透到社会生活的各个行业，各个领域都在发生着空前的变化，政府为中小企业提供的创新服务信息化就是其发展的结果，创新服务信息化能提高服务质量和运作效率，有助于企业提高自主创新能力，能推动产学研合作，解决企业融投资及人才引进等难题，提高企业的竞争力。

本论文基于某自治区科技厅中小企业创新服务平台系统，首先分析了中小企业创新服务的背景和意义，并介绍了国内外发展概况，然后对中小企业创新服务平台进行了深入的需求分析，通过 UML 建模对各个子系统进行了具体的分析，在此基础上，依据 J2EE 分层原则，从系统的可扩展性及可维护性出发，设计了系统的总体架构，并详细介绍了架构设计中表现层、业务层及持久层的设计，然后介绍了平台的功能规划，对系统进行了详细设计。在对系统数据的分析基础之上，设计了数据库的概念结构和逻辑结构。论文还讨论了系统的开发环境以及系统持久层、业务逻辑层和表现层的具体实现。论文最后总结了全文并对研究作了进一步的展望。

关键词： J2EE；创新服务；Struts； Hibernate； Spring

Abstract

SMEs (Small and medium enterprises) are an important part of the national economy of our country. And they create a large number of employment opportunities and innovation. They are the most active innovative genes, but the innovative activities of SMEs have considerable blindness all long, so the government, regarded as the largest owner of public resources, needs to guide the innovation activities of SMEs by providing SMEs with a variety of innovative services. With the rapid development of the computer technology and information technology, rapidly penetrating into all sectors of social life, which has resulted in unprecedented changes in all areas, so the innovative services provided by the government for SMEs informationized are the result of the changes. And that innovative services informationized is to be capable of improving service quality and operational efficiency, helping enterprises to improve their capability of independent innovation, promoting research cooperation for the solution to enterprises' finance such as investment and the introduction of talents and promoting the competitiveness of enterprises.

In this paper, based on innovative services platform for SMEs system of Science and Technology Department of an Autonomous Region. Firstly, it makes an analysis of the background and significance of SMEs' innovative services. Secondly, it describes the overview of the development at home and abroad. Thirdly, it conducts an analysis of SMEs' innovative service platform for in-depth needs. Fourthly, it makes a specific analysis of each subsystem through UML modeling. On the basis of four analyses, therefore, in accordance with J2EE classified principle, from the system scalability and maintainability, it designs the overall structure and gives a detailed design of the representation layer, business layer and persistence layer in structural design and then introduces a platform for the function of planning and system design in details.

Based on the analysis of the data on the system, the paper also makes a design of the connectional structure and logical structure of the database. And what is more, it makes a discussion on the development of the system environment and the specific achievement of system persistence layer, business logic layer and the representation layer. Finally, the text summarizes the outlook for further research.

Keywords: J2EE; Innovation Service; Struts; Hibernate; Spring

西南交通大学

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权西南交通大学可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复印手段保存和汇编本学位论文。

本学位论文属于

1. 保密□，在 年解密后适用本授权书；
2. 不保密☒，使用本授权书。

（请在以上方框内打“√”）

学位论文作者签名：黄义平
日期：2009.6.10

指导老师签名：张强
日期：2009.6.10

西南交通大学学位论文创新性声明

本人郑重声明:所呈交的学位论文,是在导师指导下独立进行研究工作所得的成果。除文中已经注明引用的内容外,本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体,均已在文中作了明确的说明。本人完全意识到本声明的法律结果由本人承担。

本学位论文的主要创新点如下:

对中小企业创新服务平台作了功能分析和组织分析,并结合作者在攻读硕士期间对基于 J2EE 架构的企业级应用系统的设计、实现方面的研究,通过 UML 技术对中小企业创新服务平台需求进行建模,在对 Struts, Spring, Hibernate 技术研究基础上,完成了基于这三个框架的平台架构设计,并实现了具有高度可扩展性,可维护性的创新服务平台系统。

学位论文作者签名:黄义平
日期:2009.6.10

第 1 章 绪 论

1.1 课题的背景

1.1.1 我国中小企业创新发展现状

中小企业是我国国民经济的重要组成部分，是推动国民经济发展、构造市场经济主体和促进社会稳定的基础力量。尤其在当前确保我国经济增长，缓解就业压力和优化经济结构等方面发挥着越来越重要的作用。随着社会主义市场经济体制的建立和逐步完善，中小企业的改革和发展日益被人们所关注，并逐步提到国家重要议事日程上来。2000年的中央经济工作会议，国家就明确提出了要高度重视发展中小企业的问题，要求为各类中小企业创造良好的政策环境并加快加强中小企业自身的创新发展。而且为了改善中小企业经营环境，促进中小企业健康发展，扩大城乡就业，发挥中小企业在国民经济和社会发展中的重要作用，国家颁布了《中小企业促进法》。然而在宏观经济环境发生变化的情况下，中小企业的发展仍然面临着很多困境，如：资金困境、管理困境、技术困境、人力困境和市场困境等状况，存在着如高投入、高消耗、产品低端、产业技术水准相对较低等问题，而且中小企业普遍盈利能力差、倒闭率高以及生存期限短。究其根源既有外部环境的原因，也有企业自身创新能力差的原因^[1,2]。

分析我国中小企业创新现状，可以得出目前我国中小企业创新具有以下特点：

- (1) 从创新目标来看，重产品、轻服务；重生产、轻经销。
- (2) 从创新方式来看，各自为战居多，合作共赢不足。
- (3) 资金投入规模偏小，自主创新能力较弱。
- (4) 信息来源以业内为主，科研和中介机构参与企业创新的程度较低。
- (5) 质量创新最受认可，创新领域有明显的外向性。
- (6) 组织架构方面：重内轻外。
- (7) 人才是阻碍创新的最主要因素^[3]。

造成我国中小企业创新这种现状的因素是多方面的，既有宏观经济环境方面的因素，也有中小企业自身的因素。归纳起来主要有以下几个方面：

(1) 创新资金投入严重不足：企业创新特别是技术创新需要投入大量的经费，国际上一般认为，研究开发经费占销售额 1% 的企业难以生存，占 2% 可以维持，占 5% 以上才有竞争力。但是我国中小企业的研究开发经费支出远远低于这个水平，我国中小企业研发经费大约仅为美国中小企业的 1.2%，为日本的 1.1%^[2]。造成这种局面的原因，一是中小企业普遍面临的是自有资金不足，二是融资难度大，三是政府激励中小企业创新的改革措施跟不上，政府支持、扶持的力度不够。

(2) 管理水平低，创新人才缺乏：我国的中小企业管理相对来说比较落后，企业多采用家族式的管理方式，领导权多集中在极少数人的手中，组织结构不合理，规章制度不健全，企业运作不规范，由于企业管理及创新人才的缺乏，使得先进的管理理念在企业得不到推广应用，影响了企业创新能力的提高，阻碍了企业发展。

(3) 企业市场意识以及获取市场信息能力不足：多数中小企业市场信息意识比较淡薄，信息面窄，收集、分析信息的手段简单，利用信息的能力不足，由于缺乏准确全面的市场信息，缺乏先进的技术、管理信息，企业创新成了空话，使得这些企业对市场形势的变化反映滞后，对市场需求、先进技术、先进管理不敏感，导致了我国企业在创新决策上存在着一定程度的盲目性，并缺少动力，严重影响了中小企业的发展。

(4) 企业文化落后：企业的生命力来源于企业的文化，企业文化是企业的精神和灵魂，是企业的大动脉和神经网络，企业文化强调的是企业精神、全体员工共同的价值取向及在此基础上形成的凝聚力、向心力，创新力，能给企业带来的巨大财富。然而我国中小企业生存时间短、发展速度慢，重要的原因就是企业文化落后，员工没有创新的源泉与动力。

为了促进中小企业创新发展，政府需充分发挥组织和推动作用，为企业创新发展提供良好的环境。另外，充分发挥企业的主体能动作用，多渠道的扫清中小企业创新障碍。充分调动中介服务的积极性，作为中小企业创新的助推器，鼓励中小企业与科研机构、高等院校结合，建立长期合作关系^[2]。

1.1.2 促进中小企业创新发展得到各国高度重视

中小企业是各国经济健康发展的重要基础，也是提高国家综合国力与国际竞争力的重要生力军。面对国际竞争，各国都不遗余力的采用各种方式来

对中小型企业扶持与培育。

首先,各国都制定了各种政策法规以保障中小企业的合法权益,促进企业的创新。例如美国先后颁布了《史蒂文森——怀特勒创新法》、《中小企业创新发展法》等等。为了更好地发挥小企业在创新中的作用,美国政府特地制定了“小企业创新研究计划”和“小企业技术转让计划”^[9]。日本科技型中小企业法律支持体系由宪法、科学技术基本法、技术创新服务单行法、行政法规、部门规章、地方法规和地方规章组成,如《小企业基本法》、《禁止垄断法》、《中小企业振兴资金助成法》、《中小企业现代化促进法》等等,目前又制定了《加强中小企业技术创新减税法》等十几项法律^[9]。

其次,各国都设立了专门的管理机构来处理与中小企业发展的相关事务。例如英国工业部内设立了小企业局,德国在经济部下设了中小企业局,法国政府成立了专门扶持中小企业技术创新的机构,即国家技术交流转让中心。日本政府采取主动干预中小企业政策,早在1948年就在通产省设立中小企业厅,作为管理中小企业的最高机构。

另外,许多国家政府尽可能的为中小企业提供如人才培养,中介,融资等社会服务。例如德国政府采用各种方法如资助鼓励各类中介机构参与中小企业服务,法国政府要求政府及公立机构加强对中小企业的信息服务。英国教育部规定学校必须提供小公司管理方面的培训,并且学生在毕业前要在小公司进行15个星期的商业培训^[9]。

1.2 中小企业创新服务平台的研究和发展情况

在我国,为了促进中小企业创新发展,各地区中小企业管理部门在中小企业公共服务平台建设方面作了大量的探索和实践,并取得了很大的成效。国家相关部门也大力引导和推动中小企业创新服务平台的建设,例如国家发改委为加强引导,合理、有效地组织和调动社会资源,建设相关中小企业公共服务平台,促进中小企业的创新发展,就下发了申报中小企业服务平台的相关通知,对各区/platform建设予以资金和技术等方面的扶持。目前,各地区开通的中小企业创新服务平台大部分都包含了如政策法规、通告通知、咨询服务、投资融资服务、求职招聘、项目申报等服务,很多地方政府还与大学,科研机构合力打造中小企业服务平台,例如扬州市人民政府就与南京大学一起共建科技创新服务平台。由于国家相关中小企业创新服务平台很难兼顾各地区的特有行业,而各地区的创新服务平台又缺乏统一规划,因此涉及各具

体行业的中小企业创新服务平台也随之出现。从当前国内中小企业创新服务平台建设情况来看,产业集群或产业相对集中的区域平台建设要完善一些,而且中小企业参与的程度也比较高。总的来说,由于国家,地区相关部门,包括各个中小型企业越来越意识到创新的重要性,

由于我地区经济、文化发展的差异,造成各地区在中小企业创新服务平台的建设力度不一,各地区平台发展程度也不尽相同。同时国内存在着的某些问题也一直制约着中小企业创新发展,主要包括主客观两个方面,具体体现在以下几个方面:

- (1) 对中小企业创新范围、地位认识不足。
- (2) 标准建设滞后。
- (3) 缺乏统一规划,网络分离,互联性差,各地区很难统一。
- (4) 社会整体信息化水平低,基础设施尚不完备^[7]。

但是在我国在努力营造中小企业创新环境的发展势头还是良好的,服务平台的整体建设速度正在加快。

1.3 课题研究内容

本文研究的中小企业创新服务平台,用于某自治区科技厅生产力促进中心为促进区内中小企业创新发展提供各种服务。作者在实习期间,参与了该平台公共技术服务,投资融资服务,行业科技,人才引进及培训,商务信息交流等子系统的设计与编码实现过程。论文结合作者的实践,主要完成以下工作:

- 系统开发框架的研究。在本文中,研究了 Struts, Spring, Hibernate 框架,并基于这几个框架对系统进行了设计。

- 对中小企业创新服务平台公共技术服务,投资融资服务,行业科技,创新成果转化,人才引进及培训,商务信息交流等业务进行需求分析和建模,并分析了系统的非功能需求。

- 在需求分析的基础之上,进行了系统的架构设计,功能设计以及数据库设计,并结合系统设计,对系统实现进行详细说明。

1.4 论文结构

本论文分为五个章节,内容如下:

第一章:绪论。本章主要介绍课题的研究背景,中小企业创新服务平台

的研究和发展情况，介绍了课题的研究内容，最后介绍了论文结构。

第二章：需求分析。本章主要对科技厅中小企业创新平台政务系统的需求进行了分析。本章首先介绍了业务目标，然后对各子功能进行了详细的介绍，最后介绍了非功能性需求。

第三章：系统设计。本章首先介绍了系统架构设计，然后对架构各层中使用到的框架技术进行了研究，然后介绍了整个系统的总体功能设计并分析了系统主要功能详细设计，最后介绍了系统数据库设计。

第四章：系统的实现与讨论。本章主要系统的详细实现。主要从表现层的实现，业务逻辑层的实现，数据持久层的实现等几个方面展开。

第五章：总结与展望。对论文工作进行了总结，提出未来工作展望。

第 2 章 需求分析

2.1 业务目标概述

中小企业创新服务平台是为了通过积极动员和整合某自治区各类社会资源,建设一个以网络平台为基础、服务体系为支撑、共享机制为保障的服务平台,实现中小企业创新科技服务的资源整合、系统集成、业务联动、共享服务,帮助区内中小企业提高自主创新能力,推动产学研合作,解决企业融资、引进人才和培训人才难的问题,全面提高中小企业对外开放水平和企业的核心竞争力,建立自治区内中小企业创新服务平台运行机制和服务体系,使创新服务落到实处,切实为区内的中小企业服务。

本文拟研究的中小企业创新服务平台为了使中小企业创新活动的顺利进行,需为中小企业提供以下服务:

(1) 公共技术服务:通过平台给企业提供国家和地方政府相关政策法规,使得企业能够规范经营活动,为保障企业的合法权益提供依据;提供如科技期刊、图书、报告、论文、会议资料和声像等文献库供用户检索,方便用户查阅参考;为了促进科技资源共享,平台建设创新资源共享库,企业检索共享库,使用其他企业和个人共享的各种资源,降低企业研发成本;平台技术标准服务,使得企业能够参考相关标准指导自己的创新活动;另外,为了解决企业的技术难点,管理机制,提供一个企业与专家之间的交流平台,企业能够向专家咨询相关问题。

(2) 投资融资服务:为了方便企业的投资融资,使企业融投资能够良好运转以及企业和个人的资本能更好的运作,提供一个投资融资的平台,企业能够通过平台了解最新融投资动态,指导融投资行为。

(3) 行业科技:使企业能够检索各行业的产品信息,各行业标准以及行业新闻,提高企业对各行业的认识,为企业的创新活动打下基础。

(4) 人才引进及培训:开展培训活动,提高企业竞争力。搭建人才供需平台,使企业能够招聘到合适的人才。

(5) 创新成果转化:使企业能将科技成果最终转化为生产力,提供平台展示各科技成果。筛选各中介机构,供用户选择。另外使企业了解到最新的

创新动态, 指导企业创新成果转化活动。

(6) 商务信息交流: 挖掘和利用供需双方供需市场, 使企业能够捕捉到各种商机。

中小企业创新服务平台由西藏自治区科技厅生产力促进中心维护和管理。西藏自治区生产力促进中心是隶属于科技厅管理的处级事业单位, 下设综合部、信息服务部和市场部。其中信息服务部工作人员负责维护中心信息网站, 负责建立中心人才专家库、企业产品库、成果库等, 为企业提供信息、培训、政策、信息咨询、项目包装、技术与新产品引进等方面的综合服务; 市场职能部工作人员通过平台来了解区内中小企业需求, 组织专家对中小企业进行诊断, 为企业与政府和科研院所的合作牵线搭桥, 为企业开展质量认证提供咨询与服务, 管理人才交流市场, 组织企业参加有关的交流与交易等活动。

2.2 业务需求分析

2.2.1 公共技术服务管理

公共技术服务在中小企业创新服务平台中是一个比较重要的部分。根据需求分析, 平台需要给企业和个人提供以下服务:

- 政策法规服务: 政策法规是企业区内合法经营需要遵守的规范, 是企业的合法权利得到保护的依据, 也是政府为企业提供服务的依据。平台提供国家以及地方政府的科技政策、知识产权政策、相关法律法规、优惠扶持政策, 以方便企业掌握政策导向, 并为企业提供政府科技主管部门的办事流程文件和规定, 充分发挥政府政策的引导作用。其中政策法规初始包括国家法规、自治区法规以及其他地区法规。随着各类法规的变更, 平台提供的政策法规文章应该能够由信息部工作人员更新, 政策法规的类型也可以由信息部工作人员进行配置。通过上述分析, 政策法规服务主要两种参与者: 一般用户与信息部工作人员。其中一般用户主要涉及到政策法规的浏览, 而信息部工作人员则负责管理各类型法规文章以及法律法规类型。

政策法规服务的功能模型如图 2-1 所示:

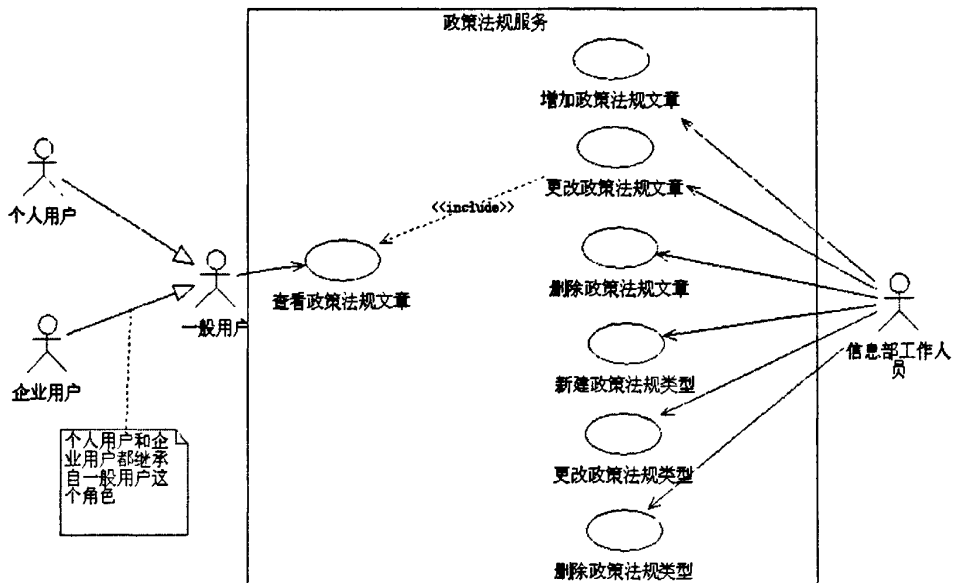


图 2-1 公共技术服务政策法规功能模型图

● 科技文献服务：为自治区科技界和广大科研人员提供快捷、高效、优质的文献信息服务，使中小企业服务服务平台成为为各企业提供高效科技服务的重要载体。科技文献服务以数据库的形式提供科技期刊、图书、报告、论文、会议资料、声像等科技文献检索服务。检索时需要提供如标题、作者、出版社、关键字以及出版年限等多种检索方式。数据库资源基于原科技厅所收集的各种文档，视频，会议资料，科技期刊和自治区内相关论文等。科技文献服务的维护主要由信息部工作人员来完成。这一部分参与者主要是信息部工作人员与一般用户。信息部工作人员主要完成各种科技文献的管理，而一般用户通过不同的检索方式来获得相关信息。

