

毕业设计（论文）说明书

学 院 计算机科学与技术

专 业 计算机科学与技术

年 级 2003 级

姓 名

指导教师

2007 年 6 月 15 日

毕业设计（论文）任务书

题目：无纸化试卷管理系统的研究与开发

学生姓名_____

学院名称 计算机科学与技术

专业 计算机科学与技术

学号_____

指导教师_____

职称 讲师

一、原始依据（包括设计或论文的工作基础、研究条件、应用环境、工作目的等。）

本次论文的工作基础是建立在无纸化试卷管理系统的研究与开发的基础之上的，通过对这个系统的开发与研究可以了解和学习到当前最先进的开发技术，同时将可以充分地认识到面向对象的开发理念，要求运用 JSP/Servlet/Bean 等相关技术，同时学会使用 Struts/Hibernate/Spring 框架，要求了解 CSS、JavaScript、Ajax，熟练使用 SQL 语句，来构建一套完整的无纸化试卷管理系统，本套系统可广泛的应用于各种需要进行试卷管理的场合，如学校、医院、政府等，开发这套系统能够解决人们对于试卷管理方面的需求，实现试卷管理的无纸化。

二、参考文献

- [1]孙卫琴，李洪成. Tomcat 与 Java Web 开发技术详解[M]. 北京：电子工业出版社，2004.
- [2]孙卫琴. Java 面向对象编程[M]. 北京：电子工业出版社，2006.
- [3]孙卫琴. 精通 Hibernate：Java 对象持久化技术详解[M]. 北京：电子工业出版社，2005.
- [4]孙卫琴. 精通 Struts：基于 MVC 的 Java Web 设计与开发[M]. 北京：电子工业出版社，2004.
- [5]Ryan Aseleson, Nathaniel T.Schutta, 金灵等译. Ajax 基础教程[M]. 北京：人民邮电出版社，2006.
- [6]Craig Walls, Ryan Breidenbach, 李磊，程立，周悦虹译. Spring in action 中文版[M]. 北京：人民邮电出版社，2006.
- [7] David Gallardo, Ed Burnette, Robert McGoven. Eclipse in Action[M]. Greenwich: Manning, 2002.
- [8] H.M.Deitel, P.J.Deitel 著. 施平安，施惠琼，柳赐佳译. Java How to Program (Fifth Edition) [M]. 北京：清华大学出版社，2004.

三、设计（研究）内容和要求（包括设计或研究内容、主要指标与技术参数，并根据课题性质对学生提出具体要求。）

本毕业设计希望基于 Java 构建一套完整的无纸化试卷管理系统，通过对管理系统的规划与设计来培养学生综合运用所学知识分析和解决实际问题的能力，强化计算机实际应用技能训练，为今后开展网站规划、设计和开发打下初步的基础。

1. 要求懂得 MVC 设计模式的概念，学会运用 Struts/Hibernate/Spring 框架和技术。
2. 建立严格的软件文档，包括需求分析，模块构建，概要设计，详细设计文档。
3. 系统要求具备的基本功能：能够自动随机生成多种题型的试卷页面，题型包括单选，多选，判断，填空以及简答。能够完成试卷的自动组卷，客观题和填空题自动评分，教师以及考生的登录和修改等任务。
4. 系统建立后要完成测试，最后将运行稳定的系统所有文件拷贝交付指导老师。

指导教师（签字）

年 月 日

审题小组组长（签字）

年 月 日

天津大学本科生毕业设计（论文）开题报告

课题名称	无纸化试卷管理系统的研究与开发		
学院名称	计算机科学与技术	专业名称	计算机科学与技术
学生姓名		指导教师	

一、课题的来源及意义

随着计算机的不断发展，它正在逐渐取代纸张，成为最常用的信息载体。网络更是以其惊人的速度发展，当今的网络可以说改变了很多人的生活，以网络为载体的大量的应用也应运而生。计算机的联网考试便是时下的一个热门研究领域，通过计算机联网考试可以大大提高考试的效率，缩短考试时间，节约考试费用，同时，网络考试还具有公正性，准确性，安全性，灵活性，方便性以及及时性等特点，已成为当前国际上普遍采用的一种考试和认证方式。

考试系统是在教育测量理论指导下对传统纸笔考试过程的数字化改造。而面向教师、试题的题库系统（包括试题管理、组卷、试题参数分析等功能）是实现无纸化试卷管理的基础组成部分，同时也是实现考试系统的无纸化、网络化的重要组成部分。

无纸化试卷管理系统的开发是现代化教育发展的要求，通过该系统可以大幅度减少纸张的浪费，提高教学与试卷管理的效率，同时，也能切实的减轻教师的工作负担。试卷的生成是本系统的一个关键之处，在该系统中试卷可以根据题库中的内容即时生成，从而可避免考试前的押题，而且可以采用大量标准化试题，从而使用计算机阅卷，大大提高阅卷效率；还可以直接把成绩送到数据库中，进行统计、排序等操作。目前的网络考试一般是将试题内容放在服务器上，考生通过姓名、准考证号码和口令进行登录，考试答案也存放在服务器中，这样考试的公平性、答案的安全性可以得到有效的保证，而且采用无纸化的方式使得考试的灵活性得到了很大的提高。因此，采用网络考试方式将是以后考试发展的趋势，而作为考试系统的重要和基础的组成部分，无纸化试卷管理系统是当前研究的热门领域。

试卷管理系统是一个庞大的系统，构建这个系统需要使用到很多技术，而该系统的试卷和试题管理、还有是这个系统的关键部分，对于它设计的好坏将直接关系到考试系统的整体的性能，系统所能给用户提供的功能，系统的稳定性和安全性的多方面。现在关于这种类型的系统设计与开发的技术很多也很成熟，这为该系统的研究奠定了良好的基础。

二、国内外发展状况

现在许多大型公司的认证考试，比如微软公司的 MCSE, MCDBA, MCS 等考试，SUN 公司的 JAVA 认证考试，Cisco 公司的 CCNA, CCIE 等认证，IBM 公司的 AIX、DB2、XML 认证考试等等，还有英语的托福、GRE 都已经采用了计算机联网考试。目前关于考试系统的研究也是一个很热门的方向，现在的考试系统的研究正往如何能开发出自适应、高智能、具有网络负载平衡能力和高安全性方向发展。试卷管理作为这个系统的基础和关键一环已经越来越显现出其重要性，因此开发一套适用范围广，受众面大的试卷管理系统是一项很有意义同时也是一项富于挑战性的工作。

三、本课题的研究目标

本课题要求在对国内外现有考试及试卷管理系统进行深入的调研的基础上，设计实现一个基于网络的、面向对象技术的 B/S 架构的无纸化试卷管理系统的研究与开发。

四、本课题的研究内容

本课题研究的中心内容是无纸化试卷管理系统的建立，研究此问题应结合现代考试的特点，只有对现代考试的特点、网络相关技术以及当前已经存在的考试系统有了深刻的认识，才有可能建立起适合现代教学要求的、满足考生需要的无纸化试卷管理系统。

研究内容包括两大部分：

1. 试题和试卷管理功能；
2. 教师和科目管理的功能。

在对试卷管理系统应具备的功能研究的基础上，提出适合现代考试特点的新型试卷管理方式，并对这种管理方式的特点、可行性进行探讨与实践，从而建立一套完善的无纸化试卷管理系统。

本系统由 2 个模块组成：管理模块，试题和试卷模块。

管理模块：主要负责管理考试科目名单和教师名单三者之间的关系。

试题和试卷模块：是本系统重要部分，它包括：管理所考科目的试题库，具备相应的添加，修改和删除等功能，按照教师特定需求的方式生成相应的试题，以及指定与考试相关的考生。

五、本课题的研究方法

本课题的研究方法选用当前比较流行的网络编程技术——Java, Struts, Spring, Tomcat, Hibernate 等技术来实现项目。作为一项当前流行的面向对象的网络编程技术，它们具有适时性好，保密性高等特点，这些可以充分保证试卷管理系统的稳定性，可靠性与安全性的原则。

除此之外，面向对象的技术具有随需应变的特点，能够适应当前需求变化快的需要，因此本次课题所构建好的试卷管理系统可以不需修改或只进行很小的改动就能满足各行各业的特殊需求，同时，基于 MVC 的设计模式也使得系统以后的升级很方便。

六、本课题的研究手段

本课题基于 JavaEE 等一系列相关技术来实现，通过相应的前期需求分析，用例分析，模块划分与设计，编码，测试，文档形成和交付等过程来完成整个项目。

七、实验方案的可行性分析

目前，本课题的研究已经积累了相当多的前人的经验，各种开源的工具和软件都可以从网上免费获得，因此项目的成本较低，同时，现在试卷管理系统已经是相当成熟的实现，这也为本课题的顺利实现创造了前提条件。

1. 社会因素方面的可行性：

本系统仅以用以考试和试卷管理为其目的，无法律和政策方面的限制，而且目前学校已经建立了一套完善的考试管理制度和措施，因此可以保证系统具有严格的可管理性。

2. 经济方面的可行性：

本系统作为一个毕业设计的一个项目，无需开发经费，而且，在系统实施以后将会显著提高考试的效率。因此，在经济上是可行的。

3. 技术方面的可行性

软件方面，网络化考试需要的各种软件环境都已具备，系统的软件开发平台已成熟可行。例如，该考试系统采用了当前流行的 Browser/Server 模式进行开发。数据库服务器选用 Mysql 数据库，它是一个免费的数据库，具有强大的数据处理能力，同时保持数据的完整性并提供许多高级管理功能，它的灵活性、安全性和易用性为数据库编程提供了良好的条件。硬件方面，科技飞速发展的今天，其硬件平台能满足此系统的需要。

八、已具备的实验条件

服务器使用与编码和测试相同的 PC，在该 PC 上架设了 Tomcat 服务器。客户机可以采用能上网的任何一台 PC，通过该 PC 可以测试和验证系统。软件开发的操作系统为 windows XP，开发平台为 Eclipse，数据库采用 Mysql，同时采用开源软件作为辅助。

九、进度安排

1. 2006 年 12 月 15 日——2006 年 12 月 30 日试卷管理系统前期交流
2. 2007 年 1 月 1 日——1 月 15 日试卷管理系统前期需求分析

3. 1月15日——2月15日试卷管理系统详细需求分析与规划
4. 2月15日——2月28日试卷管理系统概要设计，前后台分工
5. 3月1日——3月15日试卷管理系统模块分析与设计
6. 3月15日——3月30日试卷管理系统详细设计，接口划分
7. 4月1日——4月15日试卷管理系统编码，形成系统的详细文档
8. 4月15日——4月30日试卷管理系统编码与测试
9. 5月1日——5月15日试卷管理系统测试收尾阶段，形成系统的测试文档
10. 5月15日——5月30日试卷管理系统的交付调试阶段，同时开始相应的论文工作
11. 6月1日——6月15日完成论文以及整理材料，装订论文的工作。

十、主要参考文献

- [1]孙卫琴, 李洪成. Tomcat 与 Java Web 开发技术详解[M]. 北京: 电子工业出版社, 2004.
- [2]孙卫琴. Java 面向对象编程[M]. 北京: 电子工业出版社, 2006.
- [3]孙卫琴. 精通 Hibernate: Java 对象持久化技术详解[M]. 北京: 电子工业出版社, 2005.
- [4]孙卫琴. 精通 Struts: 基于 MVC 的 Java Web 设计与开发[M]. 北京: 电子工业出版社, 2004.
- [5]Ryan Aseleson, Nathaniel T.Schutta , 金灵等译. Ajax 基础教程[M]. 北京: 人民邮电出版社, 2006.
- [6]Craig Walls, Ryan Breidenbach, 李磊, 程立, 周悦虹译. Spring in action 中文版[M]. 北京: 人民邮电出版社, 2006.
- [7]David Gallardo, Ed Burnette, Robert McGoven. Eclipse in Action[M]. Greenwich: Manning, 2002.
- [8]Y.Daniel Liang. Introduction to Java Programming, Comprehensive Version, Fifth Edition[M]. U.S. : Prentice Hall. 2004.

选题是否合适: 是 否

课题能否实现: 能 不能

指导教师 (签字)

年 月 日

选题是否合适: 是 否

课题能否实现: 能 不能

审题小组组长 (签字)

年 月 日

摘 要

计算机及其相关技术的发展使其产生了很多与人们日常生活息息相关的產品，网络考试就是其中之一，其作为现代远程教育的一个重要组成部分，其因公正性，准确性，安全性，灵活性，方便性以及及时性等特点，已成为当前国际上普遍采用的一种考试和认证方式，而试卷管理系统的建设与开发关系着整个考试系统的性能与功能，本系统是一个基于 B/S 模式的管理系统，它的实现结合了网络技术和数据库技术，利用当前的校园网络资源，可以实现本校范围内各种可以进行试卷管理和测试，能够切实的减轻教师和考生的负担，提高教学效率。

本文首先分析了当前考试方式的特点和不足之处，同时，指出了采用无纸化试卷管理的优点，并因此提出试卷管理系统构成，设计思想和实现，本文的特别之处在于对一些组卷方面的算法深入的研究和探讨，最后分析了当前系统的不足和下一步的开发方向。

关键词：试卷管理系统；MVC 模式；试卷生成算法；数据库

ABSTRACT

With the development of computer and its relate technology, more and more products which relate to the people's life are produced, the network examination, which is an important part of distance education and for its advantage including equity, veracity, security, agility, celerity, has been widely used in the international examination. And the development of paper management system is related to the function and performance of the network examination system. This system is based on B/S mode and is an important application of computer in this age of Information Technology in that it can realize paperless management and the examination on all courses, using the resource of campus network.

This article analyzes the speciation and weaknesses of the recent way of examinations indicating that it will gradually be substituted by network examinations, thus present a design of a web-based paperless management system as well as its structure and implementation. The special of this paper is discussing many algorithms to realize generate test paper, At last the insufficient of the system and the future work directions are discussed.

Key words: paperless management system; MVC patterns; the algorithms to generate test paper; Database

目 录

第一章	绪论	1
1. 1	系统背景	1
1. 2	主要内容和研究意义	1
1. 3	论文的组织结构	3
第二章	无纸化试卷管理系统研究综述	4
2. 1	试卷管理系统研究现状与发展趋势	4
2. 2	试卷管理系统的作用	5
2. 3	试卷管理系统设计的相关原理	5
第三章	无纸化试卷管理系统的分析	10
3. 1	系统需要解决的主要问题	10
3. 2	系统需求功能描述	11
3. 3	系统运行环境	12
3. 4	系统数据库的设计	12
第四章	无纸化试卷管理系统的总体设计	19
4. 1	整个系统的总体设计	19
4. 2	试题管理与阅卷子系统	21
4. 3	科目与用户管理子系统	22
第五章	无纸化试卷管理系统的实现	23
5. 1	试题管理与阅卷子系统模块设计	23
5. 2	用户与科目管理子系统模块设计	25

5.3 对组卷算法的探讨与研究	27
5.4 与数据库的连接	30
第六章 系统的不足与展望	32
6.1 系统的不足	32
6.2 总结与未来的发展展望	32
参考文献	34
外文资料	
中文译文	
致 谢	

第一章 绪论

1.1 系统背景

自从人类进入 21 世纪以来，随着以计算机和互联网为代表的信息技术的高速发展和现代化教育改革的不断深入，使得远程教育和无纸化的考试将成为人们接受再教育或终身教育的主要形式，传统的考试方式必将面临着变革，因此，基于 Web 技术的网络考试将会是一个很重要的发展方向，网络考试因其公正性，安全性，准确性，灵活性，及时性以及方便性等特点，已成为当前国际上普遍采用的一种考试和认证方式。例如全球闻名的美国思尔文学习系统有限公司（Sylvan Learning System Inc.）的计算机化考试，包括计算机类的微软专家认证，外语的 GRE, TOFEL 等，还有 IBM, Cisco 等公司推出的各类认证考试，他们设计的这些考试形式可以说从一个侧面反映了未来考试的发展方向。在这些红火的考试背后无一不有着试卷管理系统的重要作用，只有通过这个系统对试题和生成的试卷进行良好的管理才能保证考试的顺利进行。

基于 Web 技术的网络考试系统可以借助于遍布全球的互联网进行，因此考试既可以在本地进行，也可以在异地进行，将不会再受到时间和空间的限制，这就大大提高了考试的灵活性，同时，试题是根据题库中的内容即时生成的，这样就可以避免考试前的押题，而且可以采用大量标准化试题，从而可以用计算机进行自动阅卷，这将大大提高阅卷效率，还可以直接把成绩送到数据库中，进行统计、排序等操作，现在的网络考试通常是把试题内容放在服务器上，考生通过用户名和考号进行登录，考试的标准答案存放在服务器的数据库中，这样考试的公平性、答案的安全性都可以得到有效的保证。