● 技术标准服务：以国家标准文献资源为基础，整合主要国际组织、行业、部门与地方标准文献，提供技术标准数据库，形成完善的标准文献的检索与传递公共服务系统。技术标准服务在建设初期包括国家畜牧业标准，世界其他相关标准和国家标准。具有对应权限的信息部工作人员在后台能够灵活的增加、删除和修改标准类别，并能对增加和更新各种标准。

● 热点调查服务：在一个信息系统中，经常会有各种热点调查提供给用户参加，通过热点调查，平台能够了解用户的真实观点和看法。具有权限的信息部工作人员能够在后台添加和编辑热点调查内容，选项等。用户访问平台时，能以较直接的方式反馈给用户。

● 创新资源共享平台：通过提供区域内的创新资源信息，制定创新资源的共享机制，促进科技资源共享，提高创新资源的利用率，降低企业研发成本，为发挥科技资源的社会化作用，努力提高区内各种创新资源的的使用效益。该平台包括仪器设备共享、实验材料共享、测试设备共享、科研数据共享、信息资料共享、创新方法培训等服务内容。其中仪器设备共享检索内容包括仪器名称、产品编号、产品简介以及包装与规格等；测试设备包括仪器名称、仪器图片、功能应用范围、开放机时安排、依托单位、操作员、联系电话、备注以及预约功能；创新方法培训需要包括培训时间、单位、联系人、联系电话、培训地址以及预约功能等。企业或者个人能够提出各类型创新资源共享申请，经过信息部工作人员审核后能够在前台进行发布。

根据以上分析，可知创新资源共享平台的参与者主要有：一般用户与信息部工作人员。一般用户主要涉及到的操作有检索各种创新资源及申请共享自己的设备。信息部工作人员涉及到的操作有审核设备共享申请，增加资源共享信息及修改资源共享信息。创新资源共享平台的功能模型图如图 2-2 所示：

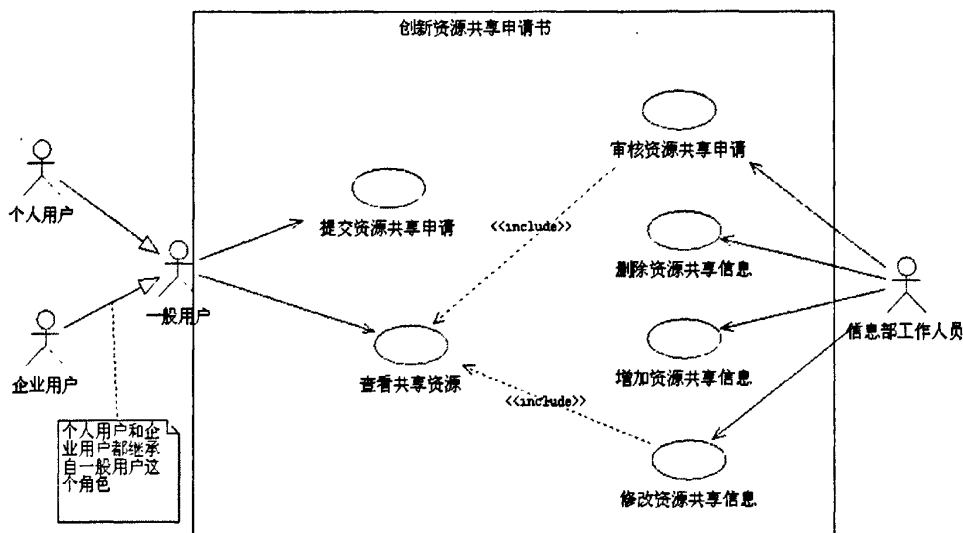


图 2-2 创新资源共享功能模型图

● 咨询服务：建设自治区专家数据库，为创新活动提供技术咨询。围绕区域的经济社会发展的重点、热点和难点问题，充分发挥专家优势，开展各种形式的决策咨询活动，为促进决策科学化，围绕区域科技创新体系建设、企业研发管理机制、企业技术难点问题等重大问题献计献策。

企业或者个人向专家提出咨询服务步骤包括：1，用户查阅申请流程，填写用户情况及咨询问题 2，信息部工作人员将受理咨询审核 3，审核通过后信息部工作人员作需求登记汇总 3，专家回复用户咨询 4，企业和个人提交意见反馈意见 5，信息部工作人员查看用户意见反馈 6，将用户反馈意见资料归档。咨询服务活动图如图 2-3 所示：

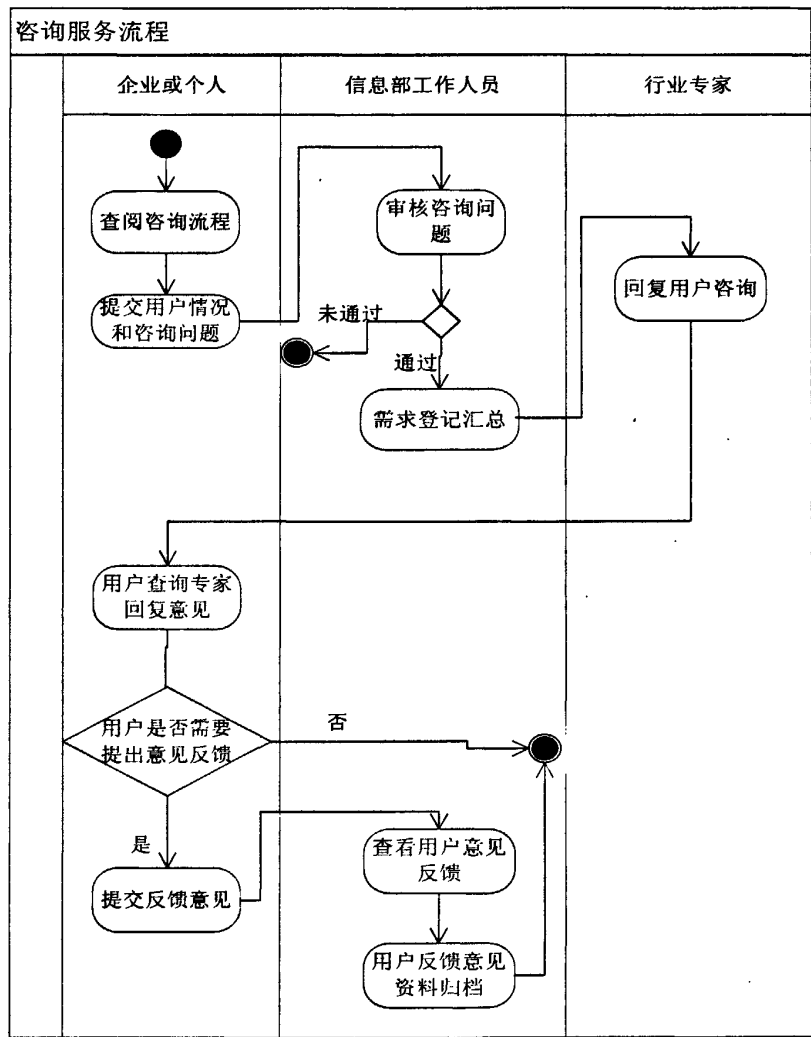


图 2-3 咨询服务活动图

根据以上分析，咨询服务涉及到的参与者包括：一般用户，行业专家以及信息部工作人员。其中一般用户包括查询咨询服务流程，提交自己的咨询申请，查看行业专家的咨询回复以及回复意见反馈；行业专家能够查看用户的咨询问题，回复企业用户或个人用户的咨询，查看用户的意见反馈；信息

部工作人员需要审核用户咨询问题，咨询需求登记，查看用户的反馈意见以及反馈意见归档。咨询服务功能模型如图 2-4 所示：

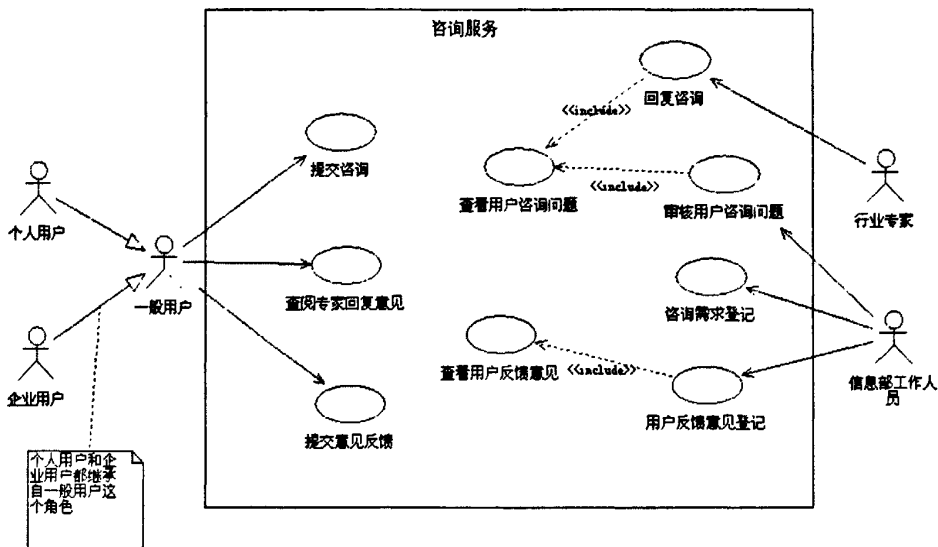


图 2-4 咨询服务功能模型图

2.2.2 投资融资服务管理

● **融投资信息服务：**融投资信息服务是给企业和个人提供一个平台使得其资本能够良好的运作。获得相关权限的用户和个人能够发布融资投资信息，投资信息应该包括所属行业、实施地区、合作方式、投资总额以及详细的描述。融资信息应该包括行业、地区、投资总额、善缺资金、公司简介以及详细描述。发布相关投资融资信息后，市场部工作人员可以对信息审核、编辑和删除。另外获得相应权限的企业和个人能够查看各种融投资信息。

● **公共信息交流平台：**为了使得融投资信息的顺利进行，用户可以发布公共信息，公共信息包括标题、分类标签、联系人、联系电话、联系地址以及详细描述。具有权限的用户能够浏览各行业的公共信息。另外市场部工作人员可以对用户发表的公共信息进行审核、编辑和删除。

● **融投资新闻管理：**平台提供相关融投资新闻，使得企业和个人能够了解最新融投资动态等。融投资新闻管理主要是市场部工作人员完成。市场部工作人员可以编辑新闻、增加和删除新闻等。

投资融资服务子系统用例图如图 2-5 所示：

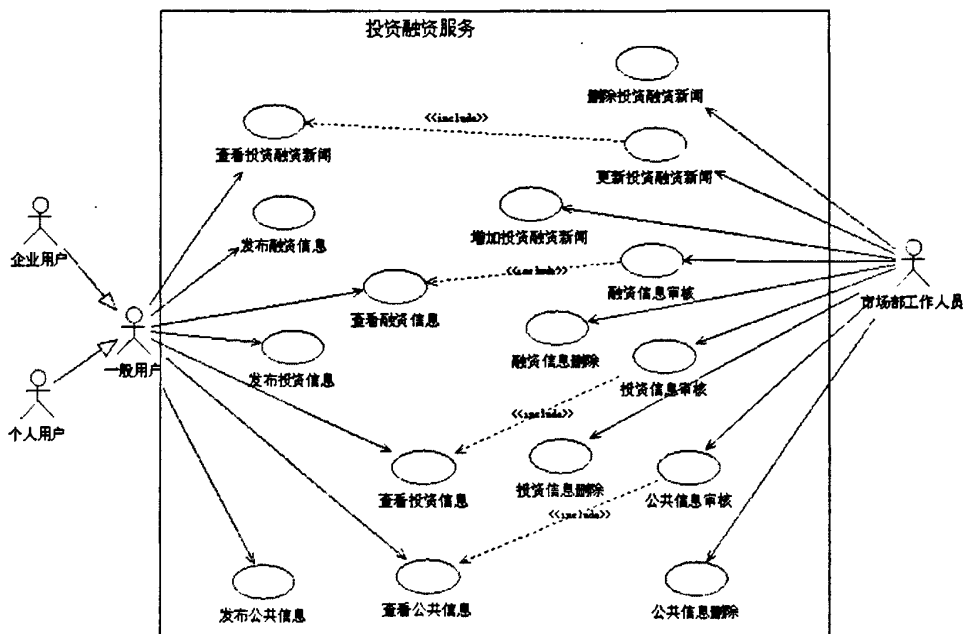


图 2-5 投资融资功能模型图

2.2.3 行业科技服务管理

行业科技服务主要是根据自治区的实际情况，对各行业的科学技术数据和行业信息进行整理和分类，建设起具有自治区行业特色的子系统，例如藏医药、旅游、手工业、农牧业等以供用户检索。行业科技服务主要包括以下几个部分：

- **行业数据库管理：**信息部工作人员在原来遗留数据(主要以 excel 表的形式存在)的基础上经过整理及筛选建立新的行业数据库，这些数据库可以为其它子系统共享。另外，信息部工作人员能够添加、修改和删除新的行业数据。在项目初始阶段，先建立藏药数据库。经过授权后，合法用户可以检索行业数据库。行业数据库的检索按模糊搜索进行。

- **行业标准建设：**信息部工作人员根据自治区行业特色，能够增加和删除自治区独有行业的行业标准，并进行推广。用户可以参照有关标准来执行自己的创新活动。

- **行业平台新闻：**信息部工作人员在后台添加行业相关的新闻，用户能够在前台查看。信息部工作人员也能够进行编辑和删除已有的平台新闻，

还应该可以新建, 修改和删除新闻类型。另外, 平台新闻需要提供给前台用户灵活的检索方式如按标题、新闻类别和发布日期等。

行业科技子系统功能模型图如图 2-6 所示:

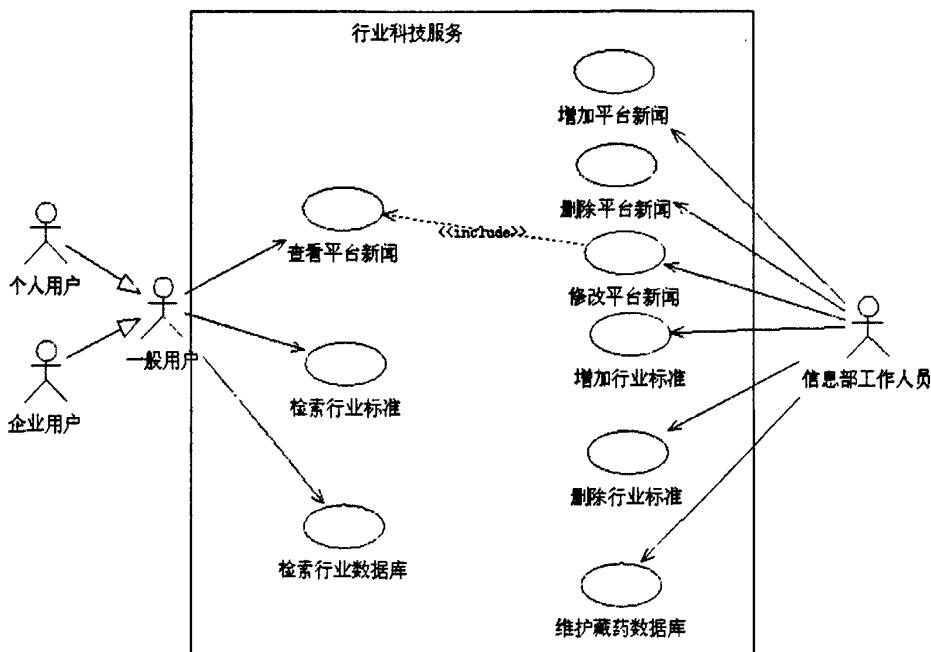


图 2-6 行业科技子系统功能模型图

2.2.4 人才引进及培训服务管理

人才引进及培训可以分为以下业务: 平台人才相关新闻管理, 人才培训管理, 人才需求信息管理, 平台人才引进机制信息管理, 公司招聘信息管理, 个人求职信息管理。

下面对各业务作详细的概述:

- 平台人才相关新闻: 市场部工作人员获取相关权限后, 在后台可以添加、删除和编辑平台人才类相关新闻。前台用户可以检索相关人才新闻介绍。

- 人才培训管理: 为了提高企业和个人创新能力、降低企业耗能、提高企业管理水平和增强企业的竞争能力。平台利用专家资源, 不定期开展各种技术培训, 包括科技创新的培训等。人才培训分为需要申请和不需要申请两种, 需要申请时企业和个人参加培训步骤是: 1, 市场部工作人员发布技

术培训通知,其中技术培训包括时间、地点、培训地址、人数上限、培训类别、主讲人、培训对象要求和详细描述信息 2,用户查看培训通知 3,用户如果满足培训申请条件则提交参加培训申请 4,市场部工作人员审核培训申请 4,查看人数是否超过人数上限 5,发出邀请函分配标识号 6,提交确认信息 6,市场部工作人员将确认信息归档。其申请参加培训流程图如图 2-7 所示:

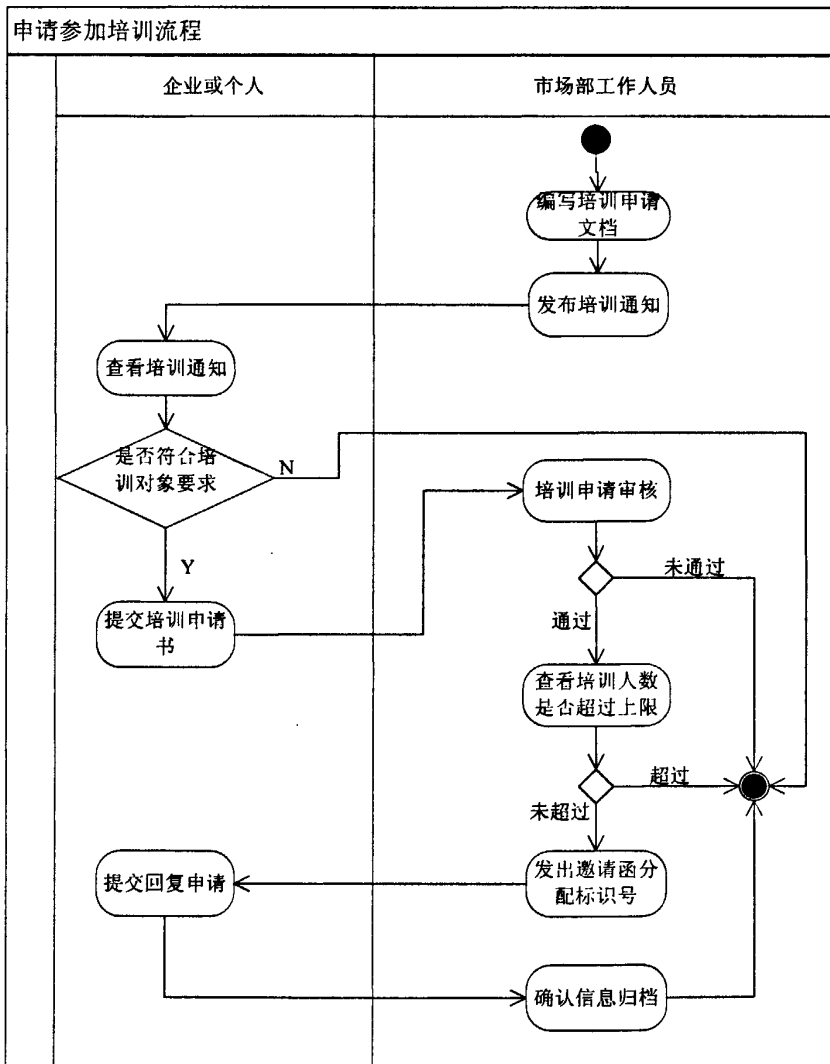


图 2-7 用户申请参加培训流程图

● 公司招聘信息管理: 企业用户在获得对应权限后能够在创新服务平台上发布企业招聘信息, 招聘信息包括招聘名称、职位名称、职位类别(全

职、兼职或者临时)、学历要求、需求人数、工作地点、职位工资、职位要求、职位说明、截止日期以及应交材料等。同时企业用户在获得权限后能够查看其他企业发布的招聘信息。另外在提交招聘信息后可以修改和删除自己发布的招聘信息。企业用户发布的招聘信息通过市场部工作人员审核后才能在前台显示,市场部工作人员还能够检索和删除企业用户发布的招聘信息。

- 个人求职信息管理: 个人用户在获得对应权限后能够在平台上面发布个人求职信息,求职信息包括求职意向、工资要求、期望工作地点、求职名称、职位类别、工作职位(说明行业)以及自我描述等。另外个人用户能够填写个人简历,填写个人简历后能够修改和删除自己的简历。同样填写求职信息后也可以修改或者删除自己的求职信息。另外在获得权限后,个人用户和企业用户能够查看其他人的求职信息,查看其他人的求职信息时,如果已填写个人简历,还可以浏览他人的求职简历。个人求职信息在后台市场部工作人员审核后方可在前台显示。市场部工作人员还可以检索和删除个人求职信息以及个人简历等。

- 人才需求信息管理: 市场部工作人员也可以发布自治区内人才需求相关信息,这样可以对仅注册企业用户发布招聘信息形成补充。人才需求信息由市场部工作人员在后台维护,包括人才需求信息的增加、删除、检索和修改等操作。企业用户和个人用户在获得相应授权后能够在前台检索人才需求信息。

- 人才引进机制信息管理: 市场部工作人员可以发布自治区内人才引进的相关规定、人才引进策略及人事仲裁相关信息。人才引进机制信息由市场部工作人员在后台维护。企业用户和个人用户在获得相应授权后能够在前台检索人才引进机制信息。

根据以上分析,人才引进及培训子系统主要有三个参与者:个人用户、企业用户和市场部工作人员。个人用户参与的用例有:检索人才相关新闻及人才引进机制、查看培训通知、提交接受培训申请、提交培训回复、填写个人求职信息、修改个人求职信息、删除个人求职信息、填写个人简历、修改个人简历、删除个人简历、检索个人求职信息、检索企业招聘信息以及检索平台人才需求信息等;企业用户参与的用例包括:检索人才相关新闻及人才引进机制、查看培训通知、提交接受培训申请、提交培训回复、填写招聘信息、修改招聘信息、删除招聘信息、检索个人求职信息、检索企业招聘信息以及检索人才需求信息;市场部工作人员参与的用例有:增加人才相关新闻、

修改人才相关新闻、删除人才相关新闻、发布人才引进机制、发布培训通知、审核用户培训申请、用户培训确认归档、审核个人求职信息、审核企业招聘信息以及对人才需求信息进行管理。由于人才引进及培训涉及到的用例很多,因此下面人才引进及培训功能模型图仅列出相关核心用例,如图 2-8 所示

为多个类别,用户检索科技转化相关政策法规时,应该提供如题目,类别以及按类别等的检索方式。

另外,平台还需要提供区内可靠的中介机构供给用户选择,市场部工作人员能够添加新的中介机构、更新和删除已有的中介机构信息。中介机构信息应包括单位名称、联系人、联系电话以及中介机构单位类别。用户在搜索中介机构时,也应该提供多种检索方式。

除此之外,平台还需要提供科技动态服务。科技动态以文章的形式供用户查看。科技动态文章可以由信息服务部工作人员在后台进行管理。

根据上述需求分析,可以得到创新成果转化服务功能模型如图 2-9 所示:

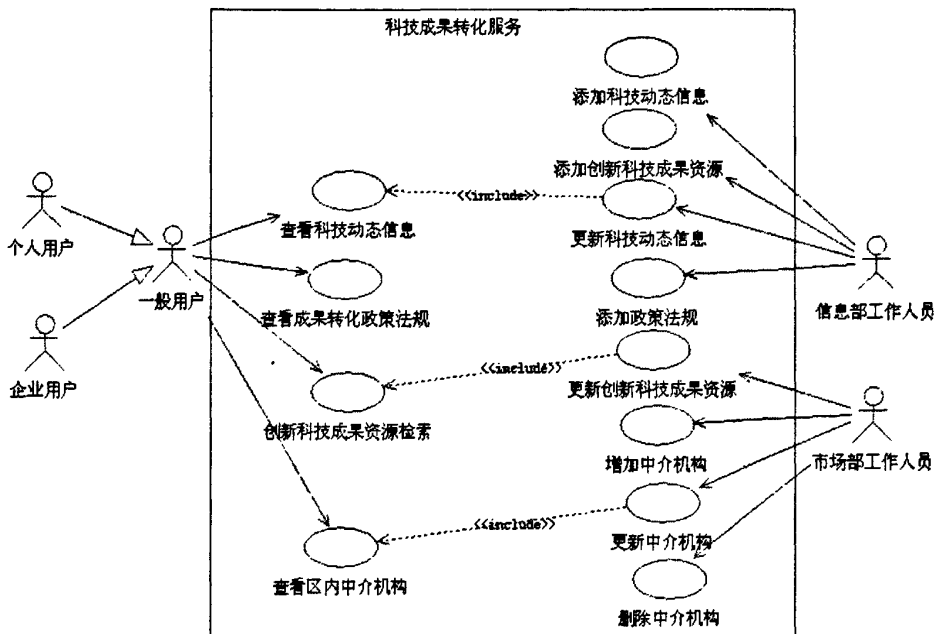


图 2-9 创新成果转化服务管理功能模型图

2.2.6 商务信息交流服务管理

为了使得用户之间的交流更加直接,从挖掘和利用供需双方市场入手,建立产业界商务信息稳定的交流渠道,使得能够企业捕捉对有利的经贸信息,挖掘潜在的商机,因此在平台上建设类似一个小型论坛的商务信息交流交流平台,建立产品供给信息,产品求购信息等版块供用户进行信息交流,而且随着平台的逐步完善,信息服务部工作人员还可以添加新的版块。注册

用户能够在各版块中发布供给求购信息，能够回复供给求购信息。各种信息在平台中通过信息服务部工作人员审核后能够在前台显示。

根据以上分析，可以得到商务信息交流服务管理的功能模型图如图 2-10 所示：

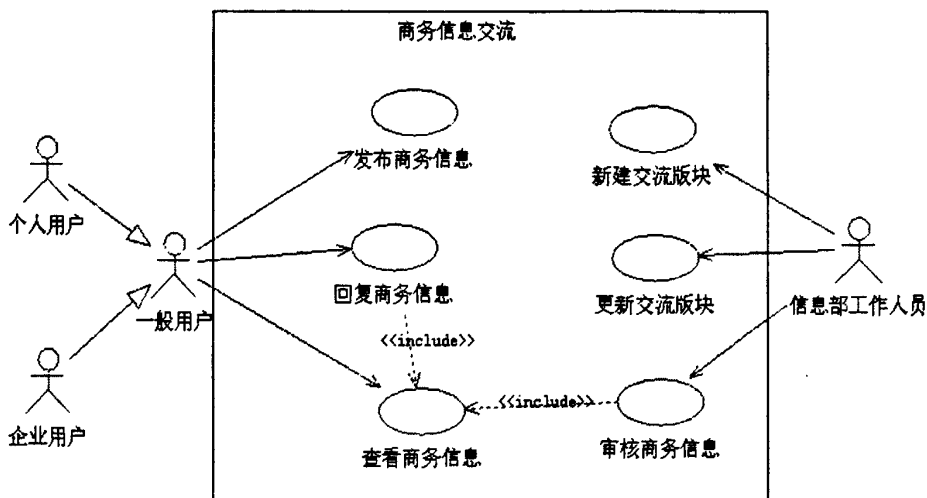


图 2-10 商务信息交流服务功能模型图

2.3 非功能需求

经过分析，平台需要满足的性能包括以下几点：

- 安全性：

中小企业创新服务平台是一个基于网络的电子政务系统，所以安全性必须放在第一位。系统在平台架构和数据库设计时需要充分考虑其安全性，要设计合理的权限管理机制，在信息处理的各个环节需要考虑用户权限，防止非法用户访问机密数据。

- 易用性：

由于该自治区信息化水平比较低，相关使用人员计算机水平参差不齐，因此系统要具有人性化，系统中信息显示和组织应比较合理，用户能较容易地发现所需的信息和执行的操作入口，另外，系统需要提供给用户完整的业务流程导航，使用户能够容易操作，而且当用户执行完一定操作之后，给用

户提供反馈信息。

- 可扩展性：

系统应该采用合理的架构设计，当系统出现新业务需求或者变更现有业务时，对现有系统的影响应尽可能的小。

第 3 章 系统设计

上一章对系统的需求进行了较高层次的分析,对各个业务进行了详细的分析。本章对系统进行详细设计,首先介绍了系统架构设计,然后对架构各层使用框架进行了介绍,再分析了系统总体功能设计,然后介绍系统功能详细设计,最后介绍了系统的数据库设计。

3.1 系统架构设计

在面向对象设计中,松耦合是 OO 设计的一个重要的追求目标,解耦性也因此成为衡量 J2EE 系统质量的首要标准,同时,在系统设计时需要兼顾系统的可伸缩性,运行性能以及开发效率。一个典型的 J2EE 系统可以划分为三个层次:表现层、业务逻辑层和持久层。表现层是一个典型的 Web 应用的末端,通常包含表示代码、用户交互 UI、数据验证等。在这一层中,系统管理用户的请求和响应,调用业务逻辑层或其他上游的处理,并将处理结果提供给用户。业务逻辑层是由大量组件组成的,也可称为组件层,这一层是系统的重点和核心,包含业务规则处理代码,负责处理应用的业务逻辑,该层处于表现层与持久层之间,在数据交换中起到承上启下的作用。持久层是一个相对独立的逻辑层面,包含数据处理代码和数据存储代码,这一层专注于数据持久化逻辑的实现,在典型的 J2EE 系统中,该层主要封装系统对数据库的访问,提供对业务逻辑层的支持。这三层中的每一层都有各自的任务,功能上能够清晰的区分,各层之间相互分离,它们之间通过接口来进行通信。

SUN 公司在推出 J2EE 标准时,分别规定了这三层的标准实现:表现层使用 Jsp/Servlet 技术,业务逻辑层使用 EJB 中的会话 Bean (Session Bean),持久层使用实体 Bean(Entity Bean)。然而随着应用系统的日益复杂,这三层标准实现越来越难以满足需求,如表现层设计稍有不慎就会使得页面结构混乱,而常规的 EJB 开发会使系统代码变得相当复杂,复杂的规格说明往往使开发人员力不从心,而且调试 EJB 程序非常困难,维护 EJB 系统也相当困难。随着 JAVA 技术的发展,各种框架技术随之而生。

在设计模式中, Gamma 等人为框架给出了一个明确定义:“框架

(Framework) 是一组协同工作的类, 它们为特定类型的软件构筑了一个可重用的设计”^[11], 框架强调的是软件设计的重用性和系统的可扩充性。它是为了追求重用所带来的价值最大化, 而将被重用最多的部分封装成各个组件, 辅以回调机制将它们纳入框架的控制范围之内。所以, 框架提供了可在应用程序之间共享的复用的公共结构^[12]。现代软件开发越来越倚重框架的使用, 在系统开发中使用框架技术能大大缩短应用系统的开发周期, 提高系统生产效率和质量。中小企业创新服务平台系统在设计时也采用了 J2EE 的分层架构, 同时在各层中选择了相应的框架技术。

目前 J2EE 中表现层框架很多, 如 Struts, Tapestry, JSF 等, 其中 Struts 框架出现得比较早, 在国内使用者众多, 是一个比较优秀的表现层框架。使用 Struts 框架, 能够将业务数据、页面显示与动作处理相互分离, 能提高系统的可扩展性和可配置性, 能提高系统的开发效率, 另外 Struts 具有集中的导航机制, 能使系统结构清晰, 使系统的脉络更容易把握, 而且它提供了大量的标签库, 能大大简化 JSP 页面代码。考虑到 Struts 的这些优点, 在平台系统表现层设计时采用了 Struts 框架。

在平台业务层设计中, 系统选用了 Spring 这个优秀的开源框架, 选用这个框架是因为: ①能够使用 Spring 的 IOC (控制反转), 使得在开发时能通过简单修改配置文件来连接不同的应用组件, 使得各个组件之间的依赖性降低。②能够使用 Spring 的 AOP(面向切面编程), 使得各组件能更加关注于自己的业务, 而不需要关心系统中使用的如事务, 异常处理等其他服务。③ Spring 提供了常用 Web 框架的支持(包括本系统中使用的 Struts 框架), 而且其 O/R(对象/关系)映射模块能使得系统中能够方便的插入其他的 ORM 对象映射工具。

中小企业创新服务平台选用了 SQL Server 这种关系数据库作为数据源, 而在开发常规关系数据库的系统时, 通常通过 SQL 语句读取及操作数据库数据。在 Java 中, 最原始、最直接的方法是直接通过 JDBC 编程来访问数据库。这种方式的优点是运行效率高, 但缺点是在程序中要嵌入大量 SQL 语句, 造成数据冗余是不可避免的, 而且在代码中需要编写大量重复的代码, 如获得连接、准备语句、执行语句、遍历结果集以及其他一些 JDBC 特定元素, 尤其是在当系统中涉及到大量的数据库表, 而且表之间的关联关系比较复杂时, 通过 JDBC 来开发实施显得非常困难, 项目也难以维护。目前通用的持久层框架都建立在 OO 设计思想之上, 通过将数据库中的关系映射为

开发人员容易接受的对象,能使开发人员以更自然的方式来处理对象相关属性。另外数据库的访问往往成为整个系统的瓶颈,开发时需要花费大量的精力设计数据库来提高数据库的性能。现在大多数持久层框架都提供了优秀的性能优化机制,如内置的数据库连接池支持,数据缓存等。而且现在流行的持久层框架往往支持多种类型的数据库,这些优秀的框架在数据库访问上都进行了良好的设计,能够使开发人员在实际编码中不需要大量的涉及到某一种数据库,而只需要通过修改配置文件就能在各种数据库之间切换。因此本系统在设计中采用 Hibernate 这个优秀的开源框架来实现系统持久层。通过持久层中引入 Hibernate 框架,可以保证系统频繁地访问数据库的性能,降低访问数据库的频率,另外在开发中不需要过多关注如多线程、缓存、事务管理等细节,提高开发效率,而且还可以使用其 HQL 语言来使用面向对象的思维来操作数据库。

在本系统架构设计中,还需要一套在不同的层间移动的对象,因此系统中加入了领域模型层,这一层中包含的是表达实际业务对象的对象。该层允许能让开发者不再构建和维护不必要的数据传输对象 DTO 来匹配其领域对象。这种模型允许 Java 开发者能够以 OO 风格的方式很自然的处理对象,而不用编写额外的代码^[13]。

总之,在本系统中采用基于 Struts, Spring 与 Hibernate 的框架架构技术,能够降低系统各层之间的耦合性,使系统稳定性得到基本的保障,能给本系统的开发带来非常大的便利,提高了系统的扩展性和可维护性。在下一节中,将会对这三个框架进行深入的研究。

根据上面的分析,中小企业创新服务平台系统的总体框架结构图如图 3-1 所示:

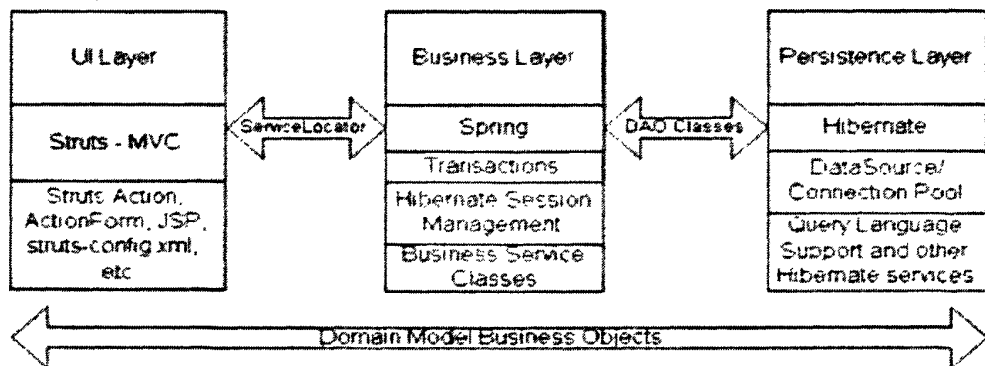


图 3-1 Struts, Spring 和 Hibernate 框架架构概览图^[14]

3.2 架构各层框架技术研究

根据上一节对系统架构的总体设计，表现层、业务逻辑层与持久层分别基于 Struts、Spring 和 Hibernate 这三个框架进行开发，下面对这三个框架进行较深入的研究。

3.2.1 Struts 框架研究

3.2.1.1 Struts 框架介绍

Struts框架是Apache软件基金会下Jakarta项目下的一个子项目，是在JSP Model2基础上实现了MVC设计模式的WEB框架，它继承了 MVC模式的各项特性，并根据 J2EE的特点，做了相应的变化与扩展。Struts框架很好的结合了 JSP, Java Servlet, Java Bean, Taglib等技术^[15]。图3-2显示了Struts框架的组件组成：

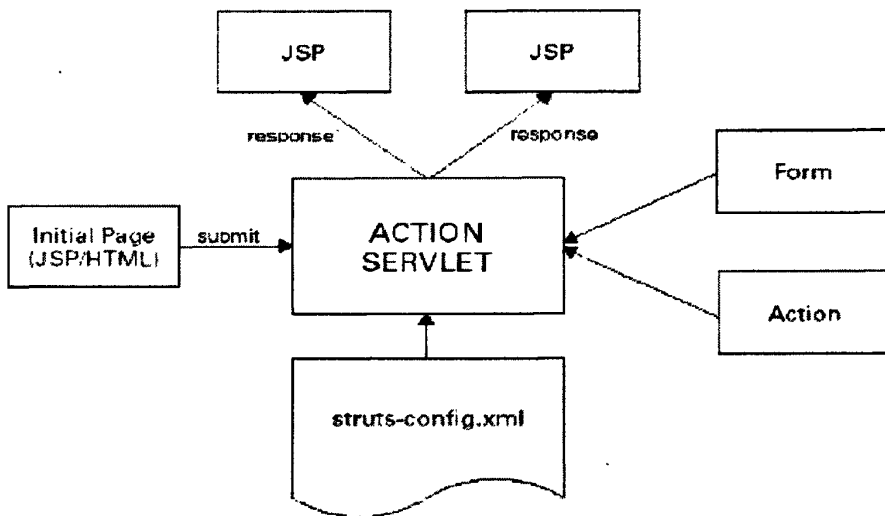


图3-2 Struts 组件图^[15]