因此，采用 Web 网络考试方式将是今后考试发展的趋势，而作为考试系统的重要和基础的组成部分，无纸化试卷管理系统是当前研究的热门领域。

1.2 主要内容和研究意义

目前，学校与社会上的各种考试大都采用传统的考试方式，在这种方式下，组织一次考试至少要经过五个步骤，即人工出卷、考生考试、人工阅卷、成绩登记和试卷分析。显然，随着考试种类的不断增加和考试要求的不断提高，教师的工作量将会越来越大，其工作将会是一件十分繁琐和容易出错的事情，同时，一次考试所能得到的反馈较弱，组卷的随意性大，全凭教师的主观感觉来评价一套试卷的难易程度，对成绩的分析停留在平均分，及格率的表面层次上，难以统计出学生对于各个知识点的掌握程度，再者，成绩公布前考生为查看成绩而给教师

和教务部门带来的麻烦也不小，人情分的情况难免出现，因此，可以说传统的考试方式已经不能适应现代教育的需要。

当然利用计算机考试也不是一件容易的事，首先，人们习惯于传统笔和纸的考试方式，如果利用计算机考试就必须考虑到实施中很多具体的问题，其次，目前计算机考试还有一定的局限性，特别是在题型方面，虽然计算机可以很好的完成客观题的阅卷评分工作，但目前对于主观题还主要是靠人工阅卷，再次，利用计算机考试需要购置相应的终端设备，如架设网络，添置服务器和个人终端等，这将会是一笔不小的投资。

尽管当前计算机考试技术还有一些不足的地方，但是我们可以看到现在传统的考试方式已经显得捉襟见肘，现在我们可以将计算机考试运用于一些合适的场合，如客观题比较多的科目上，而且随着计算机信息技术和人工智能技术的进一步发展，上述提到的种种问题将逐步得到解决，计算机考试的应用范围也将越来越广，因此，计算机考试虽是一项新生事物，当我们已经可以看到它必将有着蓬勃的生命力，而作为无纸化考试系统的一个重要组成部分，我们很有必要研究一下试卷管理系统的开发与建设，使其能在各种考试中发挥作用。

目前，网络应用软件的模式主要有二类：Client/Server 模式（简称 C/S 模式），Browser/Server 模式（简称 B/S 模式）。前者需要在客户端上安装相应的软件，这样可以减少服务器与客户端的通讯负担，但同时在维护、升级方面也会变得很麻烦，后者是近几年伴随着互联网迅速发展起来的一种技术，它的出现大大减轻了客户端的负担，客户端是一个标准的浏览器，如 IE, Mozilla Firefox 等浏览器，服务器端是 Web Server，而 Web Server 与数据库和应用服务器的紧密结合，使得这种模式的应用范围不断扩大，它已不仅仅用于网上查询，有很多部门的业务系统、企业的 MIS 系统纷纷采用这种模式，其主要优点在于便于扩充应用，便于升级维护，降低了用户的总体成本。

利用目前的网络和数据库技术，结合互联网快速发展的有利优势，我们开发了基于 Java EE 的 B/S 模式的无纸化试卷管理系统。它使用方便、操作简单，效率很高。现阶段该系统已经实现了随机出题、客观题自动阅卷，试题管理、修改题库、用户管理、科目管理、管理员管理、成绩管理等重要功能，实现了真正的无纸化试卷管理，大大减轻了教师出题和阅卷等繁重的工作量。同时，由于采用了当前先进的开发技术和架构，这套考试系统能够很方便的增加新功能，为将来的升级打下了良好的基础。

整个考试系统采用基于 MVC (model-view-controller) 架构的 Java Web 开发方案，在服务器端我们采用 Mysql 数据库系统，采用 Tomcat 作为 Web 服务器，同时使用 Java, JSP, Struts, Hibernate 等技术和理论来实现考试的应用服务系统，客户端采用普通的浏览器来完成试卷管理的全过程，同时还可进行远程系统维

护、用户和科目的管理。

Web 服务器接受请求，通过应用程序服务器执行一个 JSP 程序，实现了客户机与服务器之间信息资源的交互。数据库服务器用来存储管理系统中所用到的各种数据，数据由数据库管理程序直接录入。系统的客户端只需要一个浏览器即可。相关人员通过浏览器来实现增加、删除和修改数据的操作，对信息进行管理。

本论文论述了一个功能较强大的试卷管理系统的实现过程，特别是在组卷算法方面进行了深入地探讨和研究，对不同算法的优缺点及适用范围提出了自己的见解，同时，在对数据库的设计方面也提出了自己的看法。目前通过本论文中的设想构建的系统基本上可以满足学校及现代信息化企业的内部考试要求，可以实现学校或企业的低成本投入，高效率产出的宗旨。如果选用了本系统，老师和领导将可以通过互联网轻松地组织试卷、安排考试，大大节省了时间，目前本系统包括六种考试题型：单选题、多选题、判断题、打字题、填空题、简答题等，其中前三种题型可以通过系统自动完成阅卷工作，这可以大大加快教师和领导的工作效率，减少负担，这六种题型已经基本覆盖了当前大多数考试的题型，因此能够很容易满足学校、企事业单位的考核要求，具有很强的实际意义和利用价值。

1.3 论文的组织结构

第一章：绪论。介绍了论文的选题背景、主要内容和研究意义。

第二章：试卷管理系统研究综述。简述该系统研究概况、存在的问题和国内外发展现状，以及与系统设计相关原理。

第三章：试卷管理系统分析。介绍本系统的需求分析、功能需求描述、目标系统要求、系统平台选择、数据库设计、总体设计图及说明。

第四章：试卷管理系统的总体设计。简述系统总体设计图及其子系统。

第五章：试卷管理系统的功能实现。重点介绍各个模块以及子系统的实现过程、特别是对组卷算法进行了研究与探讨。

第六章：该试卷管理系统的不足、改进设想和总结。提出目前本系统需要进一步完善的地方，以及对整个系统的研究、开发工作进行归纳和综合。

第二章 无纸化试卷管理系统研究综述

2.1 试卷管理系统研究现状与发展趋势

网上考试现在已经成为了国际上普遍采用的考试和认证方式，通过网络考试系统，考生可以就近完成考试，这可以大大减少时间和金钱的浪费，同时，网络考试现在也已经发展成为一门产业，通过它创造了很大的利润空间，解决了很多人的就业需要。在国内，随着各种网上课堂，远程教育的兴起，也出现了不少的网络考试形式，虽然这些考试的规模和影响并不是很大，但是他们从一个侧面也推动了中国现代化教育的发展。

目前，网络考试呈现出以下的特点和发展趋势：

方便学生考试，无时间和地域的限制，考生可以在一个考场里完成多门考试而不需要对考场进行大的调整。考生可以选择就近的考点和合适的时间来完成考试，这样有利于杜绝客观因素对考试的影响，有利于考生发挥出自己的真实水平。

能够随时更新考题，由于题目是保存在后台的数据库中，在终端上只是显示考题，因此教师可以通过终端向数据库中容易地添加、删除和修改试题，这样将可以很方便的将最新的试题反映到试卷中。

考试透明度高，现在许多认证考试都能在考试完成后当场就得到分数和考试的分析，这都是借助于计算机系统自动完成的，这样就可以减少评分过程中的主观因素，考试的公开度和透明度都很高。

安全和保密性高，由于题目是保存在数据库中，而访问数据库需要相应的密码才能进行，同时，在一个考场中的每个考生的题目各不相同的，同一考生即使多次考同一门考试，其题目也是不同的，因此这可以保证考试的公正性，防止作弊现象的发生。

考试正走向标准化和全球化，计算机考试的应用打破了传统的考试模式，通过计算机考试能够保证考试的严谨性，也使得授权认证能够得到广泛的认可。在考试界普遍认为计算机考试是二十一世纪的考试方式，目前，全球的计算机化考试正在迅猛的发展，现在我们在学校里就能通过互联网参加诸如 IBM、Cisco、Microsoft 的公司组织的计算机认证考试，这些认证为我们以后出去工作起到了添砖加瓦的作用。

作为考试系统的一个重要组成部分，试题的管理与试卷的生成将直接关系到考试能否顺利的进行。可以说，对于试卷的管理是考试的中心和灵魂环节。而无纸化的试卷管理系统又是实现网络考试的关键一环，它将直接决定网络考试的适应性、稳定性以及其所能具有的功能。

2.2 试卷管理系统的作用

利用试卷管理系统，教师只要将自己精心设计的试题和分值等信息存入数据库，将来在需要的时候计算机就能自动生成试卷，自动完成评分工作，这将大大减轻教师的工作负担并提高了其工作效率，与此同时考试的质量也得到了保证，从而使考试更趋于公正、客观，更能激发学生学习兴趣。

2.3 试卷管理系统设计的相关原理

2.3.1 开发工具的工作原理

本系统基于 Java EE 构建，采用 MVC 的 Struts 架构，在这个系统中用到了 Java, JavaBean, HTML, JSP, JavaScript, Struts, Hibernate 以及数据库的相关技术。

MVC 模式是"Model-View-Controller"的缩写，译为"模式-视图-控制器"。MVC 应用程序总是由这三个部分组成。事件(Event)的发生将会导致 Controller 改变 Model 或 View，或者同时改变两者。只要 Controller 改变了 Models 的数据或者属性，所有依赖的 View 都会自动更新。类似的，只要 Controller 改变了 View，View 会从潜在的 Model 中获取数据来刷新自己，参见图 2-1 和表 2-1。MVC 模式最早是 smalltalk 语言研究团提出的，应用于用户交互应用程序中。Smalltalk

表 2-1 MVC 的分工与协作

	模型 M	视图 V	控制器 C
分工	抽象系统应用的功能 封装系统的状态 提供使用系统功能的方法和路径	抽象数据表达 表示针对用户的数 据	抽象用户和系统的事件的语意映射 把用户的输入翻译为系统事件
	管理数据的存储和一致性 当数据发生变化时通知相关部分	维护与 Model 的数据一致性	根据用户的输入和上下文情况选择合适的显示数据
协作	当他改变系统数据时通知 View 能够被 View 检索数据 提供对 Controller 的操作路径	把 Model 表征给用户 当数据被相关 Model 改变时更新表示的数据 把用户的输入提交给 Controller	把用户的输入转成对 Model 的系统行为 根据用户的输入和 Model 的动作结果选择合适的 View

语言和 java 语言有很多相似性,都是面向对象语言,很自然的 Sun 微系统公司在 petstore(宠物店)事例应用程序中就推荐 MVC 模式作为开发 Web 应用的架构^[1]。

Jakarta-Struts 是 Apache 软件组织提供的一项开放的源代码项目,它为 Java Web 应用提供了模型-视图-控制器(Model-View-Controller ,MVC)框架,尤其适用于开发大型可扩展的 Web 应用^[2]。MVC 模式其实是一种架构模式,需要其他模式协作完成,Struts 实现了 MVC 的 View 和 Controller 两个部分,Model 部分需要开发者自己来实现,Struts 提供了抽象类 Action 使开发者能将 Model 应用于 Struts 框架中。

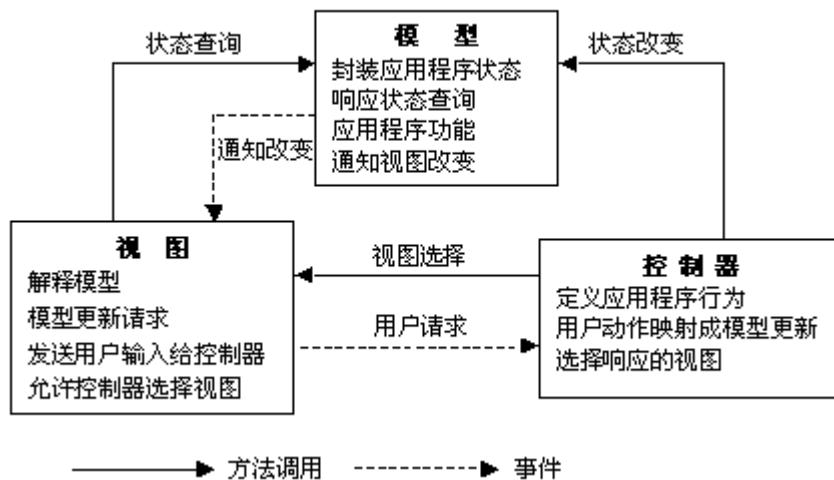


图 2-1 MVC 组件类型的关系和功能

Java 是由 Sun 微系统公司所发展出来的程序语言,它本身是一种面向对象 (Object-Oriented) 的程序语言。Java 是一种能跨平台使用的语言,这主要是因为 Java 本身被编译之后,并不是直接产生可执行的码,而是产生一种中间码叫做“字节码”,这种码必需在通过 Java 的虚拟机来解读它才能够真正的被执行,所以只要平台上装有这种 Java 的虚拟机,就能解读字节码也就能执行 Java 编译过的程序,因此,与 Java 程序是在哪种平台上被编译的,就完全没有关系了^[3]。Java 写出来的程序可分为两类,分别是 Java Applet 与一般的 Application。Application 类与一般的程序如 C++ 的作用是相似的,是一个独立可执行的应用程序。Applets 类似于应用程序,但是它们不能单独运行,Applets 可以在支持 Java 的浏览器中运行,Applets 主要是内置于 HTML 网页中,在浏览时发挥作用^[4]。

Java 语言的目标是为了满足在一个充满各式各样不同种机器,不同操作系统平台的网络环境中开发软件。利用 Java 程序语言,可以在网页中加入各式各样的动态效果。可以放上一段动画,加入声音,也可以建立交互式网页等^[5]。

JavaBean 是描述 Java 的软件组件模型,有些类似于 Microsoft 的 COM 组件

概念。在 Java 模型中，通过 JavaBean 可以无限扩充 Java 程序的功能，通过 JavaBean 的组合可以快速的生成新的应用程序。对于程序员来说，最好的一点就是 JavaBean 可以实现代码的重复利用，另外对于程序的易维护性等等也有很重大的意义^[6]。

HTML 是一种简单、通用的网络置标记语言。它允许网页制作人建立文本与图片相结合的复杂页面，这些页面可以被用户通过浏览器浏览到，无论他使用的是什么类型的电脑或操作系统^[7]。

JSP(Java Server Pages)是一种动态网页技术标准。JSP 技术是用 Java 语言作为脚本语言的，JSP 网页为整个服务器端的 JAVA 库单元提供了一个接口来服务于 HTTP 的应用程序^[8]。

Hibernate 是一个基于 Java 的开源的持久化构件，它对 JDBC 采用了轻量化的封装，通过它可以大量减少在 Java 中嵌入 SQL 语句的数量，同时，它还具有可扩展性，方便性等特点，越来越多的 Java 开发人员把 Hibernate 作为应用和数据库之间的中间件，并能以此减少和对象持久化有关的 30% 的 JDBC 的编程量^[9]。

Spring 是一种优秀的轻量级企业应用开发框架，它同时拥有自己的 Web 框架，能够提供诸如自动表单数据绑定和验证等功能，但是在本系统中，我们主要采用 Struts 框架，是基于如下的考虑：Struts 可以说已经是 MVC 框架的事实上的标准，有很多关于 Struts 的资源和应用可以很容易地得到，因此，Spring 在本系统中起到一个补充的作用^[10]。

Ajax 的主要组件是 JavaScript，Ajax 能够在当今的很多浏览器中使用，通过它使得互联网默认的请求/响应模式有了很大的转变，例如在 Netflix 公司（一家 DVD 租借公司）的网站上，当顾客把鼠标放到一个影片的图片上时，这个影片的 ID 号就会发送到服务器，然后在客户浏览器上会出现一个“气泡”，显示更多的关于这个影片的细节，而此时页面并没有被刷新，这些详细信息也并不是放在隐藏表单中，这样就可以为影片提供更多的信息而不会弄乱页面^[11]。

2.3.2 开发工具的工作过程及运行环境

前节所述的各项技术将通过 Eclipse 和 Tomcat 来实现，其中 JSP, JavaScript, HTML 等技术将主要在前台实现，即 MVC 架构的 View 层和 Control 层实现，而 JavaBean, Hibernate, 数据库技术将主要在后台实现，即 MVC 架构的 Model 层实现。

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。Eclipse 附带了一个标准的插件集，包括 Java 开发工具(Java Development Tools, JDT)^[12]。Eclipse