在 Struts 中，承担 MVC 中控制器角色的是 ActionServlet。ActionServlet 是一个通用的控制组件，这个控制组件提供了处理所有发送到 Struts 的 HTTP 请求的入口点。它截取和分发这些请求到相应的动作类(这些动作类都

是 Action 类的子类)。另外控制组件也负责用相应的请求参数填充 ActionForm(FormBean), 并传给动作类(ActionBean), 然后动作类访问核心业务逻辑, 最后动作类将控制权通过 ActionMapping 对象来传递给后续的 JSP 文件或者其他动作类, 最终由 JSP 文件生成视图。所有这些控制逻辑利用 Struts-config.xml 文件来配置。在 Struts 框架中, 视图主要由 JSP 生成的页面来完成, Struts 提供丰富的 JSP 标签库, 这既有利于表现逻辑和程序逻辑的分离, 又能够减轻开发人员的负担^[16]。

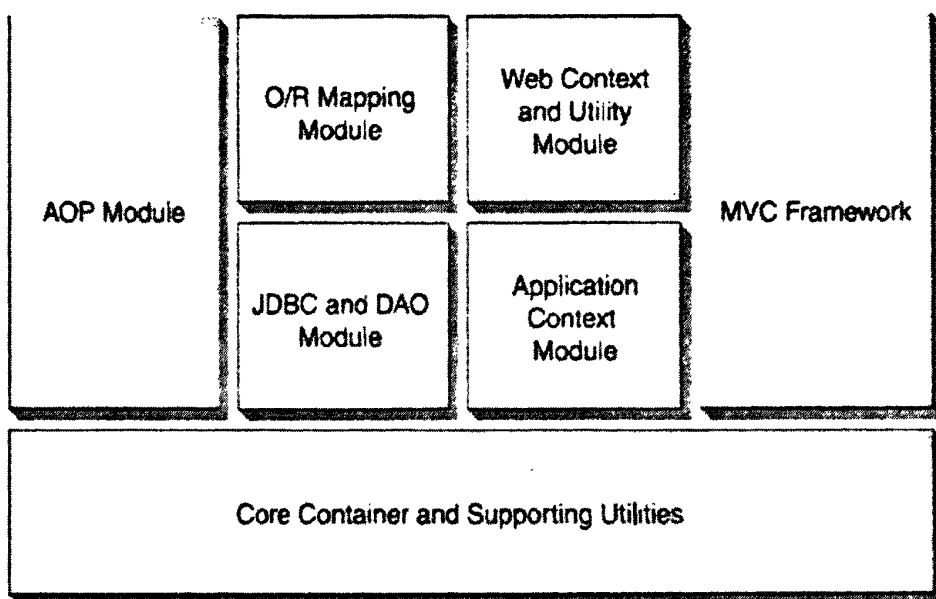
3.2.1.2 本文采用 Struts 框架的必要性

- Struts 实现了 MVC 设计模式, 继承了 MVC 的大部分优点, 能够使系统功能模块划分清晰, 降低各功能之间的耦合度, 可以提高软件的可扩展性等。
- Struts 拥有集中的控制机制与灵活的系统导航机制, 便于开发人员对系统的整体把握。
- Struts 提供了功能强大的标签库, 大大简化了 JSP 页面代码。
- Struts 是开源软件, 能够免费学习及使用, 而且其性能越来越完善。
- Struts 属于轻量级框架, 能够减少对系统资源的占用。

3.2.2 Spring 框架研究

3.2.2.1 Spring 框架介绍

Spring 也是一个开源框架, 由 Rod Johnson 创建, 是一个轻量级的 IoC 和 AOP 容器框架。Spring 既是综合性的也是模块化的。基于其分层架构, Spring 能够使开发者灵活地单独使用其任何一部分。Spring 由七个模块组成(如图 3-3 所示)。这些模块都是松耦合的, 在实际开发过程中你可以选择使用一些, 大多数, 或所有的 Spring 框架支持的组件来构建应用程序^[17]。

图 3-3 Spring 框架模块组成图^[18]

Spring 的所有模块都建立在 Core Container 即核心容器之上, Spring 容器为 Spring 框架提供了基础功能。在这个模块中有在 Spring 中非常重要的 BeanFactory 类, 这个类采用工厂模式来实现 IoC。Application Context 模块扩展了 BeanFactory, 添加了国际化、系统生命周期以及验证的支持, 这个模块还提供了如 JNDI、EJB、远程调用等企业级服务。Spring AOP 模块直接将面向方面的编程功能集成到了 Spring 框架中, 所以, 可以很容易地使 Spring 框架管理的任何对象支持 AOP, 这个模块为基于 Spring 的应用程序中的对象提供了事务管理服务。通过使用 Spring AOP, 不用依赖 EJB 组件, 就可以将声明性事务管理集成到应用程序中。JDBC 和 DAO 模块主要目标是为了简化 JDBC 的编程, 方便构建健壮的应用程序。另外 Spring 框架插入了若干个 ORM 框架, 从而提供了 ORM 的对象关系工具, 其中包括 JDO、Hibernate 和 iBatis SQL Map。Web 上下文模块建立在 ApplicationContext 模块之上, 这个模块为基于 Web 的应用程序提供了上下文。因此, Spring 框架支持与 Jakarta Struts 等的集成。而且 Spring 为 Web 系统提供了全功能的 MVC 框架。在实际开发中, 可以完全不采用 Spring 本身提供的 MVC 框架, 而使用其他的技术^[17]。

下面是 Spring 的重要特点:

- ① 轻量级: Spring 整个框架非常小, 而且它是非侵入式的, 基于 Spring

开发的系统一般不用依赖于 Spring 的类。

② IoC: Spring 提供反向控制来实现松耦合。使用 IoC, 对象在需要其依赖类时是被动的接受而不需要自己主动去找, 这就意味着对象不是从容器中查找它的依赖类, 而是容器在实例化对象时主动地将它的依赖类注入给它。

③ AOP: Spring 对面向切面编程提供了强大的支持。一个应用系统通常由很多个组件组成, 每个组件负责一部分功能, 然而这些组件除了自身所拥有的核心功能外还带有其他的一些附加功能, 如日志, 事务管理和安全等, 这些系统服务通常叫做交叉业务, 因为这些服务散落在系统的很多组件中, 给系统增加了双重复杂性: 实现系统级的业务代码在多个组件中复制, 不利于修改; 各个功能组件因为夹入了与自己核心业务无关的代码而变得复杂, 尤其是加入多个系统级服务时变得相当杂乱。AOP 能够将这些服务模块化, 并通过声明式地应用于需要它们的地方。这样就使得组件更加专注于自己业务, 而不需要关心涉及到的其他的系统服务。

④ 容器: Spring 是一个容器, 它包含并管理系统对象的生命周期和配置。

⑤ 框架: Spring 实现了使用简单的组件地组合成一个复杂的系统的方法。在 Spring 中, 各个组件能够通过 XML 配置文件组合起来, 并且 Spring 本身也提供了如事务管理, 持久层集成等的支持。

3.2.2.2 本文采用 Spring 框架的必要性

- Spring 提供的 IoC 机制, 使得开发时能够通过简单修改配置文件来连接不同的应用组件, 降低组件间的耦合度。

- Spring 提供的 AOP 模块能够将散落在系统中的附加代码抽取出来, 通过声明来将这些服务加入到需要它们的地方, 系统中使用 Spring 的 AOP 来实现事务管理等。

- Spring 框架也是一个轻量级的框架。

- 非侵入性, 应用程序对 Spring API 的依赖可以减至最小。

- Spring 框架提供了对其他 J2EE 框架的整合支持, 如 Hibernate 和 Struts, 有利于本系统的集成。

3.2.3 Hibernate 框架研究

3.2.3.1 Hibernate 框架介绍

Hibernate 是目前最流行的 ORM 框架之一，它采用非常优雅的方式，将 SQL 操作完全包装成对象化的操作。它提供了强大，高性能的对象到关系型数据库的持久化服务。Hibernate 内部封闭了 JDBC，通过向上层提供面向对象的数据访问 API，能够使系统方便的实现 Java 对象的持久化。

Hibernate 的体系结构如图 3-4 所示：

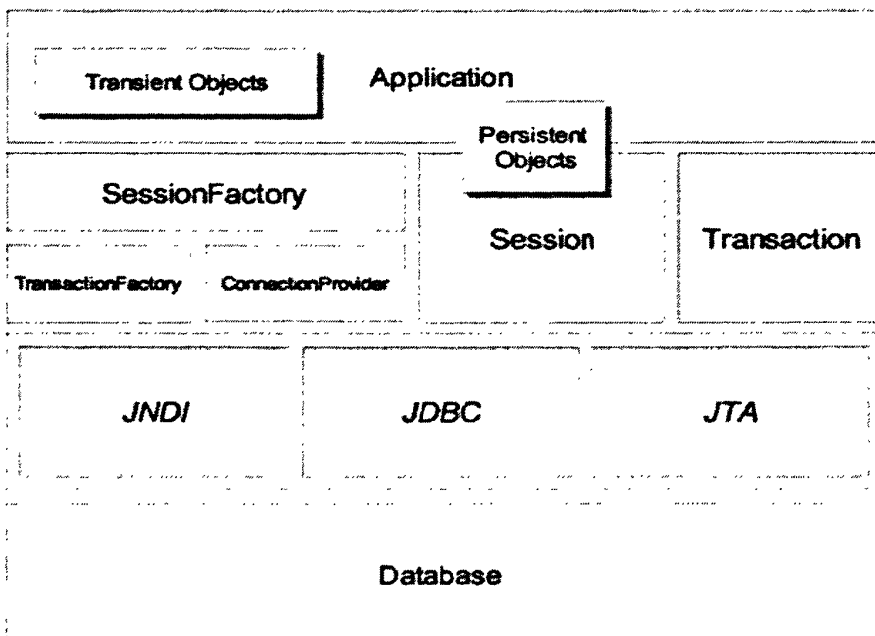


图 3-4 Hibernate 体系结构图^[19]

从图中可以看出 Hibernate 将应用层从底层的 JDBC/JTA API 中抽象出来，而由其本身来进行处理。图中 SessionFactory 是针对单个数据库映射关系经过编译后的内存镜像，它保存了当前数据库配置的所有映射关系，同时它也负责维护当前的二级缓存和 Statement Pool。它是生成 Session 的工厂。Session 表示应用程序与持久层之间交互操作的一个单线程对象，在其中隐藏了 JDBC 连接，它是 Transaction(事务)的工厂，它提供了众多的持久化方法如 save, update, delete, find 等方法。事务 Transaction 是应用程序用来

指定原子操作单元范围的对象，它是单线程的，生命周期很短。`ConnectionProvider` 是生成 JDBC 连接的工厂，它通过抽象将应用从底层的 `Datasource` 或 `DriverManager` 隔离开。`TransactionFactory` 是生成 `Transaction` 对象实例的工厂。`Persistent` 对象是带有持久化状态的、具有业务功能的单线程对象，这些对象正与 `Session` 对象相关联。`Transient` 对象是目前没有与 `session` 关联的持久化类实例^[19]。

3.2.3.2 本文采用 Hibernate 框架的必要性

- Hibernate 提供了面向对象的数据访问机制，能够使得开发者用 OO 的思维来操作数据库，能够大大提高开发效率。
- Hibernate 提供对多类型数据库的支持，在不同数据库之间切换时只需要通过配置文件就可以完成，而且代码中不需要大量的涉及到某一种数据库的操作，便于系统移植。
- Hibernate 提供了优秀的性能优化机制，能够提升系统运行效率。

3.3 系统总体功能设计

通过对某自治区中小企业创新平台的需求分析，可以把整个系统分成下面几个子系统模块：公共技术服务子系统，投资融资服务子系统，行业科技子系统，人才引进及培训子系统，创新成果转化子系统，商务信息交流子系统。这些子系统下面又包括若干个功能模块。另外为了保证系统的正常运行，在整个系统中还要加入系统管理这个模块，这个模块包括用户管理，用户组管理，权限管理，日志分析及系统功能菜单管理等。系统整体功能结构图如图 3-5 所示：

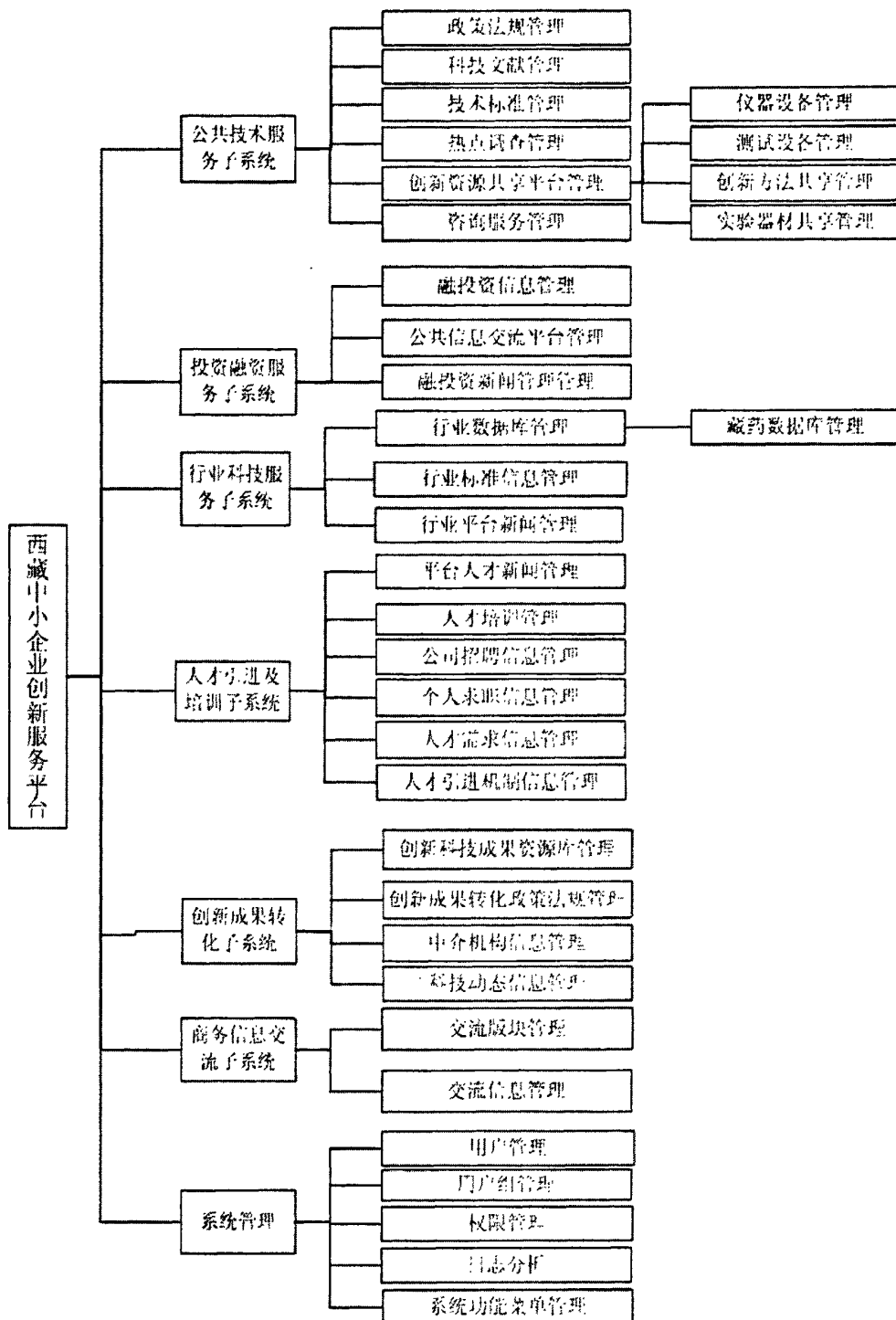


图 3-5 系统整体功能结构图

3.4 系统主要功能详细设计

根据上一节系统整体功能结构图可以看出中小企业创新平台系统具有比较多的子功能系统,由于篇幅所限,在这里并不对所有子系统进行详细设计,而仅对平台中比较重要的系统管理、公共技术服务管理、人才引进及培训服务管理以及商务信息交流管理做进一步的设计说明。

3.4.1 系统管理设计

从系统整体功能结构图可以看出,系统管理主要包括以下五个部分:用户管理、用户组管理、权限分配管理、日志分析以及功能菜单管理,下面对各个功能进行详细设计:

- 用户管理:系统管理人员登陆后台时,系统先判断用户名和密码是否正确,如果正确,则生成用户列表显示页面。系统提供对用户的查询和更新操作。查询用户可以分为按用户 ID、登陆时间范围、登陆次数范围、用户组查询以及用户状态(禁用与未禁用状态)查询。更新操作包括:对用户是否激活和锁定,用户组分配。若用户状态为锁定状态,则用户不能登陆。一个用户可以同时属于若干个用户组,可以对用户的用户组进行编辑,增加和删除用户组。用户注册时有一个默认的用户组,默认用户组对应某一组权限。另外考虑到用户密码的安全性,系统采用 DES 算法对用户密码进行加密。DES 算法的入口参数有三个:Key、Data、Mode。其中 Key 是 DES 算法的工作密钥(DES 算法中只用到 64 位密钥中的其中 56 位);Data 是要被加密或被解密的数据;Mode 为 DES 的工作方式:加密与解密。其中 DES 算法是这样工作的:如 Mode 为加密,则用生成的密钥将数据 Data 进行加密,生成 Data 的密码形式作为 DES 的输出结果;如 Mode 为解密,则采用密钥将密文解密,还原明文作为 DES 的输出结果。

Java 中密码扩展包含了 Cipher 类,这个类是所有加密算法的超类。在本系统中,生成密码对象时,设置其算法名称为 DES,再设置其模式为 ENCRYPT_MODE,并将系统生成的密钥对象(其中密钥采用 KeyGenerator 根据初始化参数来生成)传给 Cipher 对其初始化,然后就可以通过 Cipher 对象将用户密码这种明文字符串转化为密文。系统数据库存储的是用户密码经过转换后生成的密文。

- 用户组管理:系统管理人员登陆后台后,如果具有相应权限,能够

增加、查看、修改和删除用户组。系统默认的用户组有系统管理员、行业专家、个人用户、企业用户、信息服务部用户及市场部用户,在此基础上,系统管理员能够增加用户组,增加用户组时需输入用户组名称、描述、选择权限列表中对应用权限以分配该用户组默认权限。查看用户组能够查看用户组的详细信息。另外,系统管理员能够修改用户组,编辑名称,描述等。删除用户组时,先检查是否有用户属于此用户组,若用户组为空,则能够删除用户组,若不为空,则提示不能删除。