的主要部分包括一个平台运行时内核，在内核上面包括了工作台、工作空间、帮助和团队部件，其他的一些对基础框架的插件为应用提供了良好的支持^[13]，如图 2-2。

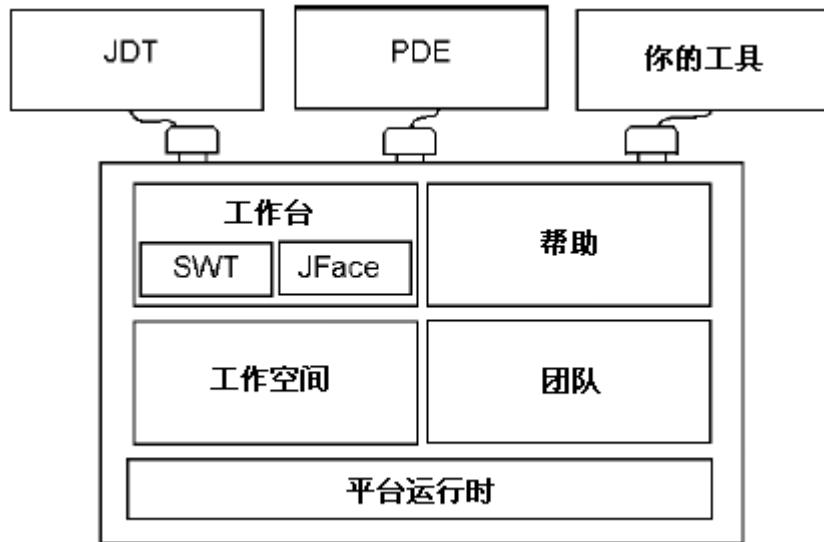


图 2-2 Eclipse 构成图

Jakarta Tomcat 服务器是在 Sun 微系统公司的 JSW JDK (JavaServer Web Development Kit) 的基础上发展起来的一个优秀的 Java Web 应用容器，它是 Apache-Jakarta 的一个子项目，它是一个开放源代码的软件，它可以和目前大部分主流的 HTTP 服务器 (IIS 和 Apache 服务器) 一起工作，而且运行稳定，可靠，效率高^[14]。

2.3.3 开发工具的编程特点

在本项目中我们基于如下的原因而采用 Java 语言：1、Java 语言是一种面向对象的语言，尤其适合 web 应用的开发；2、Java 具有跨平台的特性，因此，我们可以将我们编译的程序运用到别的平台上，这可以做到“一次编译，多次运行”；3、Java 直接支持分布式的网络应用，这也为我们以后项目的升级提供了良好的基础；4、Java 具有很强的安全性和健壮性，这使得我们写出的代码能够经受住时间的考验^[15]。

2.3.4 开发工具的环境需求设置

这套系统的应用需要先构建 Mysql 数据库，同时还要搭建 Tomcat 服务器，本套系统对硬件环境的要求不高，目前还处于服役期的电脑均可以很容易的运行

起来。

同时，由于我们采用的基本都是开放源代码的软件，因此，开发的成本和以后运行和维护成本都很低，这对于项目的顺利实施是一个很好的前提。

第三章 无纸化试卷管理系统的分析

3.1 系统需要解决的主要问题

首先，无纸化试卷管理系统的的主要用户是教师，但每一位教师所负责的科目不同，所以，在系统的开始界面上需要有一个登录页面，通过登录页面的身份验证可以跳转到该教师相应科目的试题管理页面，这个登录模块的设计也是保证系统安全性的一个重要的方面。另一方面，由于在线考试的考试环境一般为机房，考试者之间的距离很近，为了做到考试的公正性，对于每个应试者来说，试卷的试题类型和题量都应是相同的，但试题的内容却不能相同，而且在线考试基于网络环境，试卷应该从服务器的数据库随机抽取试题后动态生成，所以，这将是组卷模块需要解决的主要问题。另外，考生选择答案提交后，应该由计算机自动判卷，并将考生成绩和答案保存于数据库中以便于以后查验。此外，应该能够方便、快捷的对在线考试系统管理，如管理用户，管理科目等。

系统可行性分析：

- 1) 经济可行性：本系统作为一个毕业设计，并不需要任何的开发经费，而且，本系统实施后可以显著提高考试效率，所以本系统在经济上是可行的。
- 2) 技术上的可行性：软件方面，该考试系统采用了当前流行的 Browser/Server 模式进行开发。数据库服务器选用 Mysql 数据库，它是一个免费的数据库，能够处理大量数据，同时保持数据的完整性并提供许多高级管理功能^[16]。它的灵活性、安全性和易用性为数据库编程提供了良好的条件。因此，系统的软件开发平台已成熟可行。硬件方面，科技飞速发展的现在，硬件更新的速度越来越快，可靠性越来越高，价格越来越低，因此，硬件平台完全能满足此系统的需要。
- 3) 时机可行性：目前，互联网已经深入千家万户，本校作为中国教育网（China Cernet）的天津主节点，网络建设水平在天津地区的高校中首屈一指，学校良好的网络设施为开发和使用无纸化试卷管理系统的提供了坚实的基础。因此，在时机上也是可行的。
- 4) 管理上的可行性：由于目前学校已经建立了一套具体的，严格的管理制度和措施，因此，可以保证考试系统严格的管理性，规章制度和管理办法为系统的建设提供了制度保障。

综上所述，此系统开发目标已明确，在技术和经济等方面都可行，并且投入少、见效快。因此，系统的开发是完全可行的。

3.2 系统需求功能描述

登录:

- u 账号: 可以是英文字母, 数字和下划线
- u 密码: 六位以上的字母, 数字
- u 选择科目: 根据输入账号和选择科目综合考虑跳转到不同的界面 (教师或管理员)

考试管理界面:

- u 管理考试试卷库:

1) 添加试题:

包括试题类型 (选择、填空、判断, 打字, 简答等)、试题来源 (章节) 试题难度 (精度 1——5), 试题标准答案;

2) 删除试题:

一次可以删除一个或多个试题

3) 修改试题:

可以修改题目的来源

可以修改题目的难度, 提高或降低题目的难度

可以修改题目的标准答案

- u 生成试卷:

生成试卷规则:

根据科目, 试题类型, 题目来源, 试题数量等为每一个考生生成相应的试卷

- u 考试总结和评语:

考试总结:

1) 试卷试题分析: 统计每道题的出错率, 统计全体考生各题以及全卷的答题情况, 如: 总体的成绩, 平均分, 各分数段的人数等。

2) 成绩汇总: 成绩排名。

评语: 要求能针对每个考生写评语。

人工判断界面:

要求能够尽量在同一个页面中显示题目, 标准答案和考生答案

要求具有填写分数和评语的功能

系统用例图如图 3-1 所示:

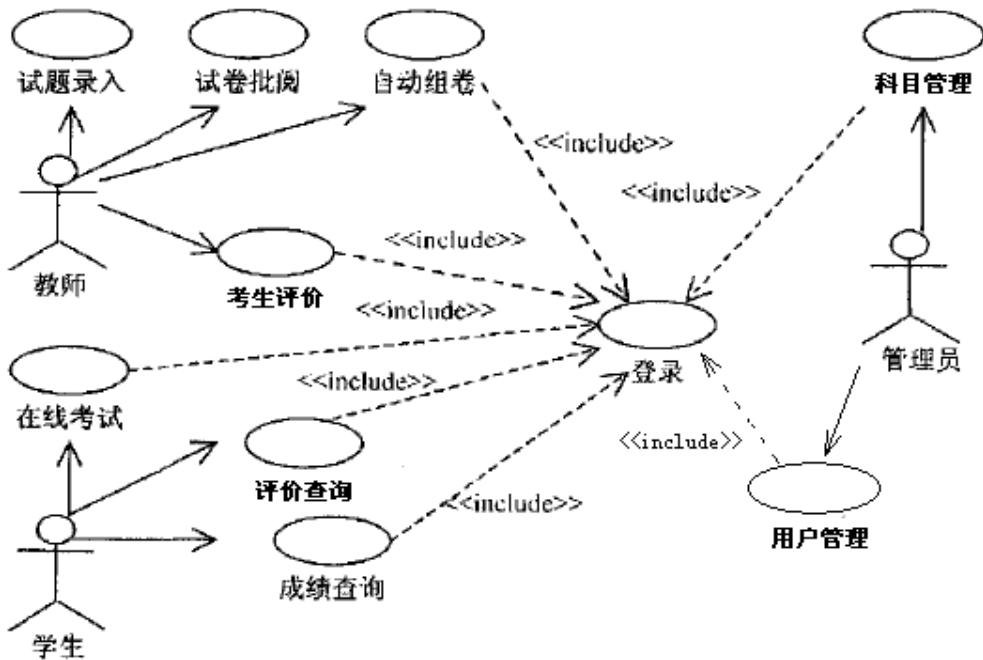


图 3-1 系统用例图

3.3 系统运行环境

系统运行于 Windows 平台上，支持 IE 等当前流行的多种浏览器，在服务器端需要搭建 Tomcat 服务器。

3.4 系统数据库的设计

3.4.1 数据库介绍

本系统的数据库采用的是 Mysql，一个最主要的原因是 Mysql 是一个免费的数据库，这将使得我们这套系统的部署成本得到很大的降低，但同时它却不是一个简单的数据库，对于我们的这个系统使用 Mysql 已经能够完全胜任，而且，Mysql 还具有功能强，使用简单，管理方便，运行速度快，可靠性高，安全保密等特点^[17]，这也是我们选择它的原因。

数据库结构的设计决定是本系统设计的关键一环，因为它将决定系统的性能和程序的结构，数据库结构的好坏将决定着本系统的成败。

3.4.2 数据库表分析

本系统的数据库根据不同的试题类型、不同的用户类型设计了多张数据库表，同时，围绕着考试的相关附属设计了考试成绩表，考生评价表等。

3.4.2.1 系统中用到的表

1. 有关成绩的数据库：

这两张表记录了考生的成绩，这里根据考生主客观题可能得分的数据类型的不同（客观题一般每题的分值为整数，主观题一般每题的得分可能会有小数），以及考生可能需要查询主观题的每题的分数的情况设计了两张表。

表 3-1 有关成绩的表

数据库中的表	说 明
ks_kegcj	考生客观题成绩表
ks_zhugcj	考生主观题成绩表

2. 有关评价的数据库：

这张表将要记录考生的评价信息，这个信息是每一个考生的每一门课程设定一条记录，而每门课程的教师将是这个记录的填写者，因此这个数据库将主要会在教师管理模块中用到。

表 3-2 有关评价的表

数据库中的表	说 明
pinj	考生评价表

3. 有关题库的数据库：

这个数据库将主要保存所有题目的信息，每门课每个教师的题目信息都将保存在这张表中，同时，考试数据库中考生每道考题的来源也来于此，可以说这个数据库是整个系统的关键之一。

该数据库的详细表单设计如表 3-3 所示。

4. 有关各类试题统计的数据库：

这个数据库主要将记录每个科目的每种题型的数量和分数，通过题数我们可以为每一道题生成一个唯一的序列号，通过分数我们可以实现计算机的自动阅卷，如表 3-4 所示。

表 3-3 有关题库的表

数据库中的表	说 明
tk_danx	题库单选题表
tk_daz	题库打字题表
tk_duox	题库多选题表
tk_jiand	题库简答题表
tk_pand	题库判断题表
tk_tiank	题库填空题表

表 3-4 有关各类试题统计的表

数据库中的表	说 明
tongj	各类型试题数统计表

5. 有关用户和科目管理的数据库

这个数据库记录的是用户（包括教师、学生和管理员）的信息，未来如有需要可以通过升级将该数据库的信息扩展，从而可以包括更多的用户信息，如表 3-5 所示。

表 3-5 有关用户和科目管理的表

数据库中的表	说 明
user	考生表
teacher	教师表
kem	科目表

3.4.2.2 数据库中各表的详细说明

1. 有关成绩的数据库：

1) ks_kegcj

考生的客观题成绩表，由于客观题成绩能由系统自动完成判分，公正性和公平性能够得到很好的控制，而且，客观题的分数一般都为整数，因此在 Grade 数据项我们设定的是 int 数据类型。在这个表中考生的考号（Uid）和考试科目（Kem）是主关键字。基于上面所述的这些原因我们设计了如下的数据库表，如表 3-6 所示。

表 3-6 考生客观题成绩表

记 录 名	含 义	数据类型
Uid	考生考号	int
Kem	考试科目	tinyint
<i>Grade</i>	客观题成绩	int

2) ks_zhugcj

考生的主观题成绩表保留了试题编号信息，这使得万一考生对成绩有异议时有据可查，同时，主观题成绩可能会保留到小数位，所以，这里我们采用的是 float 型作为主观题成绩的数据类型，如表 3-7 所示。

表 3-7 考生主观题成绩表

记 录 名	含 义	数据类型
Uid	考生考号	int
Kem	考试科目	tinyint
Tid	题库中试题编号	int
<i>Grade</i>	主观题成绩	float

2. 有关评价的表

1) pinj

评价表记录了教师对考生的评价信息，通过这个表考生可以与教师进行良好的互动，如表 3-8 所示。

表 3-8 评价表

记 录 名	含 义	数据类型
Uid	考生考号	int
Kem	考试科目	tinyint
Comment	考生评价	mediumtext

3. 有关题库的数据库：

有关题库的表存储了考试系统中各种试题的信息，这个数据库是试卷管理系统的中枢和基础，题库数据库表直接不是面向考生的，因此，在题库数据库中没有记录任何关于考生的信息。题库数据库中还记录了试题来源，试题难度等信息，这些信息为自动组卷的顺利实施提供了依据。以下列举了具有代表性的一部分数据库表：

1) tk_danx

题库单选题表记录了无纸化试卷管理系统中所有科目的单选题的信息，在该表中 Kem 和 Tid 是该表的主关键字，唯一标示了一条记录。Tid 的引入保证了每一个题目在数据库中都有一个独一无二的题号。标准答案 (Ans) 的存储是为了在进行自动阅卷时有据可依，如表 3-9 所示。

表 3-9 题库单选题表

记 录 名	含 义	数据类型
Kem	考试科目	tinyint
Tid	题库中试题编号	int
Source	试题来源 (章节)	smallint
Difficulty	试题难度	tinyint
Content	试题题目内容	mediumtext
OpA	选项 A 内容	mediumtext
OpB	选项 B 内容	mediumtext
OpC	选项 C 内容	mediumtext
OpD	选项 D 内容	mediumtext
Ans	标准答案	enum

2) tk_tiank

题库填空题表记录了无纸化试卷管理系统中所有科目的填空题的信息，在该表中 Kem 和 Tid 是表的主关键字，同时，Kem 还是 kem 表的外关键字。该表中的一个特殊之处就是需要教师设定每一道填空题的空格数，这是为了在生成考试的试卷时方便显示空格让考生填写，如表 3-10 所示。

表 3-10 题库填空题表

记 录 名	含 义	数据类型
Kem	考试科目	tinyint
Tid	题库中试题编号	int
Source	试题来源 (章节)	smallint
Difficulty	试题难度	tinyint
Content	试题题目内容	mediumtext
NumofBlank	空格数	smallint
Ans	标准答案	mediumtext

3) tk_duox

题库多选题表记录了无纸化试卷管理系统中所有科目的多选题的信息，其与单选题表不同的地方是由于多选题的答案可能有多个，因此标准答案的数据类型设置为“set”型，与题库单选题表一样，在该表中 Kem 和 Tid 是主关键字，如表 3-11 所示。

表 3-11 题库多选题表

记 录 名	含 义	数据类型
Kem	考试科目	tinyint
Tid	题库中试题编号	int
Source	试题来源（章节）	smallint
Difficulty	试题难度	tinyint
Content	试题题目内容	mediumtext
OpA	选项 A 内容	mediumtext
OpB	选项 B 内容	mediumtext
OpC	选项 C 内容	mediumtext
OpD	选项 D 内容	mediumtext
Ans	标准答案	set

4. 有关各类试题统计的表

1) tongj

统计表的设计有两个作用：1、统计目前该科目的该种类型的试题已保存的试题数，这样做的目的是为了为每一个试题生成一个独一无二的试题号；2、记录教师为这种题型所设定的分数，这样主要是为了在计算机阅卷的时候能够根据所设定的分数自动完成阅卷工作。统计表中 Tix 和 Kem 是主关键字，同时，Kem 是 kem 表的外关键字，如表 3-12 所示。

表 3-12 统计表

记录名	含义	数据类型
Tix	题型	tinyint
Kem	科目	tinyint
Tis	该题型已有试题数	int
Fens	该题型每题分数	tinyint

5. 有关用户管理的表

1) user

该表记录了考生的信息，考生登录需要在这里验证，考生的成绩也在这个表中记录，未来需要升级系统的时候可以扩展这个表，如表 3-13 所示。

表 3-13 考生表

记录名	含义	数据类型
Uid	考生考号	int
Pwd	密码	varchar
Kem	考试科目	tinyint
Score	考试成绩	float
State	考生状态 (No 未考 Yes 已考)	enum