- 权限管理:系统使用基于用户组的权限管理。用户组的权限是以字符串的形式存储的,字符串长度为 2000 位,系统中的各资源的访问权限对应字符串中的某一位,字符串每位的值为 1 和 0,其为 1 代表本组具有对应位的权限,0 代表没有权限。系统管理员或具有权限分配的相关工作人员可以给用户分配多个组的权限,用户的最终权限是用户属于的各个用户组对应权限进行“或”的结果,即:只要用户对应的各个用户组只要有一个具有权限,则此用户具有该权限。后台配置用户权限分配某资源访问权限时,只需要给用户的对应组分配访问资源的权限或者给用户添加具有访问资源权限的用户组。因为可能有不同的用户角色登陆到后台,而不同的用户角色具有不同的权限,因此登陆后生成的功能树形菜单根据其权限是不一样的,只有具有操作权限的对应功能才能在树形菜单中出现。

- 日志分析:在一个完整的信息系统里面,日志系统是一个非常重要的功能组成部分。系统管理员能够通过日志系统所记录的信息为系统的排错,优化系统性能。本系统的日志框架基于 Log4j, Log4j 是 Apache 的一个开放源代码项目,通过使用 Log4j,可以控制日志信息输送的目的地,如控制台、文件、GUI 组件、甚至是套接口服务器等。也可以控制每一条日志的输出格式,另外通过定义每一条日志信息的级别,能够更加细致地控制日志的生成过程。而且这些配置可以通过一个配置文件来完成,而不需要修改应用的代码。通过 Log4j,配置日志的输出格式,日志信息级别以及日志信息输送目的地。系统管理员能够通过生成的日志文件来对系统进行监测,保证系统的正常运行。在本系统设计中定义了两个日志输出目的地:一个为控制台、一个为文件,定义日志记录的优先级为 ERROR,系统中通过日志分析主要是为了监测系统运行状况,系统崩溃时能找到原因等。

- 系统功能菜单管理:在相关人员登陆后台进行操作时,不同的用户角色具有不同的权限,因此管理界面中功能菜单是不一样的。功能菜单以目

录树的形式提供给用户，目录树有两级菜单，以 JavaScript 的形式生成，当用户点击相应菜单项时，显示相应功能页面。

根据 J2EE 的分层结构体系，结合实际业务分析，这里以系统管理中的功能菜单目录树生成为例对系统管理作深入分析。类 MenuDAO 是业务层与数据库进行交互的入口，MenuDAO 接口继承自 GenericMenuDAO 接口，GenericMenuDAO 接口中通过泛型定义了业务层访问数据库进行增删改查等基本操作方法，MenuDAO 接口则定义了实际业务中需要实现的方法，如这里的生成目录方法 geneMenu()。MenuDAOImpl 类实现了 MenuDAO，MenuDAOImpl 类继承自 GenericMenuDAOImpl 类，GenericMenuDAOImpl 类继承自 Spring 中与 Hibernate 集成较重要的类：HibernateDaoSupport。HibernateDaoSupport 类是 Spring 为了封装 Hibernate 中基本增删改查操作而提供的工具类，这个类也封装了如异常处理等操作，为在 Spring 中集成 Hibernate 提供了极大的方便。MenuDAOImpl 实现了 GenericMenuDAO 通用接口。同时，MenuDAOImpl 也实现了 MenuDAO 中定义的几个实际业务中需要用到方法。类 TibetCServiceIndexAction 类是最终与 JSP 页面交互的类，其中的 generateLeft()、generateMain()、generateTop() 与 generateFoot() 方法分别生成功能菜单、主界面、页面头部与底部，parseMenu() 方法用户处理菜单生成，其基本功能是通过 PriviledgeHelper 类来完成的，这个类中定义了获取用户角色权限字符串的一些方法。

另外，在 TibetCServiceIndexAction 中获得 MenuService 实例，MenuServiceImpl 中获得 MenuDAOImpl 实例都是通过 Spring 框架中的依赖注入方式获得的。

系统管理功能目录树生成类图如图 3-6 所示：

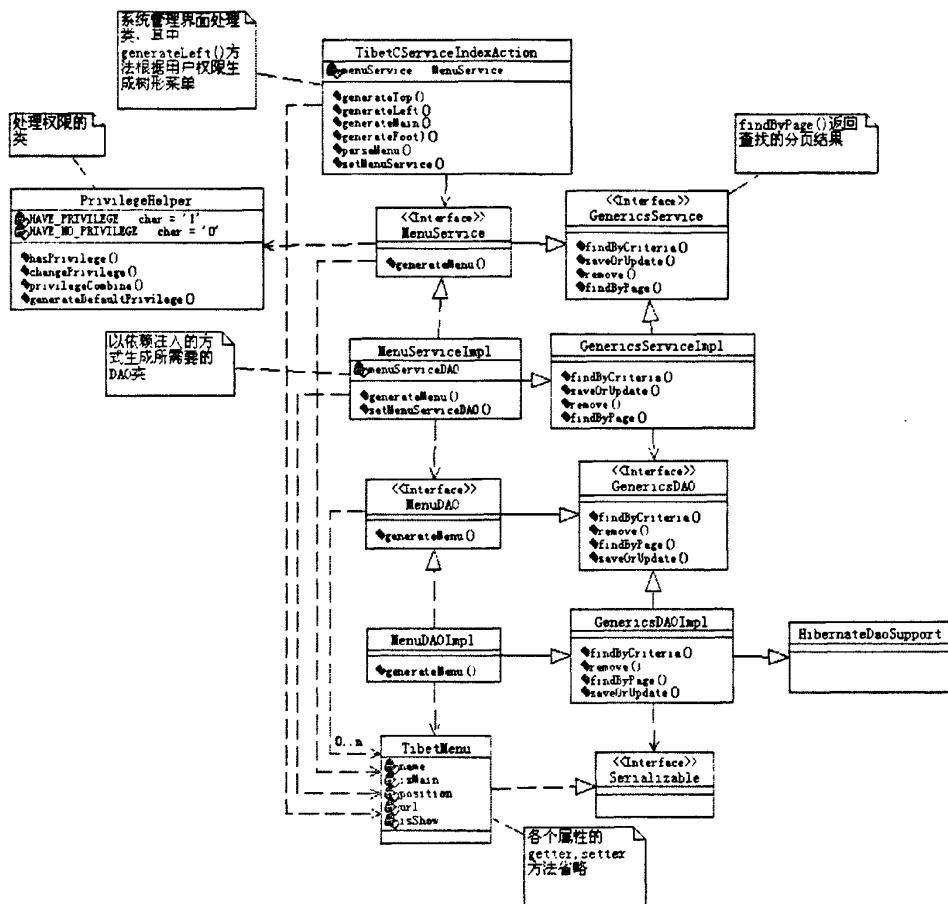


图 3-6 系统管理功能目录树生成类图

3.4.2 公共技术服务管理设计

从系统整体功能结构图可以看出，公共技术服务主要包括以下几个部分：政策法规，科技文献及技术标准服务，热点调查管理，创新资源共享管理和咨询服务管理等几个部分。下面分别对这几个部分设计进行详细的介绍：

- **政策法规，科技文献服务及技术标准管理设计：**这一部分主要涉及信息部工作人员的后台管理。信息部工作人员在获得相关权限后，后台生成相应的功能菜单目录树，信息部工作人员能够对政策法规、科技文献以

及技术标准进行处理,处理主要包括编辑、删除、添加新类型和添加、更新和删除新文章。以政策法规为例,因为一种类型下面有多种子类型,相当于建一个树形结构类型,信息部工作人员可以在任意一个结点下面添加新的类型和对现有类型进行编辑,而且信息部工作人员能够在叶子结点下面增加各类型的文章,这里的文章类型包括政策法规、科技文献和技术标准等。添加文章时,先选择分类,类型以下拉菜单显示,类型列表根据以上建立的叶子结点类型生成,文章名称是必不可少的,否则提交时会提示错误,另外,信息部工作人员也可以选择文章名称的显示颜色以突出重点文章。新的文章可能需要显示文章来源,因此提供文本框提示操作员输入对就的来源及链接地址。在编辑文章内容时,因为文章内容最终以网页的形式供用户检索,在设计中使用 FCKeditor 这个文字编辑器,FCKeditor 是一个专门使用在网页中开源的所见即所得的工具。通过 FCKeditor 编辑网页时,可以灵活的插入图片、flash、表情符号、表格和各种表单,能够使用编辑器中的各类型模板,可以对文本,图片等进行格式化处理,而且可以随时预览网页中显示的效果。相关文章中可能需要添加附件,FCKeditor 也提供了上传附件功能,为了突出显示,系统中重新设计文件上传功能,文件上传基于 Struts 来实现。信息部工作人员在后台中编辑文章时,能够添加新的附件、编辑和删除已有的附件。上传附件时,需要输入附件名称,选择上传文件路径,否则会提示错误。

- 热点调查管理设计:具有权限的信息部工作人员在后台能够新增热点调查、编辑和删除已有热点调查。新增热点调查时需要输入调查名称,调查类型(单选还是复选),调查选项内容。编辑时能够修改调查名称,选项名称。考虑到调查结果显示要比较直观,因此热点调查采用 JFreeChart 这种 java 图形解决方案,它是 JFreeChart 公司在开源网站 SourceForge.net 上的一个项目。通过 JFreeChart,能较容易的生成饼图(二维和三维),柱状图(水平,垂直),线图,点图,时间变化图,甘特图,股票行情图,混和图,温度计图,刻度图等常用图表。

- 创新资源共享管理设计:企业和个人在前台能够申请共享自己的创新资源,也能够查看、检索平台中其他用户共享的资源。用户在申请共享自己的创新资源时,需要选择资源类型,输入对应资源的其他详细信息以及自己联系方式等,提交后需要信息部工作人员进行审核。信息部工作人员在后台管理时,以分页形式显示用户资源共享申请,信息部工作人员能

够查看申请详细信息，能够单选和多选多个申请进行删除和审核通过等操作。

- 咨询服务设计：信息部工作人员能够在后台编辑和发布咨询服务申请流程。企业和个人用户需提交咨询时，需要填写用户详细信息，选择咨询问题行业和类型，否则提交时会报错。相关行业专家登陆后，页面根据未回复和已回复类别显示咨询问题列表，而且能够根据行业类别、咨询时间等进行检索，对咨询问题进行详细解答。信息部工作人员在后台登陆后，显示用户咨询申请列表和行业专家回复列表，信息部工作人员能够批量审核和删除用户咨询问题列表，能够查看申请详细信息及专家回复内容。只有在用户咨询问题列表经过信息部工作人员审核通过后行业专家才能查看到咨询问题列表。

因为在公共技术服务管理涉及到的服务比较多，这里以政策法规，科技文献服务及技术标准管理为例来对设计作进一步的详细分析：同样设计中以面向接口编程为原则。与 JSP 页面进行交互的类为 `CatalogAction` 与 `ArticleAction`，其中 `CatalogAction` 包含文章类型的操作，如对类型的增加、删除、修改和查找。`ArticleAction` 也包含了对特定类型的文章的操作，包含对文章增、删、改及查的操作。另外因为文章中需要有附件，因此也包含对附件的操作，而对特定文件的上传下载操作是通过 `FileUtil` 类来实现的。增加文章时，需要选择对应的文章类型，因此 `ArticleAction` 中需要调用 `CatalogService` 中的方法。`xxxAction` 中调用的都是 `xxxService` 接口，其实例也是 Spring 通过依赖注入来实例化的，同样 `xxxService` 也是通过依赖注入得到 `xxxAction` 实例的。政策法规，科技文献服务及技术标准等文章管理类图如图 3-7 所示：

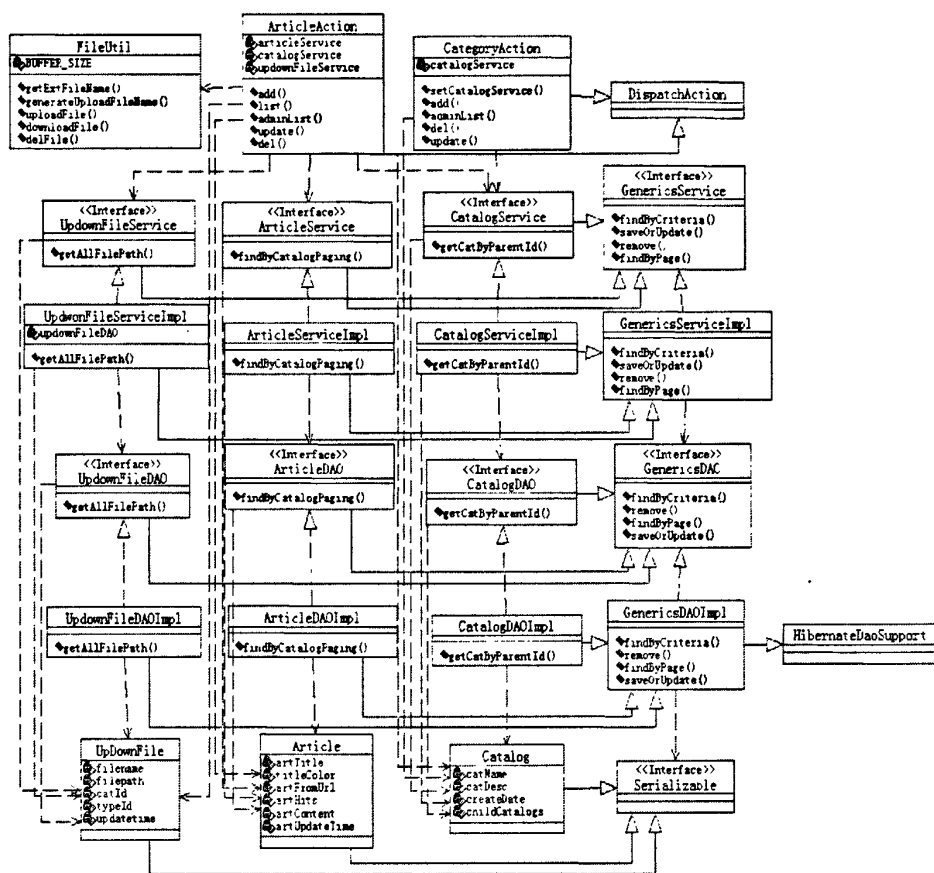


图 3-7 公共技术服务文章管理类图

3.4.3 人才引进及培训服务设计

人才引进及培训服务包括平台人才新闻管理, 人才培训管理, 公司招聘信息管理, 个人求职信息管理, 人才需求信息管理, 人才引进机制信息管理等几个部分, 下面分别对这几部分进行详细设计说明:

- 平台人才相关新闻管理, 人才需求信息管理, 平台人才引进机制信息管理: 市场部工作人员在后台登陆后, 能够发布、更新、查询和删除平台人才相关新闻、发布人才需求信息和区内人才引进机制。以平台人才相关新闻为例, 新建新的人才新闻时, 需要输入题目跟详细内容, 题目输入是必须的, 否则提交时会报错, 新闻内容编辑和更新也是通过 FCKeditor 来完成的。所有新闻以分布形式按发布时间降序来排列, 查询新闻提供按标题内容、

发布时间范围等的检索方式。删除时可以选择单个新闻进行删除或者选中相应新闻的复选框进行指删除。

● 人才培训管理设计：市场部工作人员登陆后台后，能够发布新的技术培训通知，发布时需要输入培训通知题目、培训时间、培训地点、如果有上限的话需要输入人数上限，另外需要选择培训类别、输入主讲人和详细描述信息。企业和个人用户登陆后能够查看人才培训通知，可以点击选中的培训通知申请按钮。市场部工作人员能够查看用户培训申请的详细情况。如果培训人数有上限，而且现有人数未超过上限，可以点击用户申请信息行的“申请通过”按钮同意，否则点击“不通过”拒绝申请，市场部工作人员也可以单选和多选培训申请进行操作。申请通过后，自动生成带有分配标识号的邀请函给用户。用户可以查看自己的邀请函，点击确认后提交确认信息。市场部工作人员能够检索用户确认信息，并可以将确认信息删除或者分类保存到数据库。

● 公司招聘信息管理设计：企业用户登陆后能够发布招聘信息，发布招聘信息时，需要输入招聘内容的名称、职位名称，选择职位类别，职位类别有全职、兼职和临时三种，选择学历要求，输入需求人数、工作地点、职位工资、职位的详细、职位说明、招聘信息的有效截止日期及应聘者需要提交的材料等。输入完成提交后需要通过具有相关权限的市场部工作人员审核后才能显示。市场部工作人员在后台能够按时间、类别等多种方式检索待审核的企业招聘信息。所有招聘信息也是按发布时间降序来排列的，另外，待审核的企业招聘信息通过审核后默认是可以显示的，市场部工作人员也可以选择使其为不显示状态，还能够批量审核通过、解禁和删除企业的招聘信息。企业招聘信息通过审核后能够在前台显示，只有登陆用户并具有相应权限的用户才能查看企业招聘信息。未登陆的用户点击查看公司招聘信息时，会提示未登陆并跳转到用户登陆页面，不具有权限的用户点击查看公司招聘信息时，会提示用户不具有权限。企业用户登陆后能够查看、检索和修改自己发布的招聘信息，修改招聘信息后，其状态会自动变为未审核状态，只有通过市场部工作人员审核后才能在前台显示。

● 个人求职信息管理设计：个人用户登陆后能够发布个人求职信息，发布个人求职信息时，需要输入求职信息的题目、求职意向，选择工资要求范围、输入期望工作地点、职位类别和个人的自我描述，个人简历是个人求职信息中的重要的一部分，个人用户可以在求职信息中上传个人简历。另外，

还提供一个简历模板供用户填写,个人用户能够通过简历模板详细的描述个人信息。具有权限的市场部工作人员登陆后也可以按发布时间,类别来检索待审核的个人求职信息。同样,待审核的个人求职信息以分页列表显示,操作人员能够审核通过、解禁和删除个人用户的求职信息。只有通过审核并且为可显示状态的个人用户求职信息才能够在前台显示。具有权限的用户可以查看和检索个人求职信息,查看个人求职信息时,如果用户填写了个人简历,还可以查看该用户的简历。如果用户上传了个人简历,还可以下载用户的简历。同样,如果未登陆的用户点击查看个人求职信息时,会提示未登陆并跳转到用户登陆页面。不具有权限的用户点击查看个人求职信息按钮时,会提示用户不具有权限。另外,个人用户登陆后,可以检索自己发布的个人求职信息,并可以修改自己的求职信息。同样,修改求职信息后,求职信息状态会自动转为待审核状态,只有市场部工作人员通过审核后才能够在前台显示。

3.4.4 商务信息交流服务设计

商务信息交流服务包括交流版块管理和交流信息管理两个部分,下面分别对这两个部分进行详细设计说明:

- 交流版块管理: 具有版块管理操作权限的信息部工作人员在后台登陆后,显示所有版块列表页面,信息部工作人员能够添加新版块,添加版块时需要输入版块名称和版块的描述信息。信息部工作人员还能够更新和删除已有版块,在删除版块时会提示用户是否确认删除版块及版块中的帖子列表,用户确认后才能够删除,删除内容包括版块、对应版块中的所有帖子以及帖子中的所有回复。另外,信息部工作人员还能够禁用和解禁版块,只有解禁的版块用户才能够进行查看和发帖等操作。前台具有相关权限的用户登陆后,能够查看所有版块列表,版块信息包括版块名称、版块描述、主题数及最后发表时间。点击进入版块后,能够查看分页显示的帖子列表,帖子包括点击次数、主题、回复次数、作者、发表时间以及更新时间,用户还能够查看发帖用户详细信息。点击帖子后,能够显示帖子的详细信息,包括帖子内容,回复的分布显示列表,其中回复包括内容、发布时间、回复用户,用户也可以查看回复用户的详细信息。