2) teacher

将考生表与教师表分开存储一方面是为了保证考生数据和教师数据的安全性，另一方面，也是为了目前和将来系统的需要，也许在将来需要显示更多的教师信息，这样设计就可以不用对数据库表进行大的改动，只要添加相应的字段就可以了，如表 3-14 所示。

表 3-14 教师和管理员表

记录名	含义	数据类型
Id	教师编号	int
Pwd	密码	varchar
Kem	教师管理科目	tinyint

3) kem

科目表主要是将教师或考生输入的科目选择信息转换为相应的科目编号，这样做的目的主要是为编程提供方便，如表 3-15 所示。

表 3-15 科目表

记录名	含义	数据类型
Kem	科目	tinytext
Kid	科目编号	tinyint

第四章 无纸化试卷管理系统的总体设计

4.1 整个系统的总体设计

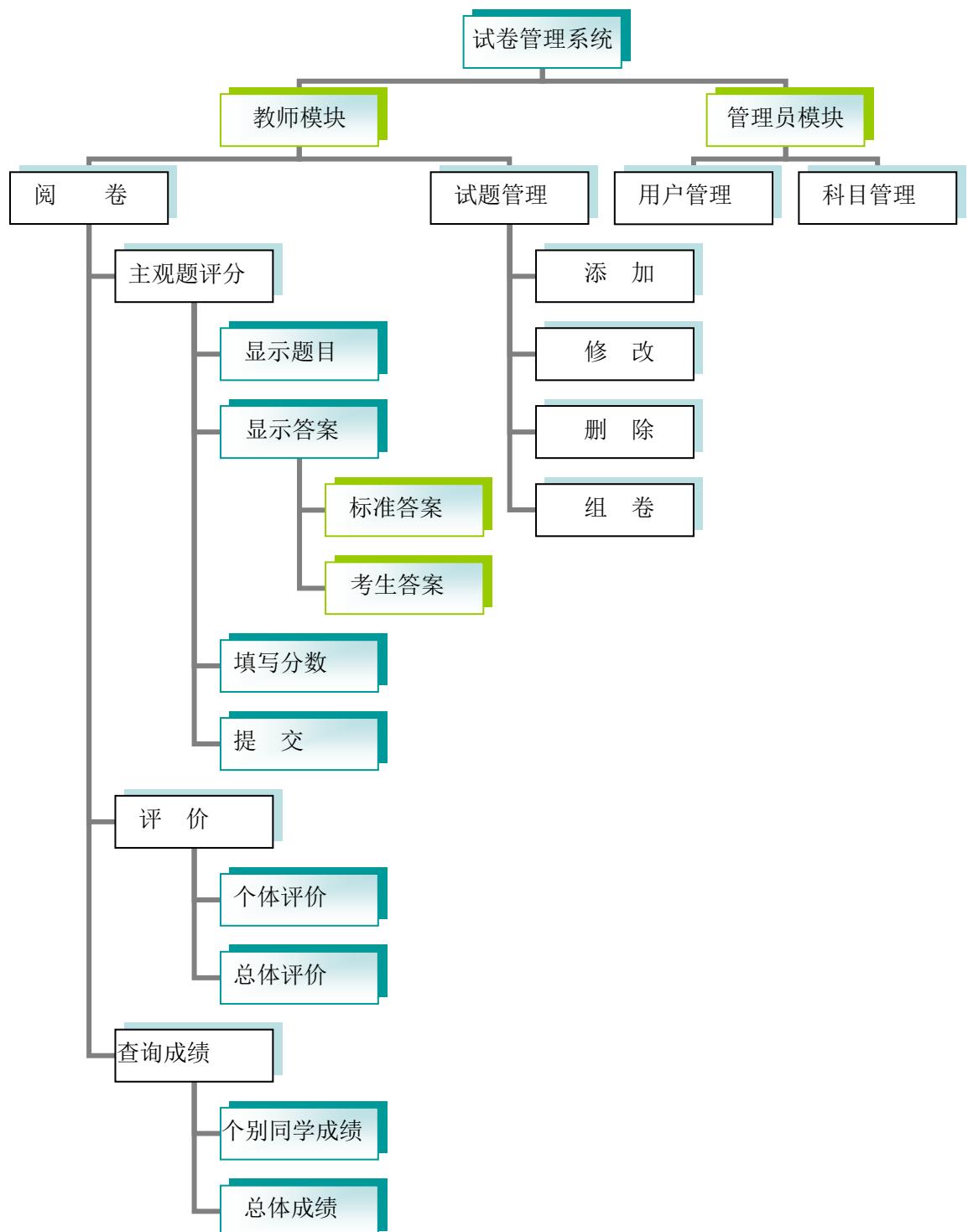


图 4-1 系统总体模块规划图

考试系统按照逻辑、功能和主要使用者的情况可以划分为以下两个主要的子系统：试题管理与阅卷子系统，科目与用户管理子系统。如图 4-2 所示。

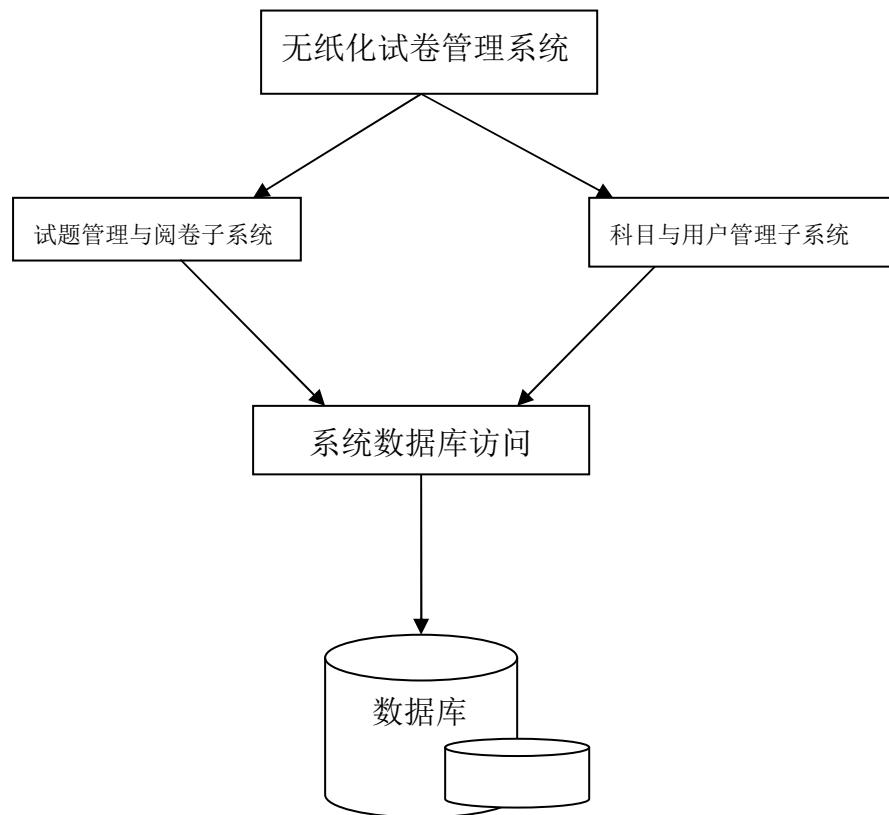


图 4-2 系统子系统划分图

4.2 试题管理与阅卷子系统

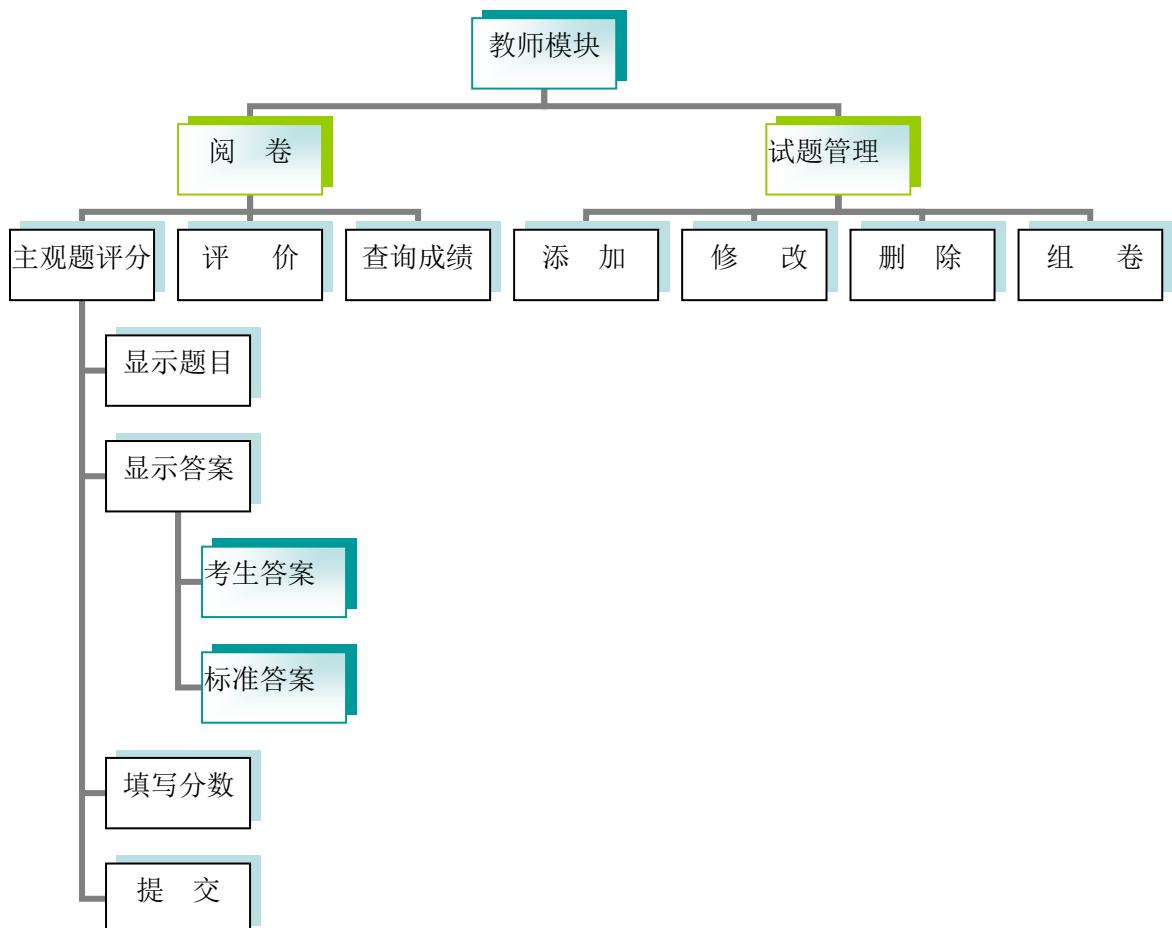


图 4-3 试题管理与阅卷子系统模块划分图

试卷管理与阅卷子系统的功能：

u 登录试卷管理系统：

试卷管理的首页，只有登录才能进入其内部。

登录界面包括：编号，科目，密码等，其中编号、密码为教师输入，科目以及选择登录者类型为通过下拉列表框选择。

u 管理试题，主要包括添加、修改、删除试题和组卷四部分：

1 添加试题：主要包括设置题目的内容、难度、题型、题目所属章节等四个部分。

1 修改试题：主要包括修改题型、难度等信息。

1 删除试题：可以进行一次一个或一次多个的删除试题。

1 组卷：设定试题总体难度，题目范围，各种题型比例，考试时间，总分，选择考生等内容。

- u 阅卷：主要针对主观性试题
 - 1 将会显示题目，标准答案，考生答案。
 - 1 同时有两个栏目：填写分数栏和填写评语栏。
 - 1 两个按键：提交和重置。

4.3 科目与用户管理子系统

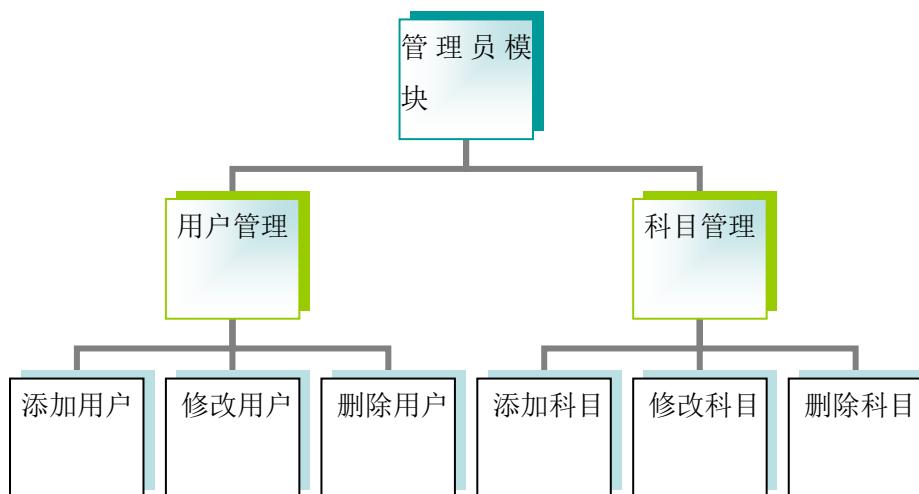


图 4-4 科目与用户管理子系统模块划分图

科目与用户管理模块的功能：

- u 登录试卷管理系统：
 - 科目与用户管理的首页，只有登录才能进入其内部。
 - 登录界面包括：编号、科目、密码等，其中编号、密码为教师输入，科目以及选择登录者类型为通过下拉列表框选择。
- u 管理员设置，包括管理考生，管理教师，管理科目三个部分：
 - 1 管理考生：页面将考生的基本信息罗列出来，并允许添加，删除和修改考生的信息。
 - 1 管理教师：系统将显示当前教师的基本信息，并允许添加，删除和修改教师的信息。
 - 1 管理科目：将显示当前科目的信息，并允许添加，删除和修改科目的信息。

第五章 无纸化试卷管理系统的实现

5.1 试题管理与阅卷子系统模块设计

该系统将主要由教师使用,设计该系统的目的在于使教师能够方便的管理其所教科目的试题,同时,由于试卷中不可避免地会有主观题存在,而以目前的计算机技术还不足以对主观题进行评分,因此,有必要将主观题阅卷模块设计出来以方便教师进行阅卷,此外,由于该系统的需求中需要有对考生的评价功能,因此评价考生的功能也将集成到该子系统中,如图 5-1 所示。

该子系统将主要包括以下功能:

1. 教师登录: 通过教师登录来进入相应的科目管理页面, 这是保证系统安全性的一个部分。在这个系统中我们将教师的编号的最高位定义为 3, 并以此作为登录验证的依据之一, 在教师通过登陆验证后会出现一个选择页面, 通过选择页面可以跳转到阅卷页面 (需要验证是否该科目有已经考过试但还没有被批阅的考生存在) 或者试题管理页面。