- 交流信息管理: 用户登陆后,能够查看商务交流平台中的各版块内容。具有权限的用户还能够发布新帖,发布新帖时需要输入帖子题目和内容。

用户还能够对某个帖子进行回复，回复也通过 FCKeditor 来编辑回复内容。用户在发布新帖和回复帖后，状态为“不显示”，只有具有相关权限的信息部工作人员在后台对其进行解禁使其状态为“显示”后，帖子和回复才能够显示。信息部工作人员登陆后，能够查看所有版块下的帖子内容，查询所有回复，可以批量删除、禁用和解禁帖子和回复内容。

3.5 数据库设计

数据库设计的优劣往往关系到整个系统的运行性能。数据库设计中要尽可能的保证查询效率，节省存储空间，减少数据冗余，另外设计过程要考虑到数据库的易维护性及可扩展性等原则。

3.5.1 数据库概念结构设计

由于中小企业创新服务平台包含多个服务子系统，系统中业务比较复杂，限于篇幅，这里以平台中较核心，业务较独立的人才引进及培训子系统来分析数据库的设计过程。

概念结构是对现实世界的一种抽象,它是各种数据模型的共同基础，它比数据模型更独立于机器，更抽象，从而更加稳定。将需求分析得到的用户需求抽象为信息结构即概念模型的过程就是概念结构设计。概念结构设计是整个数据库设计的关键^[20]。描述数据概念结构的一种有力的工具是 E-R 模型。

根据需求分析可知，参与到人才进行引进及培训子系统中有个人用户、企业用户和系统操作员，可以将这些角色合并为“用户”这个实体，其中用户属于不同的用户组。用户还可以发布新闻，而新闻属于不同的新闻类别。用户还可以发布人才需求信息和相关人才引进机制文章，其中人才引进机制中还可以添加附件。平台为了促进用户创新能力发展，还可以发布培训通知，用户查看培训通知后可以发出培训申请，还可以收到培训邀请函，收到邀请函后还能够发出确认信息。另外，用户在平台上需要招聘人才，还可以发布招聘信息。用户在平台上还能够发布个人求职信息，个人求职信息可以包含个人简历文档，如果用户没有简历文档的话，还可以使用系统中提供的简历模板填写个人简历。人才引进及培训中涉及到人才引进机制相关附件、个人简历文档，可以将这两者合并成“文件”这个实体。另外，培训管理中涉及

到培训申请, 培训邀请函及确认信息合并为“培训明细”这个实体。

根据人才引进及培训服务管理的业务分析, 可以识别出的主要实体有: 用户、新闻、新闻类别、人才需求信息、人才引进机制、培训通知、培训明细、人才招聘信息、个人求职信息、简历和文件等。

人才引进及培训业务中主要存在如下的实体关系:

用户和用户组的关系用“属于”来描述, 一个用户组可以包含多个用户, 一个用户也可以属于多个用户组, 即 $m:n$ 的关系。

用户和新闻之间的关系用“发布”来描述, 一个用户能够发布多条新闻, 但一条新闻只能由一个用户发表, 它们之间构成 $1:n$ 的关系。

新闻和新闻类别之间的用“属于”来描述, 一条新闻只能属于一种新闻类别, 而一种新闻类别可以包含多条新闻, 它们之间构成 $1:n$ 的关系。另外, 一种新闻类别可能属于另一种新闻类别, 因此新闻类别自身构成 $1:n$ 的关系。

用户和人才需求信息之间也可以用“发布”来描述, 一个用户能够发布多条人才需求信息, 而一条人才需求信息只能由于一个用户发表, 它们之间构成 $1:n$ 的关系。

用户和人才引进机制之间的关系可以用“发布”来描述, 一个用户可以发布多条人才引进机制, 一条人才引进机制只能由一个用户发表。因此它们之间构成 $1:n$ 的关系。

人才引进机制与文件之间的关系可以用“包含”来描述, 一条人才引进机制可以包含多个文件, 而一个文件可以被多条人才引进机制所引用, 因此它们之间构成 $m:n$ 的关系。

用户和培训通知之间的关系可以用“发布”来描述, 一个用户可以发布多条培训通知, 一条培训通知只能由一个用户发布。因此这两者之间构成 $1:n$ 的关系。类似的, 一个用户可以关联多条培训明细, 而一条培训明细只能对应一个用户, 因此用户与培训明细之间构成一对多的关系。至于培训通知和培训明细这两个实体, 一条培训通知能对应多条培训明细, 而一条培训明细只能对应一条培训通知, 因此, 这两者之间构成一对多的关系。

用户和招聘信息之间的关系可以用“发布”来描述, 一个用户可以发布多条招聘信息, 而一条招聘信息只能由一个用户发表。它们之间构成 $1:n$ 的关系。

用户和求职信息之间的关系可以用“发布”来描述, 它们之间很显然构成 $1:n$ 的关系。而一条求职信息可以包含一份简历文档文件或者一份简历信

息,一份简历文档文件和一份简历信息只能对应一条求职信息,因此,求职信息与文件构成 1:1 的关系,同样,求职信息与简历信息也构成 1:1 的关系。用户能够上传多个文件,一个文件只能属于一个用户,因此用户和文件之间构成 1:N 的关系。而一个用户只能维护一份简历信息,一份简历信息只能对应一个用户,因此它们之间构成 1:1 的关系。

对每个实体定义的属性如下:

用户: {用户号, 登录名, 真实名称, 密码, 最近登录时间, 最近登录 IP, 登录次数, 地址, 电话, 传真, 电子信箱, 邮政编码, 用户描述, 个人网址, 是否锁定}

用户组: {用户组号, 用户组名称, 用户组描述, 功能字符串, 创建时间, 创建者序号, 能否被删除}

简历: {简历号, 名称, 性别, 政治面貌, 所在城市, 所在省份, 籍贯, 学历, 毕业学校, 专业, 就业意向, 意向城市, 工资要求, 详细信息, 用户序号}

求职信息: {求职信息号, 用户号, 求职标题, 求职职位, 所选行业号, 工作类别, 期望工作地点, 薪水要求, 求职意向, 自我评价, 发布时期, 截止日期, 简历号, 文件号}

行业: {行业号, 行业名称, 行业介绍}

招聘信息: {招聘信息号, 用户号, 招聘名称, 职位名称, 职位类别, 职位要求, 工作地点, 招聘人数, 学历要求, 职位描述, 截止日期, 提交日期, 详细细节}

人才需求信息: {人才需求信息号, 发布者用户号, 招聘名称, 职位名称, 职位类别, 工作地点, 招聘人数, 发布日期, 截止日期, 详细描述}

文件: {文件序号, 文件名称, 文档类型, 序列号, 存放地址, 描述, 创建者号, 更新时间, 更新者号}

人才引进机制: {文章号, 文章名称, 文章内容, 发布者号, 点击次数}

新闻类型: {类型号, 创建者号, 创建时间, 父类型序号, 类型名称, 类型描述}

新闻: {文章号, 创建者用户号, 发布时间, 题目, 题目颜色, 文章来源, 链接地址, 点击次数, 文章内容, 所属类别号, 更新用户号, 更新时间}

培训通知: {通知号, 名称, 培训时间, 详情, 发布者用户号}

培训明细: {培训明细号, 培训通知号, 申请人用户号, 申请信息, 审

核人用户号, 确认信息, 序列号, 状态}

最后得到人才引进及培训服务数据库的 E-R 图如图 3-8 所示。

(注意: 为了节省篇幅, 实体与属性之间的关系没有用图形表示, 实体的标识码用下横线划出)。

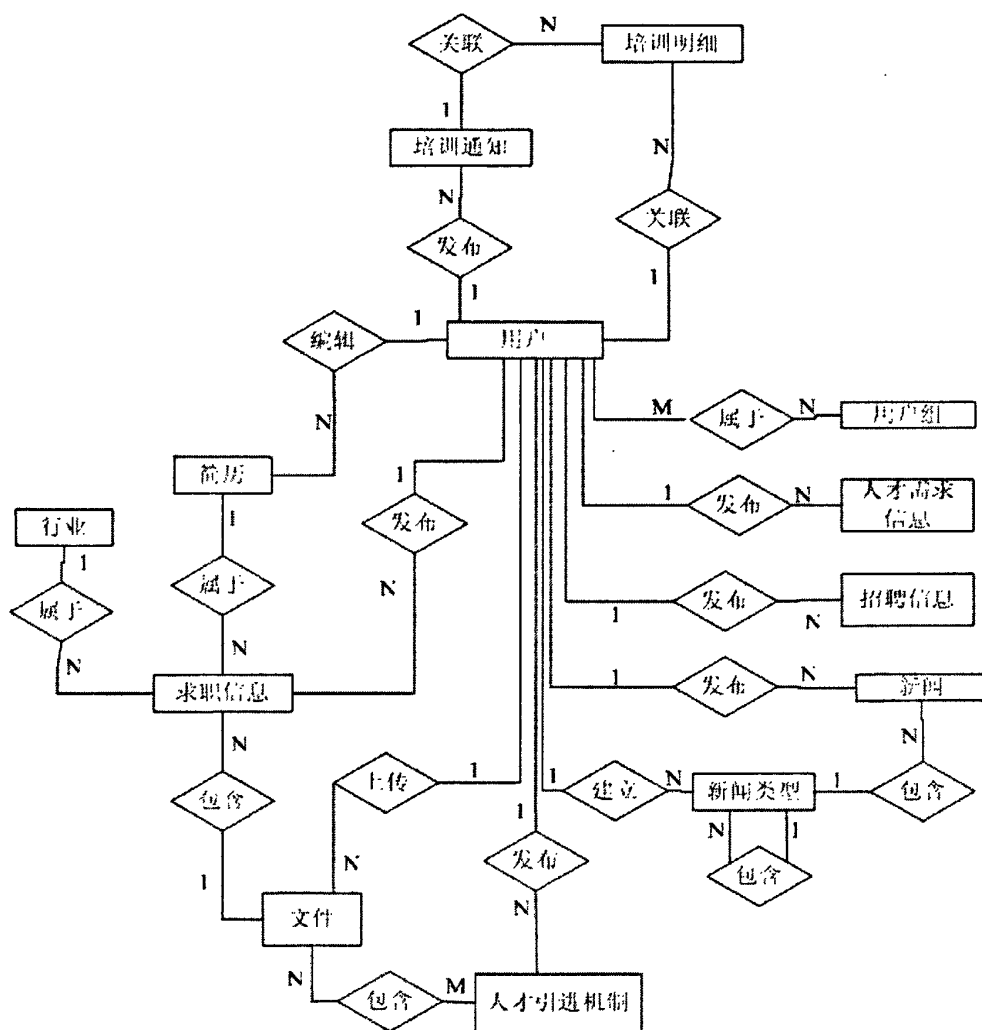


图 3-8 人才引进及求职服务数据库 E-R 图

3.5.2 数据库逻辑结构设计

概念结构是独立于任何一种数据模型的信息结构。逻辑结构设计的就是把概念结构设计阶段设计好的基本 E-R 图转换为与选用 DBMS 产品所

支持的数据模型相符合的逻辑结构^[20]。根据上一小节的数据库概念结构设计，可以设计出人才引进及培训子系统的数据库表，下面对部分表结构进行描述。

表 3-1 用户基本信息表

列名	字段类型	说明	备注
Id	BIGINT	序号	主键
loginName	NVARCHAR(50)	登陆名	不为空
realName	NVARCHAR(50)	真实名称	不为空
password	VARCHAR(50)	密码	不为空
lastLoginTime	DATETIME	最近登陆时间	可为空
lastLoginIP	VARCHAR(20)	最近登陆 IP	可为空
loginTimes	INT	登陆次数	可为空
address	NVARCHAR(100)	地址	可为空
telephone	VARCHAR(40)	电话	可为空
fax	VARCHAR(40)	传真	可为空
email	VARCHAR(100)	电子信箱	可为空
zipcode	VARCHAR(10)	邮政编码	可为空
description	TEXT	用户描述	可为空
website	VARCHAR(100)	个人网址	可为空
isLock	BIT	是否锁定	不可为空
userBy1	VARCHAR(50)	扩展字段 1	可为空
userBy2	TEXT	扩展字段 2	可为空

用户表用于存储用户的基本信息。其中 isLock 字段表示用户是否被锁定，锁定的用户不能登陆。userBy1 与 userBy2 字段用于扩展，如果由于业务变化，用户信息表需要增加某个字段而要对现有应用影响较小，如不需要改动 hibernate 映射配置文件，兼顾节约存储空间与可扩展性，这里添加两个字段。

表 3-2 用户组表

列名	字段类型	说明	备注
Id	INT	序号	主键
groupName	NVARCHAR(50)	用户组名称	不为空
groupDesc	TEXT	用户组描述	可为空

groupFunction	VARCHAR(2000)	功能字符串	不为空
createDate	DATETIME	创建时间	不为空
creatorId	BIGINT	创建者 ID	外键
isDelete	BIT	组能否被删除	不为空
userBy1	VARCHAR(50)	扩展字段 1	可为空
userBy2	TEXT	扩展字段 2	可为空

用户组表存储用户组信息，其中 groupFunction 字段用于存储权限值。其值为 0 或者 1 的字符串，长度为 2000。在进行权限管理时，字符串中位的值对应特定资源是否授权，其中 1 代表有权限访问，0 代表无权限访问。

表 3-3 用户与组关联表

列名	字段类型	说明	备注
Id	BIGINT	序号	主键
userId	BIGINT	用户 ID	外键
groupId	INT	用户组 ID	外键

用户与用户组关联表存储用户所属于的用户组信息。其中一个用户可以属于多个用户组。userId 与用户基本信息表的 id 字段关联，groupId 与用户组表的 id 值关联。

表 3-4 人才简历表

字段名	字段类型	说明	备注
id	INT	序号	主键
name	VARCHAR(50)	姓名	不为空
sex	BIT	性别	不为空
politicalStatuts	VARCHAR(30)	政治面貌	可为空
city	VARCHAR(50)	所在城市	不为空
province	VARCHAR(50)	所在省份	不为空
origin	VARCHAR(100)	籍贯	不为空
education	VARCHAR(50)	学历	可为空
graduateSchool	VARCHAR(50)	毕业学校	可为空
major	VARCHAR(50)	专业	可为空
intention	VARCHAR(50)	就业意向	可为空
intentioncity	VARCHAR(50)	意向城市	可为空
intentionsalary	VARCHAR(50)	工资要求	可为空

detail	TEXT	详细信息	可为空
userid	BIGINT	用户 ID	外键

表 3-5 个人求职信息表

字段名	字段类型	说明	备注
id	INT	序号	主键
userId	BIGINT	关联用户 ID	外键
title	VARCHAR(100)	求职标题	不为空
position	VARCHAR(255)	求职职位	不为空
typeId	VARCHAR(50)	所选行业号	外键
jobType	VARCHAR(50)	工作类别	不为空
workaddr	VARCHAR(50)	期望工作地点	不为空
salaryNeed	VARCHAR(50)	薪水要求	可为空
occupationIntention	VARCHAR(255)	求职意向	可为空
selfEvaluation	TEXT	自我评价	可为空
publishTime	DATETIME	发布日期	不为空
expireDate	DATETIME	截止日期	不为空
resumeId	INT	简历 ID	外键
fileId	INT	文档 ID	外键

个人求职信息表用户存储用户发表的个人求职信息，表中 userId 关联用户表中的 id，用户表中存储了用户的基本信息，当授权用户查看用户的个人求职信息时，除能查看求职的基本信息，还能够查看到个人用户的基本信息。resumeId 关联简历表的 id，用于个人求职信息与用户存储在简历表中的简历，而 fileId 是用户的简历文档，是用户上传到服务器的简历。

表 3-6 招聘信息表

字段名	字段类型	说明	备注
id	INT	序号	主键
userId	BIGINT	关联用户 ID	外键
title	VARCHAR(100)	招聘名称	不为空
position	VARCHAR(255)	职位名称	不为空

positionType	VARCHAR(50)	职位类别	不为空
positionRequire	VARCHAR(200)	职位要求	可为空
workaddr	VARCHAR(100)	工作地点	可为空
numberRequire	INT	招聘人数	可为空
educationNeed	VARCHAR(50)	学历要求	可为空
positionDesc	VARCHAR(255)	职位描述	可为空
expireDate	DATETIME	截至日期	不为空
commitDate	DATETIME	提交日期	不为空
details	TEXT	详细信息	可为空

表 3-7 文件信息表

字段名	字段类型	说明	备注
id	INT	序号	主键
name	VARCHAR(100)	文件名称	不为空
type	VARCHAR(20)	文档类型	不为空
serialnumber	VARCHAR(50)	序列号	不为空
filePath	VARCHAR(100)	存放路径	不为空
createTime	DATETIME	创建时间	不为空
createUserId	BIGINT	创建者 ID	外键
updateTime	DATETIME	更新时间	可为空
updateUserId	BIGINT	更新者 ID	外键

文件信息表用于存储各种附件，法律法规文档，用户简历等。其中 name 是上传时文件名称，serialnumber 是存储到服务器上的文档名称，其值根据上传时的时间生成(精确到毫秒)。

表 3-8 文章信息表

字段名	字段类型	说明	备注
id	INT	INT	主键
createUserId	BIGINT	创建者 ID	外键
createTime	DATETIME	发布时间	不为空
title	VARCHAR(100)	题目	不为空
titleColor	VARCHAR(20)	题目颜色	可为空
from	VARCHAR(100)	文章来源	可为空

fromUrl	VARCHAR(200)	链接地址	可为空
hits	INT	点击次数	可为空
content	TEXT	文章内容	不可为空
catalogId	INT	所属类别 ID	外键
updateUserId	BIGINT	更新用户 ID	外键
updateTime	DATETIME	更新日期	可为空

表 3-9 文章类型信息表

字段名	字段类型	说明	备注
id	INT	INT	主键
createUserId	BIGINT	创建者 ID	外键
createTime	DATETIME	创建时间	不为空
parentId	INT	父类型 ID	外键
name	VARCHAR(20)	类型名称	不为空
desc	VARCHAR(200)	类型描述	不为空

文章类型信息表用户存储如新闻，法律法规等文章的类型。一种文章类型可能属于另外一种文章类型。其结构是树形的，树的结构由 id 与 parentId 来进行关联。

第 4 章 系统的实现与讨论

本章在上一章系统需求分析与设计的基础之上, 利用 Struts, Spring 及 Hibernate 整合集成架构, 对系统进行详细设计及讨论。本章首先介绍系统运行环境和开发工具, 然后介绍了系统框架的总体实现, 最后分别从持久层, 业务层, 视图层出发介绍系统的详细设计过程。

4.1 系统运行环境和开发工具

- 操作系统: Windows 2003 Server
- Web 服务器: Tomcat 5.5.20
- 数据库服务器: Microsoft Sql Server 2005
- Java 编译器: JDK 1.5.0_0
- Struts: Struts 1.2
- Hibernate: Hibernate 3.1
- Spring: Spring 2.0
- 数据库驱动程序: jtds 1.2
- 开发工具: Eclipse 3.3 + MyEclipse 6.0; EditPlus 2.12
- 系统建模工具: Rational Rose 2003
- 数据库设计工具: PowerDesigner 12.5