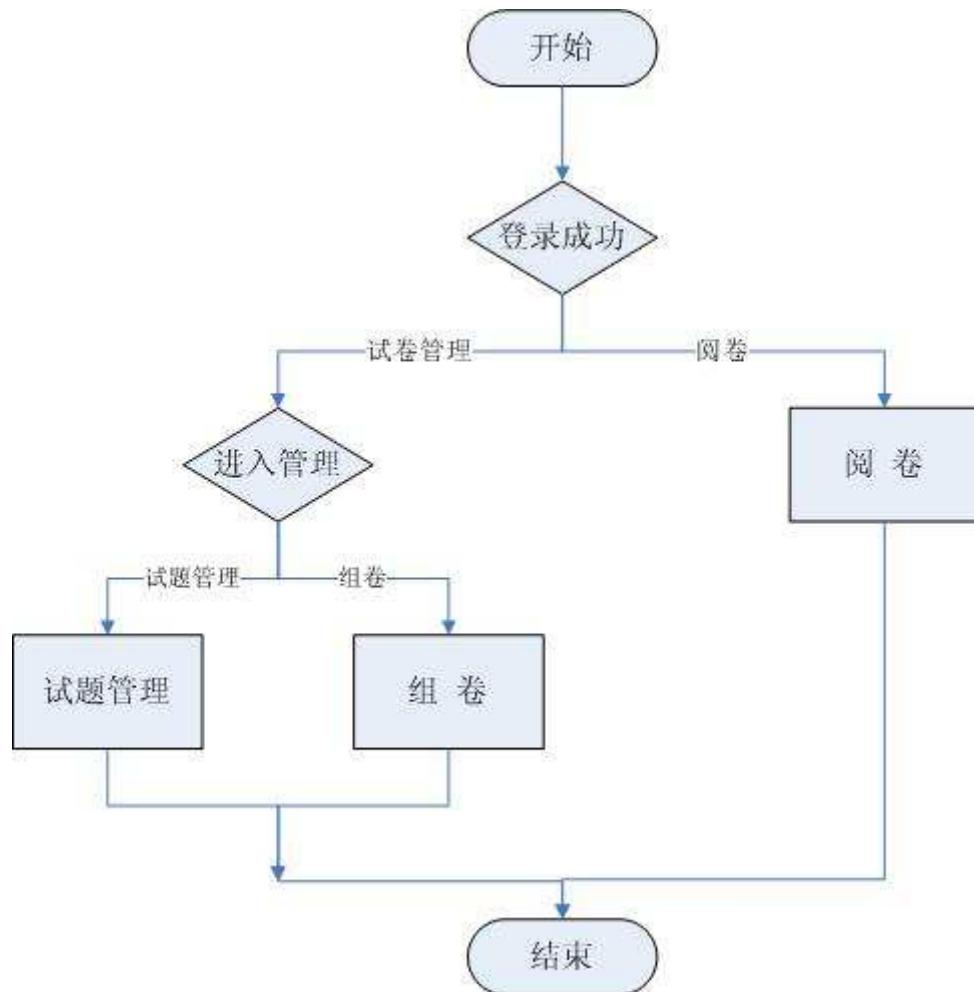


图 5-1 试卷管理与阅卷子系统流程图

2. 试题管理，该功能还将包括以下子功能，如图 5-2 所示：
- 1) 试题添加：教师可以根据科目的需要来添加相应的试题，在添加试题时可以设定相应的题目来源和难度等信息。
 - 2) 试题修改：教师可以根据自身的需要来修改试题的内容，难度，来源，答案等信息。
 - 3) 试题删除：若教师觉得该试题已不再需要时可以删除。
 - 4) 组卷功能：这个功能也是该考试系统的核心功能之一，通过该功能教师可以为每一个考生生成一份“个性化”的试卷，这样，将有利于保证考试的公正性。教师通过设定试卷中题目的来源，试卷所需的试题类型，每种类型试题所需要的数量等信息就可以由计算机自动生成一套试卷，通过计算机可以保证每名考生试卷的难度一样，这是手工组卷所做不到的，同时，通过计算机自动组卷也将大大降低教师的工作负担，提高教师的工作效率。在设计组卷程序时需要考虑这些问题：每一位考生的试卷必须题量一致，难度相当，还要考虑到考生可能会有多次考同一门考试，因此还要保证考生每次考试的试题不能一样。

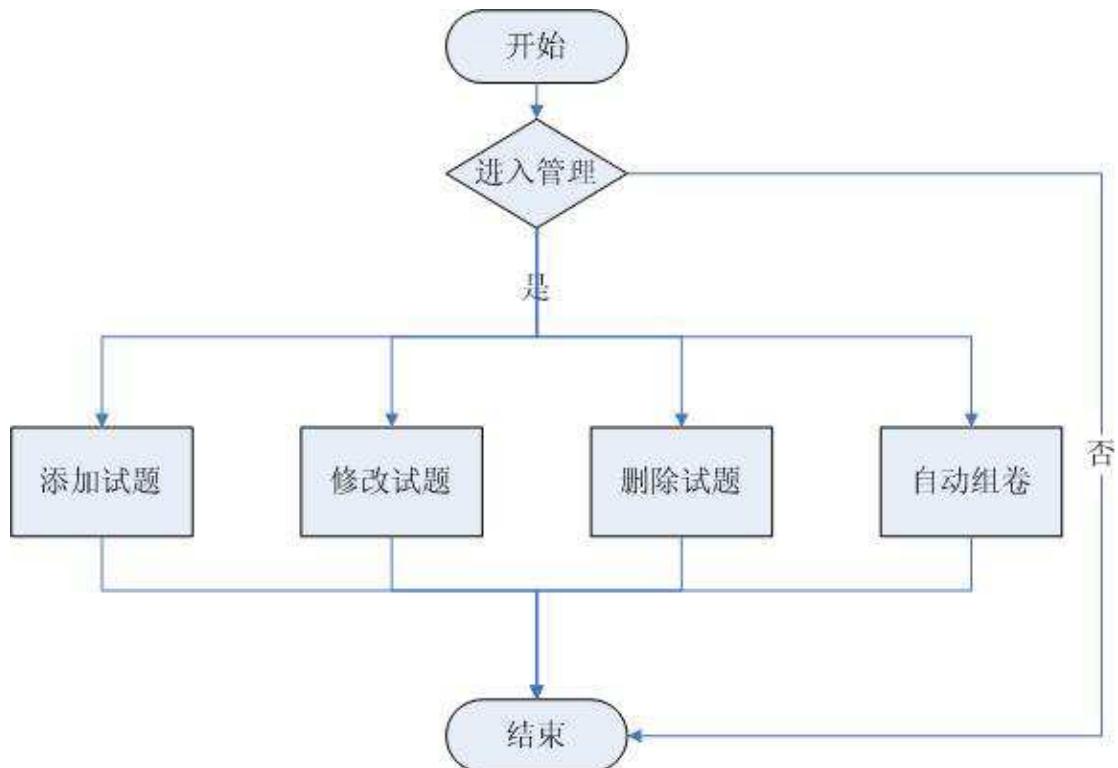


图 5-2 管理试题系统流程图

3. 阅卷与考生评价，如图 5-3 所示：

- 1) 阅卷: 阅卷功能将包括相应主观题目的显示、考生答案的显示、标准答案的显示和教师填写分数等子功能, 其中, 为保证分数的准确性, 教师的判断可以精确到小数的数量级。
- 2) 考生评价: 教师在批阅完每一个考生的所有主观题后可以对考生有一个总体上的评价, 而这个评价将被考生在查阅成绩时看到, 这样就有利于教师和学生的互动, 有利于学生发现自己的不足。



图 5-3 阅卷系统流程图

5.2 用户与科目管理子系统模块设计

该子系统模块将主要被管理员使用, 我们认为在该系统中管理员也是一名老师, 因此管理员的信息也将保存到数据库的 `teacher` 表中。科目管理和用户管理是该系统的一个基础功能。

1. 科目管理, 如图 5-4 所示:

包括如下功能:

- 1) 查询科目: 将会返回目前已存在数据库中的科目信息。

- 2) 添加科目：管理员只要添加科目，系统会自动为该科目生成相应的题型库。
- 3) 修改科目：可以修改科目的名字。
- 4) 删除科目：当该科目不再需要时可以删除，同时与该科目相关的题目等信息也会相应的自动删除。

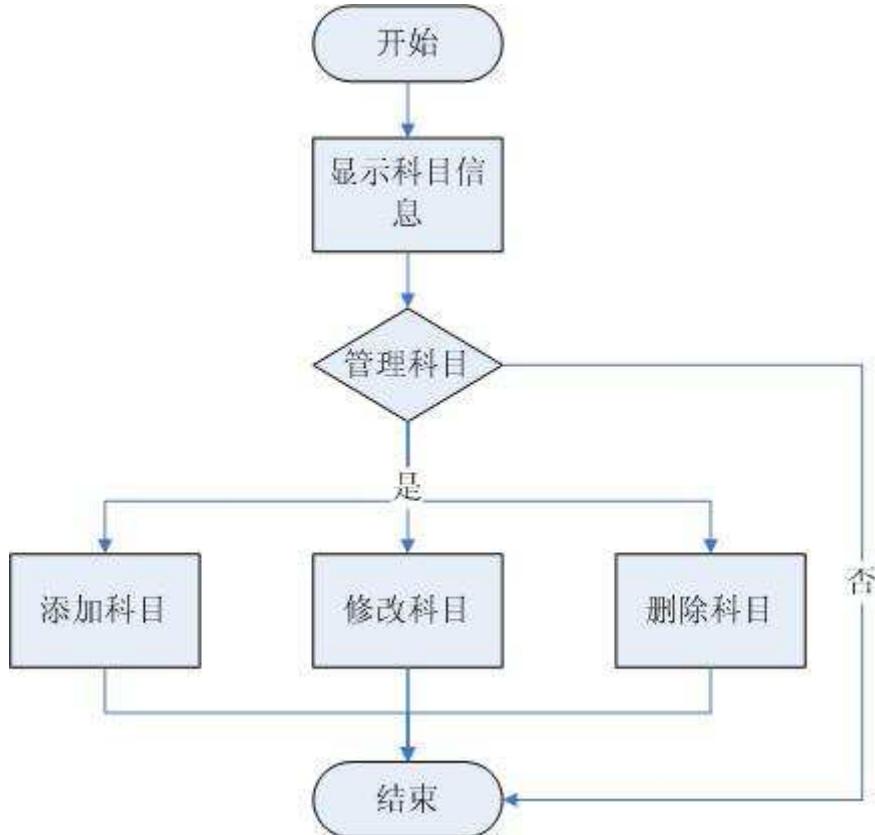


图 5-4 管理科目系统流程图

2. 学生管理：

包括如下功能：

- 1) 查询考生：在本系统中主要用于验证考生的合法性，同时，还用于管理员或教师查询考生的信息。
- 2) 添加考生：用于增加考生。
- 3) 删除考生：用于去除无效考生，如该考生已经毕业离校等之类的。

3. 教师管理：

包括如下功能：

- 1) 查询教师：在本系统中主要用于验证教师的合法性。
- 2) 添加教师：添加可以管理科目的教师。
- 3) 修改教师：本阶段限于修改教师的密码，但已经为未来的升级预留了接口。

- 4) 删 除 教 师: 取 消 不 再 管 理 相 关 科 目 的 教 师 的 信 息, 这 也 是 系 统 安 全 性 的 一 部 分。

5.3 对组卷算法的探讨与研究

5.3.1 组卷分析

本系统的一个关键的地方就是要建立一个良好的组卷系统, 采用自动组卷系统可以大大减轻教师的负担, 而且, 更重要的是, 自动组卷可以很好的保证试卷的难度和信度, 提高考试的公平性和公正性。

如何能保证生成的试卷能够最大程度的满足使用者的不同需要, 并具有随机性、合理性和科学性等是自动组卷在实现方面的一个难点。而且, 在交互式的 Web 环境下人们对试卷生成的速度的要求很高, 而一个理论上较完美的算法可能会以牺牲时间作为代价^[18]。

5.3.2 组卷算法介绍

目前在组卷方面有以下几种算法:

1. 随机抽题法

随机抽题算法^[19]根据状态空间的控制指标, 由系统随机的自动的向题库抽出一道试题加入到试卷中, 这个过程不断重复, 直到试卷生成完毕, 或者无法再从试题库中抽取出满足状态空间控制指标的试题为止。

算法的具体实现过程如下:

- 1) 建立两个数组 $U(r)$ 和 $T(x)$, 其中 $U(r)$ 的值为某种状态 r 的试题在数据库中的试题数, 例如单选题, 第二章, 难度为 3 的试题数。 $T(x)$ 为用户要求该状态的试题数目, $T(x)$ 构成的全部状态的集合构成一个线性表 List。
- 2) 如果 $T(x) > U(r)$, 则转向 5), 否则产生随机数 N , $N = \text{int}(\text{rand}(-1) * U(r))$, 其值小于 $U(r)$ 。读取 N 记录, 比对该记录作选取标志, 抽取下一道题目时, 有选取标志的记录将不再被选取。
- 3) 若 $T(x) \neq 0$, 则 $T(x) = T(x) - 1$, 重复 2);
- 4) 若 List 表未满, 则转向 1), 否则组卷成功;
- 5) 算法结束

这种算法使用广泛、简单易行, 对于单道题的抽取速度很快, 但是对于整份试卷的生成却需要耗费很多时间, 而且在组卷条件较复杂的情

况下常常会出现组卷失败。此算法适用于小型题库系统和组卷约束条件较少的场合。

2. 回溯试探算法

回溯试探算法^[19]是对随机算法的改进，它将随机抽取产生的每一状态类型记录下来，当搜索失败时就会释放上次记录的状态类型，然后再依据一定的规律变换出一种新的状态进行试探，通过不断地回溯试探直到试题生成完毕或回到出发点为止。

该算法的具体实现步骤是：

- 1) 建立 2 个数组 $U(r)$ 、 $T(x)$ 及线性表 List，含义同随机抽题算法
- 2) 如果 $T(x) > U(r)$ ，则先按一定的策略减少状态 r 的 $T(x)$ 的值；然后增加与之相接近的另一状态 $U(x)$ 的值。否则产生随机整数 N ，
 $N = \text{int}(\text{rand}(-1)*U(r))$ ，其值小于 $U(r)$ 。读取 N 记录，并对该记录作选取标志，抽取下一道题目时，有选取标志的记录不再被选取；
- 3) 若 $T(x) \neq 0$ ，则 $T(x) = T(x) - 1$ ，重复 2)；
- 4) 若 List 表未满，则转向 1)，否则组卷成功；
- 5) 算法结束

回溯算法在理论上可以遍历每一种状态组成，但是当试题库中试题很多时这种状态的组成就会很复杂，而且该方法在空间复杂度和程序设计方面都很复杂，所选的试题没有随机性，组卷效率低。

3. 基于专家系统的组卷算法

专家系统是一种能够依靠大量的专门知识解决特定领域中复杂问题的计算机智能软件系统。专家系统的特点之一就是能够进行符号操作，用符号来表示知识，它把问题概念表示成符号集合。在组卷系统中的专家系统需要构建一个知识库用来存放诸如不同的科目、不同的考试性质和时间等组卷知识信息，此外，还要构建一个推理机制来根据一定的推理策略从知识库中选取相关的知识，对用户提供的信息进行推理，直到得出相应的结论为止。组卷时，用户输入组卷参数，使用推理机制进行推理匹配，从组卷知识库中获得试卷模型，并按产生的随机数在同一类试题中选择试题，最后输出组卷结果^[20]。

这种组卷算法需要建立一个庞大的知识库，同时推理机制的设计和实现也很复杂，难以满足快速组卷的要求。

4. 启发式搜索法

随机化启发式搜索法^[19]是建立在人工智能和一定的概率模型基础上的一种有效算法。在搜索的前几步中采用随机抽题法，当搜索进入死结点时，采用以下的方法作启发后再进行搜索：首先将造成死结点的状

态类型记录下来，然后回溯走过的路径，将与该状态有关(指某分量相同)的元素全部释放，将剩下的无关元素重新构成一条路径，然后根据启发函数的最小值确定下一个结点，启发函数定义为新元素状态类型与记录死结点元素状态类型分量相同的个数。

启发式搜索法的组卷成功率高，能够满足较复杂的组卷要求，但是其程序设计复杂，组卷速度慢。

5. 遗传算法^[21-26]

遗传算法（Genetic Algorithm）是目前在组卷系统设计领域受到广泛关注和研究的算法，它最早是由美国密西根大学的 Holland 教授在六十年代提出，在进行了一系列研究后，八十年代由 Goldberg 进行归纳总结，形成了遗传算法的基本框架，其具有简单通用、收敛速度快、全局寻优等特点，且适用于并行处理，目前在计算机科学、神经网络、信号处理、人工网络等领域已经得到了广泛的运用。

它是一种模拟自然界的遗传和进化而形成的一种自适应的启发式的蒙特卡罗反演算法，它能有指导性的进行随机化的搜索，适用于解决复杂目标的非线性问题。遗传算法处理的是基因型个体，一定数量的个体组成群体，群体中个体的数目为群体规模^[24]。遗传算法中通常具有一个目标函数与生存环境相对应，由这个目标函数来确定各个个体的适应度。在遗传算法中需要进行数据转换操作，一个是表现型到基因型的转换，把问题空间的参数转换成遗传空间中的基因型个体；另一个是基因型到表现型的转换，它将执行相反的操作。Goldberg 总结了一套基本遗传算法（Simple Genetic Algorithm），这套算法只使用选择算子、交叉算子和变异算子等三种基本遗传算子，遗传过程简单，这个算法也为其他的遗传算法提供了一个基本的框架。

遗传算法的基本过程如下：

- 1) 产生初始群体，随机地（通常均匀地）产生若干个个体，每个个体看作是一个基因型个体，这若干个基因型个体组成一个群体；
- 2) 对群体中每个个体计算它的适应度；
- 3) 通过选择和复制操作从群体中选出所需要的个体，这里通常采用轮盘赌选择法即适应度比例法，而且通常需要多轮选择，然后将选出的个体放入到交配缓冲池中；
- 4) 对交配池中的个体使用交叉和变异算子仿照生物学中杂交的原理，形成下一代群体中的 N 个体，并计算每个新个体的适应度，在这里有必要提一下交叉算子和变异算子的重要性：交叉算子一方面能够使得原来群体中优良个体的特性在一定程度上得到保持，另一方面，它将使