4.2 系统框架实现

中小企业创新服务平台系统采用 Struts+Spring+Hibernate 的开源框架组合来开发, 采用 J2EE 分层架构, 系统整体上可以划分为持久层、业务逻辑层与视图层。其中表示层基于 Struts MVC 来实现, 业务逻辑层基于 Spring 来实现, 而数据持久化层基于 Hibernate 来实现。这种分层结构能降低层与层之间的耦合度, 上层通过接口与下层通信, 下层向上层提供服务, 而上层不需要知道下层服务的具体实现, 这样使得系统容易维护和扩展。系统框架组件图如图 4-1 所示:

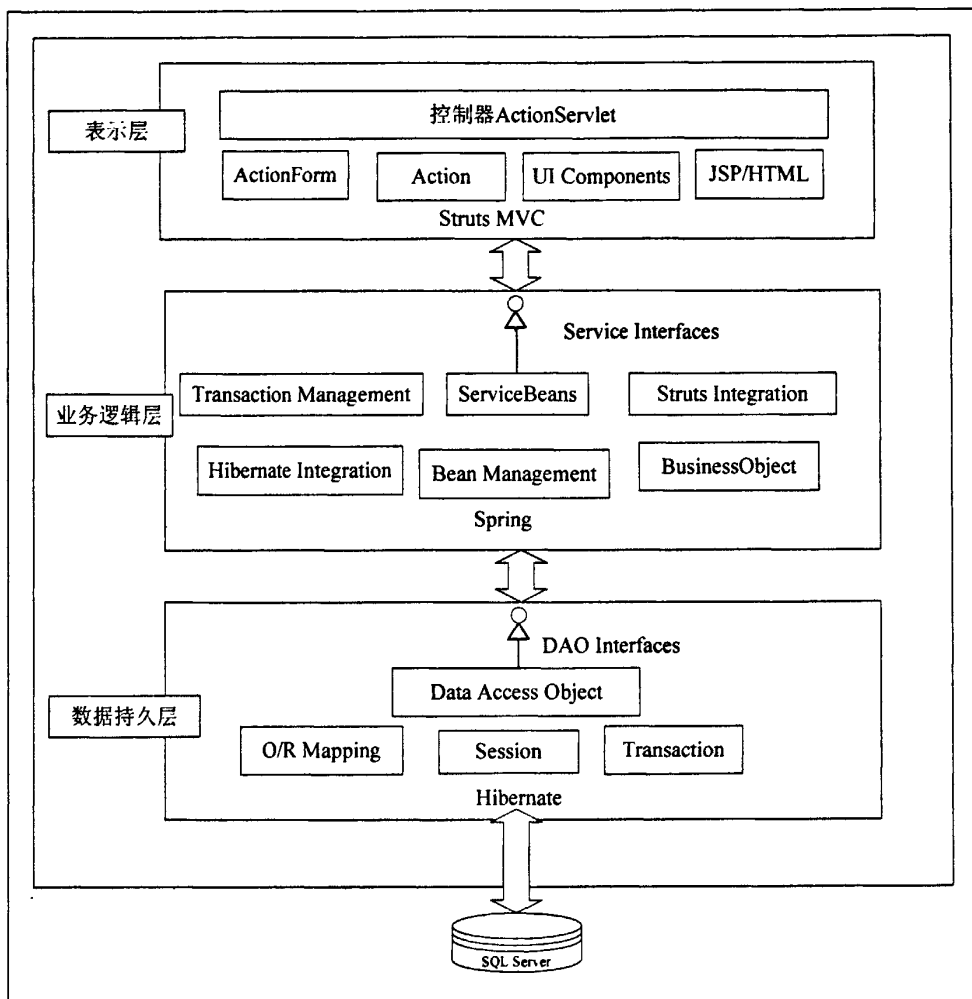


图 4-1 系统框架组件图

本文在后面将会对系统中的三层结构实现进行详细的介绍。

4.3 数据持久层的实现

持久化层是在系统逻辑层面上专注于实现数据持久化的一个相对独立的领域。与系统其他部分相对而言，持久化层封装了数据访问细节，为业务逻辑层提供了面向对象的 API，拥有一个较为清晰和严格的逻辑边界^[21]。

采用 Hibernate 进行开发，可以大大缩短开发过程的周期，使得系统数据持久层的结构清晰。另外，Spring 框架对 Hibernate 提供了良好的集成接口，其本身也对 Hibernate 操作数据库进行了一些封装，能大大提高开发效率。

4.3.1 数据库连接及相关配置

系统中数据库连接由 Spring 来管理,而 SessionFactory 是单个数据库经映射辨析后的内存镜像,是业务逻辑对数据库进行操作的基础。下面是两者在 Spring 配置文件中详细配置代码片断:

```
<bean id="propertyConfigurer"
    class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
    <property name="location">
        <value>classpath:jdbc.properties</value>
    </property>
</bean>
<bean id="dataSource"
    class="org.apache.commons.dbcp.BasicDataSource">
    <property name="driverClassName">
        <value>${jdbc.driverClassName}</value>
    </property>
    <property name="url">
        <value>${jdbc.url}</value>
    </property>
    <property name="username">
        <value>${jdbc.username}</value>
    </property>
    <property name="password">
        <value>${jdbc.password}</value>
    </property>
</bean>
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <property name="dataSource">
        <ref bean="dataSource" />
    </property>
```

```
<property name="mappingDirectoryLocations">
    <list>
        <value>classpath:</value>
    </list>
</property>
<property name="hibernateProperties">
    <props>
        <prop key="hibernate.dialect">
            org.hibernate.dialect.SQLServerDialect
        </prop>
        <prop key="hibernate.cache.provider_class">
            org.hibernate.cache.EhCacheProvider
        </prop>
        <prop key="hibernate.cache.use_query_cache">
            true
        </prop>
        <prop key="hibernate.show_sql">false</prop>
    </props>
</property>
</bean>
```

程序中以依赖注入的方式生成 SessionFactory。程序首先根据配置的具体连接配置文件位置：位于 classpath 路径下的 jdbc.properties 文件。jdbc.properties 文件中配置了详细的数据库连接，其配置以键值对的方式。在项目部署时，如具体数据库，驱动程序改变只修改这个文件即可。下面是 jdbc.properties 文件的详细信息：

```
jdbc.driverClassName=net.sourceforge.jtds.jdbc.Driver
jdbc.url=jdbc:jtds:sqlserver://221.10.44.**:1433;DatabaseName=
tibet_innovation
jdbc.username=*****
jdbc.password=*****
jdbc.defaultAutoCommit=true
hibernate.jdbc.batch_size=25
```

```
hibernate.jdbc.fetch_size=50
```

在获得数据库连接配置信息后，Spring 将生成数据源 `dataSource`，并将其注入 `sessionFactory` 这个 bean 中最后生成具体的 `SessionFactory` 实例。在随后各个 `xxxDao` 实例中也以依赖注入的方式获得 `SessionFactory` 实例，从而进行具体的数据库操作。

4.3.2 事务配置

“事务”是一个逻辑工作单元，它包括一系列的操作。事务包含 4 个基本特性，也就是 ACID。在系统开发中，事务管理是非常重要的。在本系统中，采用 Spring 的声明式事务来实现系统的事务管理，这种事务管理的优势非常明显：代码中无需关于关注事务逻辑，只需让 Spring 声明式事务管理负责事务逻辑，无需与具体的事务逻辑耦合，可以方便地在不同事务逻辑之间切换。

下面是通过 Spring 的 AOP 来实现系统事务管理的配置示例：

```
<bean id="transactionManager"
    class="org.springframework.orm.hibernate3.HibernateTransactionManager">
    (1)
    <property name="sessionFactory">
        <ref bean="sessionFactory" />
    </property>
</bean>

<bean id="txProxyTemplate" abstract="true"
    class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
    (2)
    <property name="transactionManager">
        <ref bean="transactionManager" />
    </property>
    <property name="transactionAttributes">
        <props>
            <prop key="save*">PROPAGATION_REQUIRED</prop> (3)
            <prop key="remove*">PROPAGATION_REQUIRED</prop>
            <prop
```

```
key="*">PROPAGATION_REQUIRED,readOnly</prop>
    </props>
</property>
</bean>
<bean id="businessBoardService" parent="txProxyTemplate">
    <property name="target"> (4)
        <bean class="tibet.service.impl.
            BusinessBoardServiceImpl">
            <property name="businessBoardDAO"
                ref="businessBoardDAO" />
        </bean>
    </property>
</bean>
```

下面对代码中注释的地方说明如下：

(1) 由于系统中使用 Hibernate 作为 ORM 工具，因此使用 Hibernate 特定的管理器 HibernateTransactionManager。

(2) 使用 TransactionProxyFactoryBean 定义事务处理模板。

(3) 定义事务策略：将所有以 save、remove 开始的方法纳入事务管理范围，如果这些方法抛出异常，那么 Spring 将当前事务回滚，如果方法正常结束，则提交事务，而对其他所有的方法则以只读的事务处理机制进行。

(4) 设置纳入事务管理的具体 Service 类，其中 Service 类通过依赖注入引用了对应的 Dao 类。

4.3.3 Hibernate 缓存的配置

缓存是位于应用程序与物理数据源之间，用于临时存放复制数据的内存区域，目的是为了减少应用程序对物理数据源访问的次数，从而提高应用程序的运行性能。Hibernate 在查询数据时，首先到缓存中去查找，如果找到就直接使用，找不到的时候就会从物理数据源中检索，所以，把频繁使用的数据加载到缓存区后，就可以大大减少应用程序对物理数据源的访问，使得程序的运行性能明显的提升^[22]。

Hibernate 数据缓存分为：内部缓存(Session Level，也称为一级缓存)，它是属于事务范围内的缓存，这一级别的缓存由 Hibernate 管理，一般情况

下无需干预;二级缓存(SessionFactory Level, 也称为二级缓存),它是属于进程范围或群集体范围的缓存,这一级别的缓存可以进行配置和更改,并且可以动态加载和卸载^[23]。由于一级缓存会由 Hibernate 自行管理,下面对系统中运用的二级缓存进行简单的介绍:

二级缓存机制应用策略需要慎重考虑,不恰当的缓存机制会给系统运行性能带造成很严重的影响。下面是适合放入二级缓存中的数据:1, 很少被修改的数据 2, 不是很重要的数据, 允许出现偶尔并发的数据 3, 不会被并发访问的数据 4, 参考数据, 指供应用参考的常量数据^[24]。

Hibernate 提供了面向第三方的缓存实现接口, 本系统中运用 EhCache 来实现二级缓存。

❖ 首先建立 encache.xml, 放到 class-path 下, 配置如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<ehcache>
  <diskStore path="e:\\ehcache\\"/>
  <defaultCache maxElementsInMemory="10000" <!-- 缓存最大数目 -->
    eternal="false" <!-- 缓存是否持久 -->
    overflowToDisk="true" <!-- 当系统当机时,是否保存到磁盘 -->
    timeToIdleSeconds="300" <!-- 当缓存闲置 n 秒后销毁 -->
    timeToLiveSeconds="180" <!-- 当缓存存活 n 秒后销毁-->
    diskPersistent="false"
    diskEXPIRYThreadIntervalSeconds= "120"/>
</ehcache>
```

❖ 在 Spring 配置文件 hibernate 部分加入如下配置:

```
<prop key="hibernate.cache.provider_class">
  org.hibernate.cache.EhCacheProvider
</prop>
<prop key="hibernate.cache.use_query_cache">true</prop>
```

❖ 在对应实例类的映射文件***.hbm 中加入:

```
<cache usage="read-only"/>
```

❖ 在对应 DAO 文件中与数据库交互时, 加入如下代码:

```
getHibernateTemplate().setCacheQueries(true);
```

4.3.4 Hibernate 对象关系映射配置

设计数据库之后,需创建与表对应的域对象类,每个域对象类都是具有 getter/setter 方法的 JavaBean,另外域对象须实现 java.io.Serializable 接口。在有了表跟域对象之后,须建立映射文件将表跟域对象类关联起来。下面以个人简历 TibetRenResume 表与 TibetRenResume 域对象类关联的映射文件为例来介绍映射配置:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd" >

<hibernate-mapping package="tibat.entity">
<class name="TibetRenResume" table="TibetRenResume"> (1)
    <meta attribute="sync-DAO">false</meta>
    <id name="Id" type="integer" column="id">
        <generator
class="org.hibernate.id.IncrementGenerator"/> (2)
    </id>
    <property name="Name" column="name" type="string"
        not-null="false" length="20"/> (3)
    .....<!--其他属性省略-->
    <set name="TibetRenWorkexperiences" cascade="all" (4)
        inverse="true" lazy="false">
            <key column="workerid"/>
            <one-to-many class="TibetRenWorkexperience"/>
        </set>
    <many-to-one name="tibetUsers" class="TibetUsers" (5)
        column="userid" unique="true"/>
</class>
</hibernate-mapping>
```

下面对代码中标注的地方作以下说明:

- (1)说明了表 TibetRenResume 与域对象 TibetRenResume 的映射关系。
- (2)说明了表 TibetRenResume 的主键为 id。主键值由 Hibernate 的内置主

键生成器来生成。

(3)声明表的 **name** 字段与域对象的 **name** 属性对应,并说明该字段在表中可以为空,其长度为 20。

(4)声明了一个一对多的属性:一份简历可以对应多个工作经历。该结点设置了级联方式,一对多管理方,懒加载及关联类。

(5)声明了一个多对一的属性:一个用户可以对应多份简历。

4.4 业务逻辑层的实现

在整个 J2EE 系统中,处在中间部分的组件称为业务逻辑层,也称为 Service 层,负责实现业务逻辑。系统中采用 Spring 框架来构建系统的业务逻辑层, Spring 框架提供了依赖注入来管理系统中运用到的 action,service,dao 等 bean,利用 IoC 来将容器中所需要的对象联系在一起。通过 Spring 的这种特性,能降低系统各层之间的耦合度,有利于系统的开发维护。

4.4.1 Spring 相关配置

首先在 web.xml 文件里配置一个监听器以及配置文件的路径,这样在服务器启动时就可以加载 Spring 文件并初始化,详细代码如下:

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
        /WEB-INF/applicationContext.xml,
        /WEB-INF/applicationContext-commonService.xml,
        /WEB-INF/applicationContext-cxcgzh.xml
    </param-value>
</context-param>
<listener>
    <listener-class>
        org.springframework.web.context.ContextLoaderListener
    </listener-class>
</listener>
```

从代码中可以看到, Spring 的配置文件可以分段放在不同的配置文件中,

这样有利于系统分工开发，同时也可以使系统代码结构更加清晰。

4.4.2 业务逻辑层详细实现

系统中 Service 层的基础是 `GenericsServiceImpl` 这个类，这个类实现了 `GenericsService` 接口，`GenericsService` 接口定义了一般业务类中常用方法：包括与底层数据库交互的通用方法。因为具体的业务类继承 `GenericsServiceImpl`，因此该接口与实现类中都以范型的方式定义。具体的业务类 `xxxServiceImpl`(实现了自定义接口 `xxxService`)继承 `GenericsServiceImpl`，同时与持久层交互时会用到 `GenericsServiceImpl` 中定义的具体的 `xxxDao` 接口，而 `xxxDao` 在 Spring 中以依赖注入的方式传递给 `GenericsServiceImpl`。

下面是 `GenericsService` 接口代码片断：

```
public interface GenericsService<T, ID extends Serializable>{  
    T findById(ID id);  
    List<T> findByCriteria(DetachedCriteria dc);  
    void remove(T entity);  
    PageData<T> findByPage(final DetachedCriteria  
detachedCriteria, final int pageSize, final int startIndex);  
    //其他方法省略  
}
```

下面是 `GenericsServiceImpl` 接口代码片断：

```
public class GenericsServiceImpl<T, ID extends Serializable, DAO  
extends GenericsDAO<T, ID>>  
    implements GenericsService<T, ID> {  
    private DAO dao;  
    public List<T> findAll() {  
        return dao.findAll();  
    }  
    public T findById(ID id) {  
        return dao.findById(id);  
    }  
    public List<T> findByNameQuery(String query, Object[]
```

```
parameters) {  
    return dao.findByNameQuery(query, parameters);  
    //其他方法省略  
}
```

分析上面代码可以看到，GenericsServiceImpl 中实现的与数据库交互的方法是通过调用依赖注入的具体 xxxDao 来实现的。代码中参数，返回类型都是以范型的方式存在。具体的 xxxDao 继承自 GenericsDAOImpl 类，该类中定义了与数据库交互的增，删，改，查等操作，这些操作大部分是通过 Spring 封装了 Hibernate 一般操作方法的 HibernateTemplate 类来完成的。下面是 GenericsDAOImpl 类代码的一部分：

```
public class GenericsDAOImpl<T, ID extends Serializable> extends  
    HibernateDaoSupport implements GenericsDAO<T, ID> {  
    private Class<T> persistentClass;  
    public List<T> findAll() {  
        return  
getHibernateTemplate().loadAll(getPersistentClass());  
    }  
    public List<T> findByCriteria(final DetachedCriteria dc) {  
  
        return (List<T>) getHibernateTemplate().execute(new  
HibernateCallback() {  
            public Object doInHibernate(Session session)  
                throws HibernateException, SQLException {  
                Criteria c = dc.getExecutableCriteria(session);  
                c.setCacheable(true);  
                return c.list();  
            }  
        }, true);  
    }  
    //其他方法省略  
}
```

在具体的实现方法 xxxServiceImpl 中，只需要添加一个对应 Dao 的 setter

方法即可，而具体注入由 Spring 来进行管理。由 xxxServiceImpl 类要与对应的 xxxDaoImpl 关联，而且要纳入 Spring 事务管理。另外，具体的 xxxDaoImpl 还要与 sessionFactory 进行关联，下面以热点调查管理为例来说明 Spring 中如何进行配置：

```
<bean id="xzSurveyInfoDao"
      class="tibet.dao.impl.XzSurveyInfoDaoImpl">
    <property name="sessionFactory">
      <ref bean="sessionFactory" />
    </property>
</bean>
//service 配置，以 aop 纳入 Spring 事务管理中
<bean id="xzSurveyInfoService" parent="txProxyTemplate">
  <property name="target">
    <bean
class="tibet.service.impl.XzSurveyInfoServiceImpl">
      <property name="xzSurveyInfoDao" ref="xzSurveyInfoDao"
/>
    </bean>
  </property>
</bean>
```

4.5 表现层的实现

Struts 本身提供了一组相互协作的类，Servlet 和 JSP 标记，同时也包含了大量的独立于该框架工作的实用程序类库和丰富的标记库。Struts 通过 ActionServlet 核心控制器配合 struts-config.xml 配置文件来实现对整个系统导航。增强了开发人员对系统的整体把握，提高了系统的可维护性和可扩充性。

4.5.1 Struts 相关配置

Struts 配置文件包括在 web.xml 配置和自身配置文件 struts-config.xml，当然为了使各功能模块更加清晰，自身配置文件也可以分为多个。

在 web.xml 中配置 Struts 如下所示：

```
<servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>
        org.apache.struts.action.ActionServlet
    </servlet-class>
    <init-param>
        <param-name>config</param-name>
        <param-value>
            /WEB-INF/struts-config.xml,
            /WEB-INF/struts-config-cxcgzh.xml,
            /WEB-INF/struts-config-cservice.xml,
        </param-value>
    </init-param>
    <load-on-startup>0</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