得算法能够探索新的基因空间，从而使新群体中的个体得以保持多样性^[25]；变异算子的使用能够使遗传算法具有局部搜索能力，可以维持群体的多样性^[26]；

5) 如果满足结束条件，则停止，否则转到第 3) 步。

但是遗传算法也有一些不足的地方，如其采用二进制编码，代码冗长，而且二进制编码不能有效的拓展搜索空间，其次，基本遗传算法在搜索后期会出现效率低和形成未成熟收敛的情况。

5.3.3 本系统采用的算法

本系统目前是一个在学校范围内使用的小型系统，在对组卷模块的设计与实现时考虑了以下的因素：

1. 时间因素，由于系统开发周期较短，一些理论上效果较好的算法由于代码复杂而未被考虑。
2. 环境因素，该系统目前的使用范围限于学校，因此题库不会很大，所以没有必要采用很复杂的算法来实现。

因此，目前在本系统中采用的是随机抽题算法，从当前的使用情况来看，算法的选择是正确的，是能够满足需要的，下面是该算法的部分程序实现：

```
public class ZUJBean{
    public static int zujuan(int uid ,String kem,int tix,int source,int tis){
        //自动组卷，输入：考生ID，科目，题型，来源，题数
        try{
            int id = 1;
            int count;    [
            switch(tix){
                case 0://单选题
                    while(true){
                        int r = (int)(Math.random()*100);
                        KEM kemmodel = KEMBean.queryKID(kem);
                        int difficulty = (int)(Math.random()* 6);
                        int tid = kemmodel.getKid()*10000000 + tix * 1000000 + difficulty * 100000 + source * 1000 + r;
                        TK_DANX danx = TK_DANXBean.queryTK_DANX(tid);
                        if(danx != null)
                        {
                            count = KS_DANXBean.addKS_DANX(uid, tid, id);
                            id++;
                            if(id == (tis + 1))
                                break;
                        }
                    }
                    break;
            }
        }
    }
}
```

5.4 与数据库的连接

在本系统中，访问数据库是一个很重要的环节。由于我采用基于 Java EE 的

技术来构建无纸化试卷管理系统，因此，在数据库连接方面采用的是 JDBC（Java Database Connectivity）接口来访问数据库的。

JDBC 访问数据库需要首先建立一个与数据库的连接，获得一个连接对象，通过连接对象来提供执行 SQL 语句的方法。

在此系统中，我们采用的是 Mysql 数据库，因此这里的设置上需要对其进行专门的设置。

以下是连接数据库代码：

```
public class DB {
    private static String driverName = "org.gjt.mm.mysql.Driver";
    private static String dbUrl = "jdbc:mysql://";
    private static String serverName = "localhost/";
    private static String databaseName = "onlineexam";
    private static String userName = "root";
    private static String password = "123";

    public DB() {
    }

    public void setValue(String servername, String DataBaseName, String username, String pswd)
    {
        serverName = servername;
        databaseName = DataBaseName;
        userName = username;
        password = pswd;
    }

    public static Connection connectToDb() throws Exception {
        Connection connection = null;
        //使用MySQL驱动
        String connName = dbUrl + serverName + databaseName ;
        Class.forName(driverName).newInstance();
        connection = DriverManager.getConnection(connName + "?user=" + userName + "&password=" + password +
            "&useUnicode=true&characEncoding=GB2312");
        return connection;
    }
}
```

第六章 系统的不足与展望

6.1 系统的不足

经过一段时间的使用和测试，该系统已经基本实现了当初的设想，达到了预期的目的。但是，这个系统还是有一些可以提高的地方：

- u 系统中的打字题当初设想是要实现电脑自动评判正确率，可是由于目前自身技术力量和该系统是基于 web 框架的限制，这一块还没有完全实现成功。
- u 由于当前设计的是试卷管理系统，所以在考生和教师信息的保存方面所能保存的内容还较少，未来如果需要将该系统升级成为一套完整的教学管理系统的话在这方面还需要加强，所幸的是，由于采用的是 MVC 的框架结构而不是仅仅简单地采用纯 JSP 编程的方式，所以未来升级的时候将会大大减轻劳动量，可以说，现在已经为将来的升级打下了良好的基础。
- u 目前系统的评价体系还不够完善，由于时间和该系统目前需求的原因，该系统的评价体系主要还是依靠每个考生的成绩和阅卷教师给出的相应评语来评价，未来可以进一步引进一些统计学方面的理论来对考生和考试进行更准确细致的评价。同样地，目前对于试题的评价也仅限于难度的区分，但以后可以根据考生的情况进一步引进信度，区分度等评价试题的标准。
- u 系统的容错能力还需要进一步提高，由于目前知识水平的限制，对于一些技术的应用和理解能力还需要进一步提高，因此，在程序的容错，极端情况下的稳定性方面还有待进一步测试与改进。

6.2 总结与未来的发展展望

本论文详细阐述了基于 MVC 架构的试卷管理系统的设计与实现过程。该系统使用了目前先进并被广泛应用的 Struts 架构，这为以后系统的扩展和维护打下了良好的基础，例如，系统可以很方便的改变试题管理界面而不用修改任何的业务逻辑，可以对业务规则和数据的物理表示进行改进而不用修改任何的教师界面代码。

本系统是一个平台，能够实现多学科、多课程、多层次的试题库管理，满足同一个考场中每一位考生试卷既等价又相异的要求，有利于大规模同一考试的顺利进行，目前，该系统运行平稳可靠。

由于时间的关系，还有一些技术并没有被用到这个系统中，这是系统下一步需要改进的地方，以后还需要考虑采用更先进有效的算法（如：遗传算法、线性规划算法等）对组卷模块进行修改，提高组卷的质量和速度，同时，还需要在考试系统的安全性，提高考试系统的自适应能力，实现多媒体考试等方面做进一步的研究。

基于 Web 的考试系统是当今的一项热门研究领域，它是现代远程教育不可缺少的一部分，而作为考试系统的核心部件——试卷管理系统也越来越受到人们的关注，大量的学者、专家投身到这个领域中来。网络考试正以其不可比拟的优势融入到当今社会中来，相信在不久的将来随着各种技术的不断完善，网络考试将会向着主观题自动阅卷，根据考生的水平自动制定相应的考试方向等自适应、智能化的方向发展，而试卷管理系统设计得好坏将直接关系到网络考试系统在社会上被接受的程度，相信在不远的将来更加先进、更具人性化的试卷管理系统将越来越多的融入到人们的生活中。

参考文献

- [1]中科永联高级技术培训中心. MVC 模式
[EB/OL]. <http://www.itisedu.com/phrase/200604231324325.html>, 2006-04-23.
- [2]孙卫琴. 精通 Struts: 基于 MVC 的 Java Web 设计与开发[M]. 北京: 电子工业出版社, 2004. III—IV.
- [3]孙卫琴. Java 面向对象编程[M]. 北京: 电子工业出版社, 2006. VI—VII.
- [4]H. M. Deitel, P. J. Deitel 著. 施平安, 施惠琼, 柳赐佳译. Java How to Program(Fifth Edition) [M]. 北京: 清华大学出版社, 2004.
- [5]王萌, 刘婧, 来宾. JAVA 程序设计[M]. 北京: 冶金工业出版社, 2004. 1—3.
- [6]孙卫琴, 李洪成. Tomcat 与 Java Web 开发技术详解[M]. 北京: 电子工业出版社, 2004. 123—124.
- [7]张孝祥, 张红梅. JavaScript 网页开发——体验式学习教程[M]. 北京: 清华大学出版社, 2004. 1—5.
- [8]赵强. 精通 JSP 编程[M]. 北京: 电子工业出版社, 2006. 2—12.
- [9]孙卫琴. 精通 Hibernate: Java 对象持久化技术详解[M]. 北京: 电子工业出版社, 2005. III—IV.
- [10]Craig Walls, Ryan Breidenbach, 李磊, 程立, 周悦虹译. Spring in action 中文版[M]. 北京: 人民邮电出版社, 2006. 33—35.
- [11]Ryan Aseleson, Nathaniel T. Schutta, 金灵等译. Ajax 基础教程[M]. 北京: 人民邮电出版社, 2006. 13—15.
- [12]IBM. Eclipse 平台入门
[EB/OL]. <http://www.ibm.com/developerworks/cn/linux/opensource/os-ecov/>, 2004-1-10
- [13]David Gallardo, Ed Burnette, Robert McGoven. Eclipse in Action[M]. Greenwich: Manning, 2002. 32—33.
- [14]孙卫琴, 李洪成. Tomcat 与 Java Web 开发技术详解[M]. 北京: 电子工业出版社, 2004.
- [15]Y. Daniel Liang. Introduction to Java Programming, Comprehensive Version, Fifth Edition[M]. U.S.: Prentice Hall. 2004. 11—14.
- [16]Mysql. MySQL 5.1 参考手册[EB/OL]. <http://dev.mysql.com/doc/refman/5.1/zh/index.html>, 2005-11-15.
- [17]刘彬. JSP 数据库高级教程[M]. 北京: 清华大学出版社, 2006. 63—64.
- [18]E. Zitzler, L. Thiele. Multi-Objective Evolutionary Algorithms A Comparative Case Study And the Strength Pareto Approach[J]. IEEE Transactions of Evolutionary Computation, 1999, 3(4): 257—271.
- [19]李小勇. 题库管理系统中的自动化组卷算法[J]. 西北师范大学学报(自然科学版), 2002, 38(4): 41—43.

- [20] 谢平. 基于框架模式的试题库智能组卷系统[J]. 华东交通大学学报, 1998, 15(4) : 58—63.
- [21] Holland J. Adaptation in natural and artificial systems[M]. Ann Arbor: University of Michigan Press, 1975. 88—105.
- [22] David E. Goldberg. Genetic algorithms in search, optimization and machine learning[M]. New York: Addison-Wesley Publishing Company Inc, 1989. 59—308.
- [23] David B. Fogel. An introduction to simulated evolution optimization[J]. IEEE Trans on Neural Networks, 1994, 5(1): 3—14.
- [24] M. Kantardzic, 闪四清等译. 数据挖掘——概念、模型、方法和算法[M]. 北京: 清华大学出版社, 2003. 202—240.
- [25] 徐洁磐, 马玉书, 范明. 知识库系统导论[M]. 北京: 科学出版社, 2000. 25—60.
- [26] F. Herrera, M. Lozano, J. L. Verdegay. Tuning Fuzzy Logic Controllers by Genetic Algorithms[J]. International Journal of Approximate Reasoning, 1995. 12(3): 115—123.

外文资料

Towards Increasing Web Application Productivity

Source: 2004 ACM Symposium on Applied Computing

ABSTRACT

In this paper we present and discuss a template/meta-data based partial code generation system supporting web application development. Seamlessly incorporating the recent top-notch technologies, the framework maximally exploits the capabilities of the underlying implementation technologies. Our approach primarily benefits the framework and code developers. In addition, the complete separation of data model, navigation model, and presentation model reflects on a more general conceptual process that would decouple the technique and methodology from its underlying technology choices. The decoupling between the generated code and the code that is necessarily added later on through other development pathways than the generator deals with the incremental changes and adaptations of the models in the face of an operational system, therefore further enhances the extensibility, maintainability, and reusability of the generated applications.

Categories and Subject Descriptors

D.2.2 [Software Engineering]: Design Tools and Techniques –
computer-supported software engineering, evolutionary prototyping.

General Terms

Management, Performance, Design, Reliability

Keywords

Web application development, framework, software architecture, automatic program generation and regeneration

1. INTRODUCTION

Involving with multiple disciplines relating to the computing and networking technologies, the development of web applications can be quite complex, costly and time-consuming, if not supported by a practical methodology [9]. In order to increase the productivity of web application development, a wealth of languages and standards and techniques have been created. On the basis of our consulting experience in industry, we believe that the following technologies dominate in the present web application realm. The Java 2 platform, Enterprise Edition (J2EE) defines de facto standard software architecture for multi-tier interactive web applications [16]. Meanwhile, application servers, which most commercial products available today are

based on the J2EE platform, provide system-level environmental support for web application development that prevent developers from tedious coding of system level utilities [13]. According to the J2EE paradigm, the architecture of a typical web application reflects the traditional three-tier architectural model [15]: front-end, middle tier, and back-end. The Hyper-Text Markup Language (HTML), JSP, and JavaScript provide powerful means to describe dynamic web interfaces for applications; and Java Servlet provides underlying support to construct web page information on the fly. The concepts of session bean and entity bean in EJB technique clearly decouple the code serving for business logic processing from the code of data, thus provide a commanding infrastructure to construct the middle-tier of applications. Furthermore, the EJB engine provides a higher-level abstraction over the underlying data model storage devices – the traditional relational database, which is considered to be the back-end of a web application. In addition, some other techniques further assist to firm the J2EEbased development methodology, such as Struts [23], Ant [1], and XML [21]. Struts provides an open source unified front-end Model-View-Controller (MVC) framework [10] for web applications. Ant is a Java based prompt build tool that allows developers to utilize XML technique to configure the relationships between different building blocks and generate the compiled version. As a universal description language, XML provides a de facto standard language to define and describe web applications.

J2EE-based techniques have enabled the explosion in the number of web applications that we have witnessed to date. More importantly, these techniques formalize the architecture of a typical web application; therefore make it feasible to expedite the development of web applications by exploiting automatic program generation techniques. In this paper we present and discuss a template/meta-data based partial code generation system supporting web application development. Seamlessly incorporating the XML and J2EE family of technologies, the framework maximally exploits the capabilities of the underlying implementation technologies. The rest of this paper is organized as follows. In Section 2, we discuss related work in web development techniques. In Section 3, we discuss our template/meta data based web code generator. In Section 4, we discuss an example. Conclusions and future work directions are discussed in Section 5. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or

republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM SAC'04, March 14-17, 2004, Nicosia, Cyprus.

Copyright 2004 ACM 1-58113-812-1/03/04...\$5.00.

2004 ACM Symposium on Applied Computing

2. RELATED WORK

A variety of research work has been conducted to expedite and automate the development of web applications. Among the various efforts, a lot of work focuses on assisting conceptual modeling of web applications. HDM [12] defines a popular model dedicated for hypermedia application design, which divides the conceptual schema into two categories: structural and navigational. Its variant recent descendent formats include HDM2000 [2], OOHDM [22], WebML [5], and HDM-lite [9]. Web Modeling Language (WebML) [5] is a notation of designing a complicated web site at the conceptual level, which defines a web site along five dimensions: data content, page composition, navigation links, page presentation, and customization features. HDM2000 and WebML intend to integrate a number of considerations typical of large-scale Web applications, such as requirements elicitation, customization aspects, operational aspects, and transactional aspects. HDM-lite [9] adds a presentation schema to the conceptual design of web applications. Its development environment Autoweb [9] stores the conceptual schemas together with the data content in the development database. Jweb [3] provides a design and prototyping environment that integrates XML technology with HDM in order to help design the conceptual schema. There are also other conceptual design techniques. Conallen extends the Unified Modeling Language (UML) [4] notation to model web specific elements, thus making it feasible to model whole web applications with the UML [7]. Gaedke uses WebComposition and the WebComposition Markup Language (WCML) to present a systematic approach for code reuse in component-based web applications [11]. IIPS models navigational structure, compositional structure, and user interface through ontology [20]. The formalized interfaces of J2EE techniques make it extremely suitable for code generation. There are several code generation products and tools aiming at accelerating the development process of web applications [18]. Torrisoft is an on-going environment aiming at helping web site development utilizing Web Modeling Language [5]. IIPS [20] provides a set of visual tools supporting web site generation from the conceptual design model. Some generation tools aim to help generate specific parts of a web application. CodeCharge [6] helps to generate

front-end codes in selected languages such as ASP, JSP, and ColdFusion; KoolFrog [19] helps to generate PL/SQL routines from Oracle database; EJBGen [8] helps generating EJB 2.0 code. Tekadence [25] offers a point-and-click programming interface and visual design tool to help build up Java applications. GslGen [14] is a generic-purpose code generator from XML file and schema file using generalpurpose schema language (GSL). Tekadence and GslGen generators are bound to specific script languages. JeeWiz [17] is a product that provides an environment to help generate code for J2EE-based web applications, such as EJB beans and deployment descriptor.

3. TEMPLATE / META-DATA BASED

PARTIAL CODE GENERATION

In this paper we present and discuss a template/meta-data based partial code generation system supporting web application development. Seamlessly incorporating the XML and J2EE family of technologies, the framework maximally exploits the capabilities of the underlying implementation technologies. We make an assumption here that a web application is represented by a set of web pages linked together by different operations. The web pages shown in a web application are dynamically generated based on the previous pages and the actions operated by users through the web browsers. Based on these basic assumptions, we identify three components in the conceptual design of a web application: data model, navigation model, and presentation model. Data model contains the information content to be shown on every web presentation page in an application. For example, a page displays a student's address information, thus the data model of the page is the student's address object. Navigation model manages the navigation links among different pages, therefore handles the control flow of a web application. For example, from a student's address display page, a student can navigate to a page that displays his financial account information. Presentation model stores the display features of every web page to be exhibited, e.g. the layout of each data item of a student's address information. Following the Object-Oriented paradigm, data models can be represented by data objects; and navigation models can be represented by the relationships among data models. These two types of models are naturally designed and developed by developers. Meanwhile, the presentation model focuses on the look and feel of every web page to be generated, therefore it is more natural for visual designers to handle parallel to the code developers. The information of the presentation model can be stored separately, and applied to the corresponding web page at run time. As a result,

here we will concentrate on the automatic code generation of the first two types of models. The complete separation of data model, navigation model, and presentation model reflects on a more general conceptual process that would decouple the technique and methodology from its underlying technology choices. We will discuss in more detail how our approach facilitates the code generation of a web application from the first two models.

3.1 Data Model Generation

We designed and implemented an XML-based web code generator called WebGen. Similar to existing code generators, WebGen defines a set of templates to generate the code. Based on our investigation of J2EE family of technologies and utilizing state-of-the-art XML technology, which detailed information can be found from our previous work [26], WebGen provides a complete set of fourteen templates that can help automatically generate the code of a fully functioning running web application system. This template system provides code templates from the front-end to the back-end of a web system. JSP and form bean contain templates for JSP pages and their associated data models. Pre-action and Post-action templates define actions to format a web page and to collect information interactively supplied by page users. The Java bean template defines the business data object. We use the entity-bean home interface template, remote interface template, and implementation template to generate EJB entity beans. Likewise, the session-bean home interface template, remote interface template, and implementation template generate EJB session beans. The deployment descriptor template generates deployment descriptor segments for the corresponding entity beans and session beans. The service layer template generates service method signatures. The database schema template generates SQL statements for creating and deleting tables in the database. Given the large amount of code that we generate

```
/**  
 * This class is to hold the contents of entity bean  
 */  
public class <NAME>Bean implements Serializable {  
/**  
 * Constructor  
 */  
public <NAME>Bean() {  
<*ATTR> _<_ATNAME> = new <ATYPE>();
```

```

</ATTR>
}
<*ATTR>
/**
* This method retrieves the attribute of <ATNAME>
* @returns <ATYPE> - <ATNAME>
*/
public <ATYPE> get<ATNAME>() {
return _<_ATNAME>;
}
/**
* This method sets the attribute of <ATNAME>
* @param <ATYPE> - <ATNAME>
*/
public void set<ATNAME>(<ATYPE> <_ATNAME>) {
<_ATNAME> = <_ATNAME>;
}
</ATTR>
/**
* Private attributes
*/
<*ATTR> private <ATYPE> _<_ATNAME>; </ATTR>
}

```

Figure 2. A Java class template file =DATA=NAME=Bean.java

automatically, a developer merely needs to define the data model and to implement the business logic in order for the web application to be complete. We use a concept of data file to store the data model of each web page. A data file is an XML file that describes the content of a data model to be merged into template files. Every data file is applied to template files to generate corresponding code. Figure 1 is an example of a data file that defines a data model of *StudentGrade*. As shown in Figure 1, for every item of the corresponding data model, one can define its name and data type. *StudentGrade* contains four data items: student id, year, term, and grade information. All four items have the same data type as “String”. The attribute *Key* is utilized to store the primary key internally. In addition, any aspect can be specified as a attribute of an item. For instance, as illustrated in Figure 1, the attribute *Unique* is utilized to

specify whether the particular item can be treated uniquely or not. A template file itself is a code file, which contains tags that must be replaced at generation time by the corresponding values from the data files. Figure 2 is a simplified template file for a Java bean class. It can be used to generate a fully functional Java class file, which comprises a constructor and a set of getters and setters. WebGen defines three categories of tags: simple tag, repeating pattern, and conditional. A simple tag is a single tag, which will be replaced by the corresponding value defined in the data file. For example, tag <ATYPE> in Figure 2 can be replaced by “String” defined in data file in Figure 1. A repeating pattern is an area that will repeat the pattern once per element of a list of data, such as the method body of constructor in Figure 2. The syntax of a repeating pattern is: <*PATTERNNAME> ...pattern...</PATTERNNAME>. Simple tags can be contained in the pattern. A conditional provides different patterns as options to choose from. The syntax of conditional is: <?VAL1=VAL2>...pattern...</?>. The semantics is that, if VAL1 equals to VAL2, then the pattern contained in the tag will appear in the generated file, otherwise it will not. A data file can be applied to multiple template files, and a template file can also be applied to different data files.

A configuration file provides knowledge to the WebGen, such as the style and criteria of the generation, in XML format. It bridges between data files and template files. Separated from data files, configuration files enhance the reusability of data files over different template files. There are two basic types of information that a configuration file needs to pass to WebGen. First, the output location of the generated files needs to be specified. The generated files can be configured in a way that they are stored in hierarchical directories. Second, a configuration file informs the generator about the relationships between template files and data files. This relationship is generally a many-to-many relationship. Figure 3 is an example of a configuration file for the corresponding data model *StudentAddress*. As shown in Figure 3,

```

<WGenerator>
<OBJ NAME="StudentAddress" TNAME="student_address">
<ATTR ATNAME="Key" ATYPE="String" UNIQ="YES"/>
<ATTR ATNAME="StudentId" ATYPE="String" VALID="YES"/>
<ATTR ATNAME="AddressLine1" ATYPE="String" UNIQ="NO"/>
<ATTR ATNAME="AddressLine2" ATYPE="String" UNIQ="NO"/>
...
</OBJ>

```

```

</WGenerator>

Figure 1. A simple example of a data file

<WGenerator outputDir ="univ" data="studentAddress">
<CODE NAME="StudentAddress">
<template      name=="OBJ=NAME=Bean.java"      outputDir="businessObject"
    change="ALL"/>
<template name=="OBJ=NAME=EJB.java" outputDir="ejb\entity" change="ALL"/>
<template      name=="OBJ=NAME=EJBHome.java"      outputDir="ejb\entity"
    change="ALL"/>
<template      name=="OBJ=NAME=EJBImpl.java"      outputDir="ejb\entity"
    change="ALL"/>
<template      name=="OBJ=NAME=SessionEJB.java"      outputDir="ejb\session"
    change="ALL"/>
...
<template          name="ApplicationResources.properties"
    outputDir="JRun\servers\univ\WEB-INF\classes" change="PART"/>
</CODE>
</WGenerator>

```

Figure 3. Example of a configuration file

the Java bean file will be stored in directory *univ/businessObject*; the data file *StudentAddress* will be used to generate the business object file, database table schema, and entity bean files, among others.

In real world situations the code-generation approach imperatively runs into the problems of its closed world assumption of control. Data models may change overtime, and business logic may change during the life cycle of the development. Normal code generation tools cannot handle updated generated files; thus their powers eliminate. Facing the incremental changes and adaptations of the models in the full life cycle of web application development, it is important to separate between the generated code and the code that is necessary. In order to address this issue most code-generation supporting systems are facing, WebGen utilizes a simple configurable solution. Developers are able to decide whether the target file will be completely regenerated, or partly replaced, by specifying the value of attribute *Change* to be *All* or *Part* respectively, as shown in Figure 3. If it is specified as *All*, the target file will be completely regenerated every time the generator is invoked. Otherwise the generator will first search for the target file. If this file is not found, the

generator will create a new code file. Otherwise, if a target file is found, the generator will search for the corresponding portion in the target file and replace it if the portion exists, or will add a new part if this portion does not exist. The rest of the portions of the file will be kept untouched. The separation of a generated portion and an immutable portion in one file enables regenerating a file without removing the embedded business logic. Since changes to data models are inevitable during design and implementation, this feature can improve the flexibility of the code generated.

3.2 Navigation Model Generation

As we discussed in the previous section, a navigation model stores the relationship among different web pages, and decide what a page can navigate to in the next step. In a business web application, the navigational information normally exhibits as a set of visible menu items on web pages. Therefore in our approach, we utilize the concept of menu system in order to manage the navigation model. The navigation model is stored in a menu system data structure; and the corresponding menu system generator (MSG), which is part of WebGen, assists to generate the code to display the menu system, including navigational information associated with each menu item. We follow the menu tree specified in Java Swing [24] as the data structure of our menu system. Since the description of a menu system is configured in an XML file, “visual designers” can assert their ideas by modifying the menu system independent of the coding of the system. Further, programmers do not need to do any coding of the menu system. MSG accepts configurations of a menu system from an XML file. A developer only needs to specify the attributes of each component and their relationships in an XML configuration file. Figure 4 shows a piece of a menu system configuration file. A menu bar is defined with the name *StudentRecords*. It contains several menu items and one of them is called *ViewGrades*. This item displays the text “View Grades”. It contains no image, and holds a link to “/app/viewGrades.do”. This menu item is visible and selectable. Initially, it is not selected by default, so that it is not highlighted, and this menu item does not contain any other menu bars. The configuration also defines the size of the menu item, together with the alignment features. MSG translates menu system configurations into a menu tree structure and stores it in menu system storage classes, and HTML code will be automatically generated.

3.3 WebGen Based Approach

With our WebGen, the development of web applications can be applied as a fast-prototyping methodology. Designer and developers first functionally decompose

a web application system into web pages, grouped by a menu system. For every web page, the contained data model is identified, and recorded in a data file; and the generation criteria is written to the corresponding configuration file. Meanwhile, the hierarchical menu system is defined as recorded in the menu system specification files. Then the WebGen is run to generate the prototype of the web application. This prototype can be first utilized to elicit and adjust the current requirements specifications. In this sense, the prototype is also called a mockup. In this paper we will use mockup and prototype interchangeably. Any changes to the current data model and navigational model can be integrated into the prototype via re-running the WebGen. After the prototype obtains the clients' approval, it can be transferred to the developers to embed the specific application logic. At any time, the WebGen can be applied upon the up-to-date prototype system to incorporate further changes on the data models and navigation models.

4. EXPERIMENT

One of the projects where our approach is actually being used is the design and development of an e-University suite, which is a web application that students, faculty members, staff members, and administrators use for such services as admissions, student records, registration, and financial services.

After the whole application was delivered, the code was examined for analysis. We found that only 9.62% of the final code was eventually coded by developers; while 90.38% of the system was automatically generated. As a result, since over 90% of the code was automatically generated, the code quality was guaranteed, the development process was greatly shortened, and the subsequent test phase shortened. As a matter of fact, the project was originally assigned to a team of four full-time developers with time frame of four months. With the adoption of WebGen, the project was accomplished by one a full-time developer (the first author of this article) together with a part-time JSP designer and programmer.

Our initial experience with this system indicates that WebGen can

```
<menu name="studentRecords">
  <menuItem name="viewGrades" text="View Grades"
    image="null" url="/app/viewGrades.do"
    bgrdColor="green" visible="y" enabled="y"
    selected="n" container="studentRecords"
    child="null" width="76" height="20"
    align="center" valign="top">
```

```
linkClass="menuLinkColor"
selectedLinkClass="menuActiveLinkColor">
</menuItem>
...//other menu items
</menu>
```

Figure 4. menu system configuration

provide improvements in development speed while enhancing the reliability of the resulting applications.

There were also issues that arose from our experiment. One of the most important advantages of using our approach is that our WebGen could automatically generate code from the front-end to the back-end. However, we require that the mappings between the front-end JSP pages and business objects that relate to the backend system cannot be too complicated. Therefore, a senior-level architect is required to decompose and design the object model for big-scale complex systems. Overall, we feel that more WebGen internal facilities should be provided to enhance the ability of our approach to support large-scale complicated web applications.

5. ASSESSMENTS, CONTRIBUTIONS,

AND FUTURE WORK

In this paper we present and discuss a template/meta-data based partial code generation system supporting web application development. Seamlessly incorporating the XML and J2EE family of technologies, the framework maximally exploits the capabilities of the underlying implementation technologies. Our approach primarily benefits the framework and code developers. In addition, the complete separation of data model, navigation model, and presentation model reflects on a more general conceptual process that would decouple the technique and methodology from its underlying technology choices. The decoupling between the generated code and the code that is necessarily added later on through other development pathways than the generator deals with the incremental changes and adaptations of the models in the face of an operational system, therefore further enhances the extensibility, maintainability, and reusability of the generated applications.

We understand that our approach requires discipline in design and development. However, in our experience, the benefits largely justify this additional effort. Our method leads developers to quickly construct a prototype of the web application; the analysis phase and the development phase are clearly decoupled. The availability of the mockup system gives customers the opportunity to participate in system design at

early stages of development. WebGen helps build a running mockup system with little developer intervention. Furthermore, the mockup system can be easily regenerated based on user feedback. Our future work will include an Interactive Development Environment (IDE) to support the editing of conceptual models. Currently developers need to set up data models and configurations as a set of XML files. An IDE can automatically support the definition and maintenance of these files, while providing a graphical editor to assist the design and maintenance of conceptual models.

ACKNOWLEDGEMENT

We deeply appreciate the valuable feedback from reviewers.

6. REFERENCES

- [1] <http://ant.apache.org>.
- [2] Baresi, L., Garzotto, F., & Paolini, P. From web sites to web applications: new issues for conceptual modeling. Proceedings of WWW Conceptual Modeling Conference, (Salt Lake City, 2000).
- [3] Bochicchio, M., and Palano, R. Prototyping web applications. in Proceedings of the 2000 ACM Symposium on Applied Computing (SAC) (Como, Italy, 2000), 978-983.
- [4] Booch, G., Jacobson, I., and Rumbaugh, J. The Unified Modeling Language User Guide. Addison-Wesley, Reading MA, 1998.
- [5] Ceri, S. Fraternali, P., and Bongio, A. Web Modeling Language (WebML): A Modeling Language for Designing Web Sites. Proceedings of 9th International World Wide Web Conference, (Amsterdam, Netherlands, May 2000), 15-19.
- [6] http://www.kapitec.com/Produits/CodeCharge/en/codecharge_std.htm.
- [7] Conallen, J. Modeling web application architectures with UML. Communications of the ACM, 42, 10, (October 1999),63-70.
- [8] <http://www.beust.com/cedric/ejbgen/#introduction>.
- [9] Fraternali, P., and Paolini, P. Model-driven development of web applications: the AutoWeb system. ACM Transactions on Information Systems (TOIS), 18, 4, (October 2000), 323-382.
- [10] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. Design Patterns. Addison Wesley, 1994.
- [11] Gaedke, M., and Rehse, J. Supporting compositional reuse in component-based web engineering. in Proceedings of the 2000 ACM Symposium on Applied Computing (SAC)(Como, Italy, 2000), 927-933.
- [12] Garzotto, F., Paolini, P., and Schwabe, D. HDM – a model based approach to

- hypermedia application design. ACM Transactions on Information Systems (TOIS) 11, 1, (January 1993), 1-26.
- [13] Grundy, J., et al. Extending a persistent object framework to enhance enterprise application server performance. In Proceedings of the 13th Australasian conference on database technologies, (Melbourne, Victoria, Australia, 2002), 57-64.
- [14] <http://www.imatix.com/html/gslgen/index.htm>.
- [15] Monson-Haefel, R. Enterprise JavaBeans. O'Reilly & Associates, Inc. ISBN: 0-596-00226-2. 3rd edition, 2001.
- [16] <http://java.sun.com/j2ee>.
- [17] <http://www.jeewiz.co.uk/index.html>.
- [18] <http://www.jostraca.org/links.html>.
- [19] <http://www.koolfrog.com>.
- [20] Lei, Y., Motta, E., and Domingue, J. IIPS: An intelligent information presentation system. in Proceedings of the 7th International Conference on Intelligent User Interfaces, (SanFrancisco CA, 2002), 200-201.
- [21] McLaughlin, B., and Loukides, M. Java and XML. O'ReillyJava Tools, 2001.
- [22] Schwabe D., and Rossi G. An Object Oriented Approach to Web-Based Application Design, Theory and Practice of Object Systems, 4, 4, J. Wiley, 1998.
- [23] http://jakarta.apache.org/Struts_
- [24] <http://java.sun.com/docs/books/tutorial/uiswing/mini/index.html>.
- [25] http://industry.java.sun.com/solutions/products/by_product/0,2348,ejb-5791-28,00.html.
- [26] Zhang, J., and Buy, U. A framework for the efficient production of web applications. Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC),(Kemer-Antalya, Turkey, June 30-July 3, 2003), 419-42.