由于 Spring 贯穿整个系统的应用，本系统中使用 Spring 中的 DelegatingRequestProcessor 覆盖 Struts 的 RequestProcessorSpring 来管理 action，这需要在 Struts 的配置文件中加入以下代码：

```
<controller processorClass="org.springframework.web.struts.
DelegatingRequestProcessor" />
```

通过以上配置，系统会用 DelegatingRequestProcessor 来代替原来的 RequestProcessor，事实上 DelegatingRequestProcessor 继承自 RequestProcessor，这样 Struts 会将拦截到的用户请求转发 Spring 上下文管理的 bean 中。

4.5.2 视图层详细实现

在系统的应用场景中，客户端会根据业务逻辑提交类似***.do 的请求。DelegatingRequestProcessor 会截获请求，通过依赖注入，生成相应的的 bean

类实例。因为系统中 action 继承自 DispatchAction, 在实际客户端请求的 URL 中根据不同的 method 会转到 action 中对应的方法, 方法执行完毕后, 然后通过返回的 ActionForward 的实例, 系统会根据 Struts 中的配置转入到定义的 JSP 页面或其他 action 等。

传统的 ActionForm 用来封装用户请求, 通常与表单页面对应, 请求参数是通过 JSP 页面的表单域传过来, 每个表单域对应 ActionForm 的一个属性。本系统中使用更为方便的 LazyValidatorForm, 这样就不用再需要单独定义 ActionForm, 也不需要 Struts 配置文件里详细定义每个 FormBean 的属性, 减少了工作量。

下面仍然以热点调查为例来说明视图层的工作流程:

struts-config.xml 中配置文件关于热点调查配置片断:

```
<action name="dynaForm" path="/survey" scope="request"
    parameter="method"
    type="org.springframework.web.struts.DelegatingActionProxy"
        validate="false">
    <forward name="adminList"
        path="/tibetPages/admin/sysManage/admin_survey_list.jsp" />
        <forward name="goUpdateSubjectPage"
            path="/tibetPages/admin/sysManage/admin_vote_edit.jsp" />
            <forward name="goUpdateItemPage"
                path="/tibetPages/admin/sysManage/admin_voteOption_edit.jsp" />
            <forward name="goSurveyList" path="/tibetPages/survey.jsp"
        />
        <forward name="subjectDetail"
            path="/tibetPages/vote_result.jsp" />
    </action>
```

配置文件中的 action 节点中的 name="dynaForm", 其中 dynaForm 是引用的一个 LazyValidatorForm。dynaForm 配置如下:

```
<form-bean name="dynaForm"
    type="org.apache.struts.validator.LazyValidatorForm" />
```

当客户端有如 `***survey.do` 请求到达时, Spring 管理的 action 进入该 action

中执行指定的方法，如管理员增加新的热点调查执行 addSubject 方法，如下所示：

```
public class SurveyAction extends DispatchAction {
    private XzSurveyHeadService xzSurveyHeadService;
    private XzSurveyInfoService xzSurveyInfoService;
    public void setXzSurveyHeadService(XzSurveyHeadService
xzSurveyHeadService) {           //(1)
        this.xzSurveyHeadService = xzSurveyHeadService;
    }
    public void setXzSurveyInfoService(XzSurveyInfoService
xzSurveyInfoService) {
        this.xzSurveyInfoService = xzSurveyInfoService;
    }
    public ActionForward addSubject(ActionMapping mapping,
ActionForm form,
        HttpServletRequest request, HttpServletResponse
response) {
        //判断是否有操作权限
        if (!PrivilegeHelper.hasPrivilege(32, request)) { //(2)
            request.setAttribute("message", "您访问的页面不存在");
            return new ActionForward("/info.jsp");
        }
        XzSurveyHead surveyHead = new XzSurveyHead();
        String type = request.getParameter("type");
        String isShow = request.getParameter("isShow");
        String title = request.getParameter("title");
        surveyHead.setTitle(title);
        surveyHead.setType(type);
        if (isShow.equals("1"))
            surveyHead.setIsShow(true);
        else
            surveyHead.setIsShow(false);
    }
}
```

```
surveyHead.setStartDate(new Date());  
xzSurveyHeadService.saveOrUpdate(surveyHead);    //(3)  
request.setAttribute("message", "添加成功");  
return new ActionForward("/survey.do?method=adminList");  
//(4)  
}
```

//其他的方法省略}

下面对上面代码标注的地方加以解释:

- (1) action 中需要的 service 是通过 setter 方式由 Spring 注入的。
- (2) 系统中用户权限配置, 执行方法之前检查是否具有此权限。
- (3) 业务方法执行完毕后持久化到数据库中。
- (4) 整个 action 方法执行完毕之后, 返回到一个 action 中执行指定的方法。

当 action 方法执行完毕后, 在本系统中, 最终会根据 Struts 的配置文件跳转到相应的 JSP 页面。

图 4-2, 图 4-3, 图 4-4 分别为运行后系统后台权限管理主界面, 平台投票管理界面以及个人求职信息发表界面。

添加用户组		
名称:	企业用户	
描述:	企业用户	
用户组权限设置		
权限名称	权限状态	权限描述
是否允许用户登录	是 ☒ 否 ☐	该权限表明用户是否有在后台登录的功能
是否分配管理员	是 ☐ 否 ☒	该权限表明用户是否有后台管理的功能
是否可以修改所有用户密码	是 ☐ 否 ☒	是否可以修改所有用户密码 一般分配给管理员!
是否可以操作 信息管理->新闻中心 一般分配给管理员	是 ☐ 否 ☒	是否可以操作新闻中心
是否可以操作 信息管理->政策法规	是 ☐ 否 ☒	是否可以操作政策法规 一般分配给管理员
是否可以操作 信息管理->站内通知通告	是 ☐ 否 ☒	是否可以操作站内通知通告 一般分配给管理员
是否可以访问用户管理	是 ☐ 否 ☒	是否可以访问用户管理 一般分配给管理员
是否可以访问系统公共服务	是 ☐ 否 ☒	是否可以访问系统公共服务
是否可以访问知识版权	是 ☐ 否 ☒	是否可以访问知识版权
是否可以访问技术标准	是 ☐ 否 ☒	是否可以访问技术标准

图 4-2 后台权限管理主界面

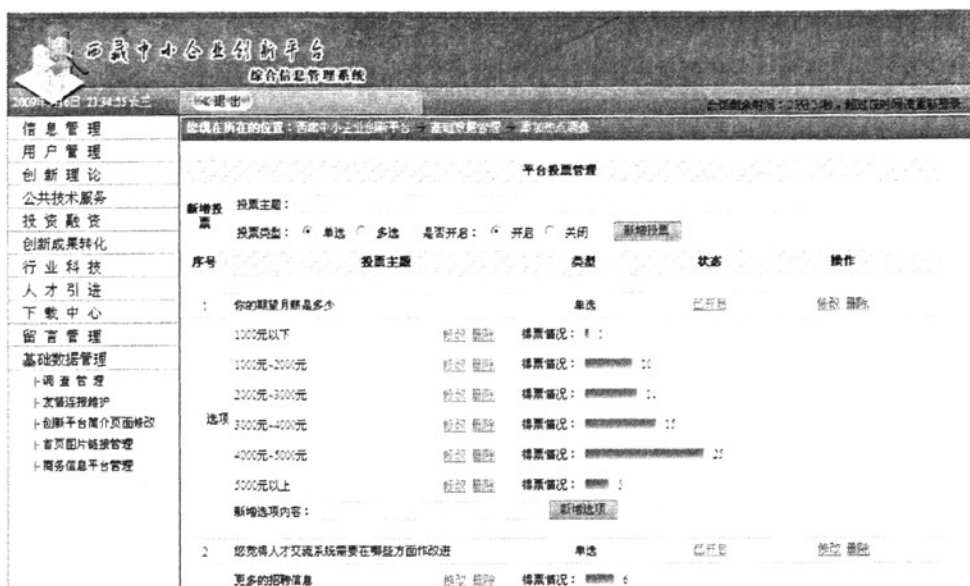


图 4-3 平台投票后台管理

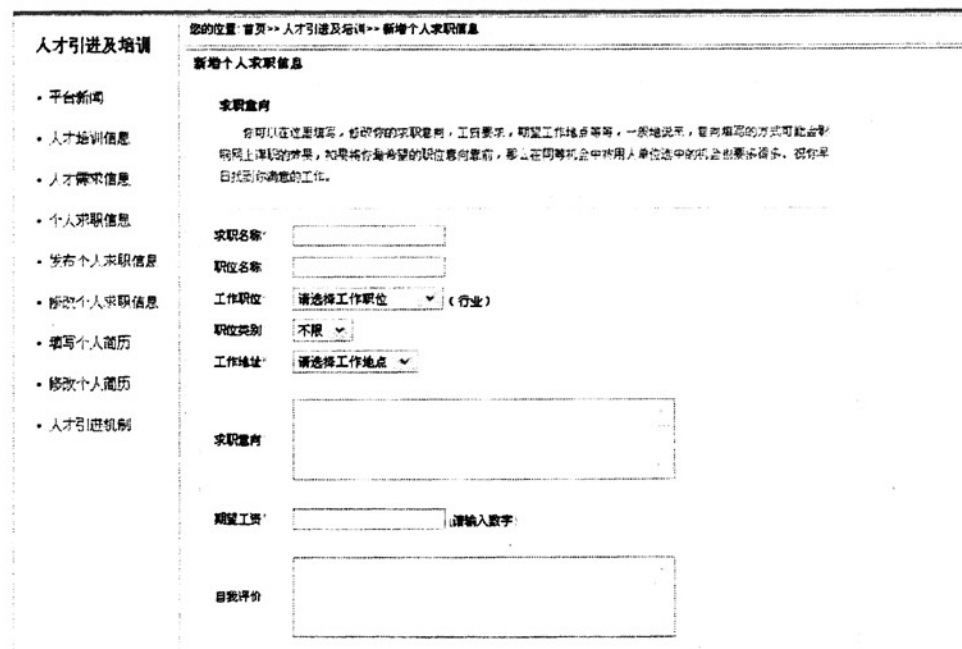


图 4-4 个人求职信息发布界

第 5 章 总结与展望

各种开源框架技术的发展,大大地推动了软件的发展,提高了软件的可复用性,缩短了软件的开发周期。随着软件规模的不断扩大,使得软件架构的复杂度越来越高,传统开发越来越难以适应这一要求。本文通过 Struts, Spring 与 Hibernate 的集成架构,结合某自治区科技厅生产力促进中心中小企业创新服务平台信息系统的开发,讨论了 Struts 的 MVC, Spring 对 Struts, Hibernate 的支持, Spring 的 IoC 与 AOP, 另外还讨论了优秀的 ORM 框架 Hibernate, 给出了一个服务平台系统的详细实现。总的来说,完成了课题最初的设想跟要求,为今后工作打下了良好的基础。总结如下:

- 对系统中采用的 Struts, Hibernate, Spring 等框架技术进行了较深入的研究,并在此基础上对这三个框架的集成架构进行探讨。
- 对系统需求分析和建模有了更深刻的认识。
- 通过实际项目的开发,对企业级应用系统的测试及性能优化有了一定的了解。

项目完成后,通过压力测试及其他性能测试,表现良好,但是由于平台功能的复杂性以及研究水平的所限,论文和系统还存在着一些问题,主要体现在:

首先系统各子系统功能还不尽完善,需要进一步改进升级,如投资融资子系统中,需要添加企业通过第三方担保机构来从银行融资服务功能,以及融资中需要参考的企业信用等级评定系统等。其次平台的安全性需要进一步完善,使系统能够防止恶意破坏和非法使用。另外平台的缓存及事务需要更合理的实现,使系统的性能能够进一步的提升。更重要的是,由于中小企业创新发展是一个长期的过程,涉及到的领域非常广,而相关部门在促进中小企业创新发展所需要提供的服务也会随着中小企业的发展变得越来越丰富,到一定阶段,需要与相关部门的平台集成会变得很。这些都是作者今后继续深入研究的重点。

致谢

本硕士论文的顺利完成，凝聚了许多老师和同学的关心和帮助，在此对他们表示深深的谢意。在此特向我的导师赵慧娟教授致以最崇高的敬意和衷心的感谢。读研期间，赵老师无论从生活上还是科研上都给了我很多受益匪浅的教诲和指导。她学识渊博、治学严谨、工作敬业，使我受益终身。正是在赵老师耐心细致的指点和教导下才使这篇论文得以顺利完成。在研究生学习期间，我一直得到了赵慧娟教授、孙林夫教授、周荣辉教授、唐慧佳教授、聂绪海老师、王淑营老师、韩敏老师的谆谆教导和大力帮助，他们无论在生活上还是在学习上都给予了我很大的帮助，正是在他们无私的指导和关怀下，我的论文才能顺利完成，在此向各位老师表示衷心的感谢。

在学习和工作期间，得到了许多同学的帮助，在这个集体中，我们相互学习，共同进步。感谢在读研期间一起奋斗的所有同学，正是他们在科研中的帮助和支持，才使我的科研和论文工作顺利的完成。同时，他们在平时的生活中也给予了我很多的关怀，使我的研究生生活丰富多彩，在此表示衷心的感谢。我还要向我的父母、亲人、朋友以及共同工作和学习的同事致以深深的谢意，谢谢你们对我学业工作的支持和帮助。

最后，我衷心感谢为评阅本论文付出辛勤劳动的专家、教授和老师们！

参考文献

- [1] 濮敏.关于中小企业技术创新服务体系建设的探讨.现代管理科学.2007 年第五期
- [2] 王天达.浅析中小企业创新.河北煤炭.2008 年第 4 期
- [3] 徐晓菊.我国中小企业创新现状与存在的问题.中国统计.2008 卷 5 期
- [4] 刘小川.论中国科技型中小企业发展的政策推动.南京大学学报(哲学·人文科学·社会科学).2002 年,第 6 期第 39 卷:第 137 页
- [5] 周彩虹、李廉水.政策供给与我国中小高科技企业的发展.科学学与科学技术管理.2004 年第 2 期:第 140 页
- [6] 蓝寿荣.国外科技型中小企业创新制度述要
- [7] www.most.gov.cn
- [8] 基于 C/S、B/S 混合模式的 MIS 安全性研究.蒙自高等师范专科学校学报. 2002 年 4 月.第 4 卷第 2 期
- [9] www.javaeye.com
- [10] www.csdn.net
- [11] 刘君强、顾海全、王讯等.基于开源框架的高质量 J2EE 应用架构[M]. 计算机工程与设计.2007.1(1): 16-18
- [12] 孙美莹.基于 Struts+Spring+Hibernate 架构的价格鉴定系统的设计与实现.东北师范大学硕士学位论文.2008.6: 10
- [13] 罗红梅. 集成 Hibernate、Spring 和 Struts 的架构应用研究. 电子科技大学硕士论文.2006.3
- [14] 一种优雅的流行架构: Struts+Spring+Hibernate <http://www.cn-java.com>
- [15] Ted Husted.Struts in Action 中文版.Eric Liu 译.机械工业出版社.2004 年 8 月
- [16] 肖杰浩.Hibernate+Struts 的 J2EE 应用开发.计算机与信息技术.2007 年 8 月
- [17] Craig Walls.Ryan Breidenbach.Spring in Action(第二版)中文版.毕庆红、王军等译.人民邮电出版社.2008 年 9 月
- [18] Craig Walls、Ryan Breidenbach.Spring in Action.Manning.2005:10
- [19] Hibernate2 参考文档
- [20] 萨师煊、王珊.数据库设计概论.第三版.高等教育出版社.2000 年 2 月

-
- [21] 王尖.基于 J2EE 架构的销售模式管理信息系统的设计与实现.北京邮电大学硕士论文.2008:49
- [22] Hibernate 缓存机制.<http://www.javaeye.com>.2008 年 10 月
- [23] 夏昕、曹晓钢、唐勇.深入浅出 Hibernate.电子工业出版社.2005 年 6 月
- [24] 陈天河.Struts, Hibernate, Spring 集成开发宝典.电子工业出版社.2006 年 11 月
- [25] 阎宏. Java 与模式.电子工业出版社.2002-10
- [26] James Goodwill. Mastering Jakarta Struts. Wiley Publishing, Inc., 2002 年
- [27] Cay S. Horstmann, Gary Cornell. Core Java™ 2: Volume II-Advanced Features.Prentice Hall PTR, 2001-12
- [28] 罗时飞.精通电子 Spring.电子工业出版社.2005 年
- [29] 孙卫琴.精通 Hibernate:Java 对象持久化技术详解.电子工业出版社.2005
- [30] R. Mateosian. J2EE developer's handbook. Micro.2003
- [31] 良葛格.Java 学习笔记.清华大学出版社.2006-8
- [32] Cay S.Horstmann、Cary Cornell.Java 核心技术.叶乃文等译.第二版.机械工业出版社.2008
- [33] Wendr Bosss、Mlcheal Bosss. UML with Rational Rose 从入门到精通. 电子工业出版社.2001-10
- [34] Apache Software Foundation.The Struts User's Guide V1.1.
<http://jakarta.apahche.org>. 2003
- [35] Mark Priestley.Practical object-oriented design with uml.龚晓庆、卞雷等译.第二版.清华大学出版社.2005-5
- [36] Martin Fowler.Patterns of Enterprise Application Architecture.
Addison-Wesley.2002
- [37] 刘晓华.UML 基础及 Visio 建模.电子工业出版社.2004 年
- [38] 陈刚.Eclipse 从入门到精通.清华大学出版社.2005 年
- [39] 吴际、金茂忠.UML 面向对象分析.航空航天出版社出版.2002 年
- [40] 段靖荒、林子禹、万丰.J2EE 企业解决方案的平台.计算机应用,2001.8(1)
- [41] 何成万.MVC 模型 2 及软件框架 Sturst 的研究.计算机工程,2002.28(6)
- [42] Walker.An Initial Assessment of Aspect-oriented Porgramming.Software Engineering,1995.5(16)
- [43] 吴建、郑潮、汪杰.UML 基础与 Rose 建模案例.人民邮电出版社.2004-10
-

-
- [44] Robertufore.Java 数据结构和算法.第二版.中国电力出版社.2003-2
 - [45] 华涛、郝克刚、葛玮.基于 HibeMate 和 Spring 框架的 Web 应用研究.计算机技术与发展,2006,16(11)
 - [46] 赵强.精通 JSP 编程.电子工业出版社.2006
 - [47] 吴雪阳、王洪强、陈英武.信息系统三层结构及其实现技术.电脑与信息技术,1999,(4)
 - [48] 张龙样.UML 与系统分析设计.人民邮电出版社.2001
 - [49] 康建萍.基于 MVC 模式的 Struts 框架研究与应用.计算机与信息技术.2006
 - [50] 蔡雪熹.Hibernate 开发及整合应用大全.清华大学出版社.2006 年
 - [51] 李刚.Spring2.0 宝典.电子工业出版社.2006 年 10 月
-

攻读硕士学位期间发表的论文及科研成果

- [1] 基于 Hibernate 的关联映射研究. 软件导刊. 2009. 6
- [2] 参与山东五征集团售后服务系统的开发与研究
- [3] 参与西藏自治区科技厅中小企业创新服务平台系统的开发与研究