中文译文

面向提高 Web 应用生产力

出处：2004 年 ACM 计算应用专题讨论会

Jia Zhang	Jen-Yao Chung
Department of Computer Science	IBM T. J. Watson Research
Northern Illinois University	Yorktown Heights,
NY 10598	
DeKalb, IL 60115	
01-312-718-2468	01-914-945-3422
jiazhang@cs.niu.edu	jychung@us.ibm.com
Carl K. Chang	
Department of Computer Science	
Iowa State University	
Ames, IA 50011	
01-515-294-4377	
chang@cs.iastate.edu	

摘要

本论文介绍并讨论了基于模板/汇总数据（template/meta-data）的部分代码生成系统支持 Web 应用的开发。它将无缝的集成当今最顶尖的技术，该框架将最大限度的挖掘基本工具技术的能力。我们的方法将主要有益于框架和代码的开发者。此外，完全独立的数据模型，导航模型和显示模型反映了更多的一般概念过程，这个过程将把技巧和方法从它们的基本技术选择中分离出来。在面对一个业务系统时，在合成码和以后必须通过其他方式添加的代码之间去耦合而不是通过发生器来处理不断增长的变化和模型的改变，因此将更进一步增强生成的应用的可扩充性，可维护性和可重用性。

分类和主题描述

D. 2. 2 [Software Engineering]: Design Tools and Techniques – computer-supported software engineering, evolutionary prototyping.

通用词

Management, Performance, Design, Reliability

关键字

Web application development,
Framework ,
software architecture ,
automatic program generation and regeneration

1、介绍

由于包含了许多与计算机和网络相关的技术上的约束, 开发网络应用如果没有被一项实践的方法论支持的话将会是一项复杂的, 耗费昂贵的和费时的工作 [9]。为了提高网络应用开发的生产率, 大量的语言, 标准和技术被创造出来。基于我们在工业方面的咨询经验, 我们认为以下的技术在目前的 Web 应用领域具有优势。在 Java2 平台, 企业版(J2EE 平台)为多层交互式 Web 应用定义了事实上的标准软件结构[16], 同时, 应用服务器, 其中大多数商业产品可供今天使用的都是基于 J2EE 平台的, 他们为 web 应用提供了系统级环境的支持, 这可以使得开发者从恼人的系统级代码中解脱出来[13]。根据 J2EE 的范例, 构筑一个典型的 Web 应用反映了传统的三层建筑模型[15]: 前端, 中间层以及后端, 超文本标记语言(HTML), JSP 和 JavaScript 为应用提供有力的手段来描述动态网页界面, 同时, Java Servlet 为网页页面的信息提供了基本的支持。在 EJB 技术中会话bean 和实体bean 清晰的分开了为商业逻辑过程服务的代码和为数据服务的代码, 因此它们为构建中间层的应用提供了一个关键的基础。此外, EJB 引擎为被认为是后端 web 应用的基本数据存贮装置——关系型数据库提供了高级的抽象。此外, 一些其他的技术协助进一步坚定了 J2EE 的开发方法, 如 struts[23], Ant[1] 以及 XML[21]。struts 为 Web 应用提供了一个开放源码的统一前端模型—视图-控制器(mvc)的框架[10]。Ant 是一种基于 Java 迅速构造工具, 让开发商使用 XML 技术来配置不同构造模块的关系和生成已编译版本。作为一个普遍性的描述语言, XML 提供了一种事实上的标准语言来定义和描述的 Web 应用。

基于 J2EE 的技术已经使我们目睹了很多 web 应用方面的爆炸性应用。更重要的是, 这些技术使得一项典型的 web 应用结构体系能够成型, 因此, 通过开发自动的程序生成技术以加速 web 应用的开发是可行的。在本论文我们介绍并讨论了基于模板/汇总数据 (template/meta-data) 的部分代码生成系统支持 Web 应用的开发。它将无缝的集成 XML 和 J2EE 系列技术, 该框架将最大限度的挖掘基本工具技术的能力。本论文的剩余部分将会如下组织。在第二部分, 我们会讨论在 web 开发技术中的相关工作。第三部分, 我们会讨论我们的基于模板/汇总数据的 web 代码生成器。第四部分, 我们会讨论一个例子。结论和未来的工作方向将会在第五部分讨论。

2、相关工作

大量的研究工作已经加速和使得 web 应用开发更加自动化了。在多方努力下, 大量的工作重点放在协助 Web 应用的概念建模。HDM [12] 为超媒体 (hypermedia) 定义了一个专用的模型, 它把概念模式分为两类: 结构和导航。它近期的各种变化形式包括: HDM2000 [2], OOHDM [22], WebML [5] 和 HDM-lite [9]。web 建模语言(WebML) [5] 是一项从概念上设计复杂的 web 站点的注释, 它确定了一个网站的五个方面: 数据内容页组成, 导航链接, 页面的介绍和定制的

特点。HDM2000 和 WebML 打算整合一些典型的大型 Web 应用，如需求获取，定制方面，业务方面，交易方面。HDM-lite [9] 为 web 应用的概念设计添加了一个提出图表（presentation schema），它的开发环境 Autoweb[9]

在开发的数据库中保存了概念模型和数据内容。Jweb[3] 提供了结合 HDM 的 XML 技术的设计和样机环境，这样做是为了帮助设计概念模式。同时，还存在其它一些概念设计的技术。Conallen 为模型化网络的具体元素而扩展了统一建模语言（UML）[4] 的概念，因此，使得用 UML 来描述整个网络应用是可行的[7]。Gaedke 使用 WebComposition 和 WebComposition 标记语言（WCML）来表示一种系统化的方法，这种方法的目的是为了在基于构件的 web 应用中能够代码重用[11]。IIPS 通过本体论模型化了导航结构，组成机构和用户接口[20]。J2EE 技术定型的接口使得它非常适合代码的生成。有很多代码生成的产品和工具能都加速网络应用的开发进程[18]。Torrisoft 是一个持续的环境，旨在帮助利用网站建模语言开发网站[5]。IIPS [20] 提供一整套可视化工具，支持网站从概念设计模型到生成。一些生成工具是为了帮助一个网络应用生成具体的部分。CodeCharge 帮助使用选定的语言例如 ASP，JSP 和 ColdFusion 生成前端的代码；KoolFrog[19] 帮助从 Oracle 数据库中生成 PL/SQL 例子；EJBGen[8] 帮助生成 EJB2.0 代码。Tekadence[25] 提供了一个指向和点击（point-and-click）的程序界面和可视化的设计工具来帮助建立 Java 应用。GslGen [14] 是一个通用的代码生成工具，通过它可以使用多用途的图表语言（GSL）生成 XML 文件和图表文件。Tekadence 和 GslGen 被捆绑到具体的脚本语言中。JeeWiz[17] 是一项提供帮助生成基于 J2EE 的网络应用，如 EJB beans 和 deployment descriptor，的环境。

3、模板/汇总数据的部分代码生成

在本论文我们介绍并讨论了基于模板/汇总数据（template/meta-data）的部分代码生成系统支持 Web 应用的开发。它将无缝的集成 XML 和 J2EE 系列技术，该框架将最大限度的挖掘基本工具技术的能力。我们在这里做出一个假设：一个网络应用代表着一系列的网页通过不同的方式连接在一起。在网络应用中的网页根据前一页和用户通过浏览器执行的动作动态生成。基于这些基本的假设，我们在一个网络应用的概念设计时定义了三个组成成分：数据模型，导航模型和表示模型。数据模型包含的信息内容将在一个应用中显示在每个网站页面上。举例来说，一个页面显示出学生的地址资料，因此数据模型的页面是学生的地址对象。导航模型管理导航链接不同的页面，因此在一个 Web 应用处理的控制流。例如，从学生的地址显示页，学生可以导航到一个显示他的金融账户的页面。表示模型存储将要被显示的每个网页的显示特征，比如一个学生地址信息的每一个数据项的布局。

根据面向对象的理论，数据模型能够用数据对象表示，导航模型能够被一些

有关系的数据模型表示。这两类模型自然是由开发者来设计和开发。同时，表示模型面向的焦点是每个生成网页的外观和感觉，因此，它自然是由视觉设计这与代码设计者来并行的处理。表示模型的信息能够单独的存放，并且在使用时能够应用到相应的网页中。因此，我们将关注的是对于前两种模型的自动代码生成。完全独立的数据模型，导航模型和表示模型反映了更多的一般概念的过程，它将从基本的技术选择中分离技术和方法。我们将从更多的细节上讨论怎样才能使我们的方法在一个网络应用的前两个模型上使得代码生成更容易。

3.1、数据模型生成

我们设计并实现了基于 XML 的代码生成器 WebGen。与现在已经存在的代码生成器一样，WebGen 定义了一系列生成代码的模板。基于我们对 J2EE 系列技术的研究和利用最新的 XML 技术，这些细节信息能在我们以前的工作中找到[26]，WebGen 提供了一系列完整的 14 个模板，通过它们能够自动的生成一个具有完整功能的网络应用系统的代码。这个模板系统不仅能够为一个 web 系统提供前台的代码生成，还能提供后台的代码生成。JSP 和形式 bean 含有模板 JSP 页面及其相关的数据模型。预先行（Pre-action）和后继行（Post-action）的模板定义了行为（actions）来形成一个 web 页面和保存由页面用户发出的交互信息。Java bean 模板定义了商业数据对象。我们使用实体 bean 库的接口模板，远程接口模板，和工具模板来生成 EJB 实体 beans。同样的，会话 bean 库的接口模板，远程接口模板，和工具模板将会生成 EJB 会话 beans。部署描述模板生成部署描述各环节相应的实体 bean 和会话 bean。服务层模板生成了服务方法的签名，数据库图表模板位在数据库中创建和删除表生成了 SQL 语句声明。由于已经自动生成了大量的代码，开发者仅仅需要定义数据模型和为完成 web 应用而实现商业逻辑。

```

<WGenerator>
<OBJ NAME="StudentAddress" TNAME="student_address">
<ATTR ATNAME="Key" ATYPE="String" UNIQ="YES"/>
<ATTR ATNAME="StudentId" ATYPE="String" VALID="YES"/>
<ATTR ATNAME="AddressLine1" ATYPE="String" UNIQ="NO"/>
<ATTR ATNAME="AddressLine2" ATYPE="String" UNIQ="NO"/>
...
</OBJ>
</WGenerator>

```

Figure 1. A simple example of a data file

我们使用了一个数据文件（data file）的概念来为每一个 web 页面存储数据模型。一个数据文件就是一个 XML 文件，它用来描述一个将要被用到模板文件中的数据模型的内容。每一个数据文件都会被应用到模板文件中以生成相应的代码。图 1 是一个数据文件它定义了一个学生年级（StudentGrade）的数据模型。正如图 1 所显示的与数据模型相对应的每一条都可以定义数据名字和类型，“学

生年级”里包括四个数据条目：student id, year, term 和 grade information。所有四个条目都是相同的数据类型“String”。Key 的属性是用来存储内部的主键。此外，每一个条目都能从它的任意方面地定义其属性，例如，正像图 1 所显示的那样，属性 Unique 能决定每一个具体的条目是否是唯一的。

一个模板文件其实就是一个代码文件，它里面包含的标签（tags）在生成代码的时候必须被数据文件中的相应的数值代替。图 2 是一个专门为一个 Java bean 生成的简单的模板文件。它能被用来生成一个拥有完全功能的 Java 类文件，它包含了一个构造方法和一系列获

```


    /**
     * This class is to hold the contents of entity bean
     */
    public class <NAME>Bean implements Serializable {

        /**
         * Constructor
         */
        public <NAME>Bean() {
            <*ATTR>      _<_ATNAME> = new <ATYPE>();
            </ATTR>
        }

        <*ATTR>
        /**
         * This method retrieves the attribute of <ATNAME>
         * @returns <ATYPE> - <ATNAME>
         */
        public <ATYPE> get<ATNAME>() {
            return _<_ATNAME>;
        }

        /**
         * This method sets the attribute of <ATNAME>
         * @param <ATYPE> - <ATNAME>
         */
        public void set<ATNAME>(<ATYPE> <_ATNAME>) {
            _<_ATNAME> = <_ATNAME>;
        }
    </ATTR>

        /**
         * Private attributes
         */
        <*ATTR>  private <ATYPE> _<_ATNAME>; </ATTR>
    }


```

Figure 2. A Java class template file =DATA=NAME=Bean.java

取（getters）方法和设置（setters）方法。WebGen 定义了三种标签：简单标签(simple tag), 重复模式(repeating pattern)和条件模式(conditional)。简单标签就是一个单独的标签，它将被数据文件中的相应的数值代替。例如，图 2 中的标签<ATYPE>能够被图 1 中定义为“String”的数据代替。重复模式是一个区域，它将重复一个数据列表中的每一个元素的模式，例如图 2 中的构造方法体。重复模式的句法是：<*PATTERNNAME> …pattern…</PATTERNNAME>。简单标签能够包含在模式中。条件模式提供了不同的可选择的模式。条件模式的句法是：<?VAL1=VAL2>…pattern…</?>。该句法的意思是，如果 VAL1 等于 VAL2，则该模式包含的标签将会在生成文件中显示。反之，则不会在上面显示。一个数据文

件能多次应用在模板文件中，同样，一个模板文件也能多次应用在不同的数据文件中。

一个配置文件提供了应该给 WebGen 知晓的东西，例如，用 XML 生成的式样和标准。她连通了数据文件和模板文件。由于其独立于数据文件，所以配置文件增强了数据文件在不同模板文件中的可重用性。有两类基本的信息需要配置文件传给 WebGen：一、生成文件的输出位置需要具体化，根据它们存储的等级目录生成文件可以被配置。二、一个配置文件将通知生成器模板文件和数据文件之间的关系。这种关系通常是多对多（many-to-many）的关系。图 3 是一个与 StudentAddress 相适应的配置文件的例子。正如图 3 所显示的那样，Java Bean

```

<WGenerator outputDir="univ" data="studentAddress">
  <CODE NAME="StudentAddress">
    <template name="=OBJ=NAME=Bean.java" outputDir="businessObject" change="ALL"/>
    <template name="=OBJ=NAME=EJB.java" outputDir="ejb\entity" change="ALL"/>
    <template name="=OBJ=NAME=EJBHome.java" outputDir="ejb\entity" change="ALL"/>
    <template name="=OBJ=NAME=EJBImpl.java" outputDir="ejb\entity" change="ALL"/>
    <template name="=OBJ=NAME=SessionEJB.java" outputDir="ejb\session" change="ALL"/>
    ...
    <template name="ApplicationResources.properties" outputDir="JRun\servers\univ\WEB-INF\classes" change="PART"/>
  </CODE>
</WGenerator>

```

Figure 3. Example of a configuration file

文件将会被存储在目录 univ/businessObject 下，数据文件 StudentAddress 将会被用来生成商用对象文件，数据库表，实体 bean 文件等等。

在现实世界代码生成方法不可避免的将会遇到它在封闭世界试图控制的问题。数据模型也许会超时地改变，商业逻辑也许会在开发周期内改变。普通的代码生成器不能处理更新的生成文件，因此，它们将会被淘汰。面对在整个 web 开发周期里不断增加的改变和模型的不断改编，分开合成代码和必须代码是很重要的。为了处理这些问题很多代码生成的支持系统面市了，WebGen 利用了一个简单的可配置的解决方法。正如图 3 所显示的那样，开发者通过改变 Change 的属性为 All 或 Part 将能决定目标文件是被完全替换，还是仅仅是部分替换。如果选择的是 All 则目标文件将会在每次使用生成器时被完全的替换。否则，生成器将会首先搜索这个文件，如果文件不存在，生成器将会创建一个新的文件。如果文件被找到，则生成器将会寻找目标文件中相应的部分然后替换它，或者是在目标文件中增加一部分（如果这部分原来不存在的话）。文件的其它部分将会保持不变。在一个文件中分开生成的部分和不可改变的部分将使得重新生成一个文件时不用去掉已经嵌入到里面的商业逻辑。由于在设计和实施期间变化数据模型是必然的，这一特点可以改善代码生成的灵活性。

3.2 导航模型的生成

正如我们在前面讨论的，导航模型存储的是不同 web 页面的关系。并且它将决定在下一步中哪个页面将会被导航。在一个商业 web 应用中，导航信息通常是在网页上显示一系列可见的菜单。因此在我们的方法中，我们使用菜单系统的概念来管理导航系统。导航系统将会按照菜单系统的架构来存储，并且和作为

WebGen 系统一部分的菜单系统生成器 (MSG) 一致, 协助生成代码来显示菜单系统, 同时每个菜单条目中会包括导航信息。我们根据在 Java Swing[24]已经具体化的菜单树来作为我们的菜单系统的数据结构。由于菜单系统的描述已经在 XML 文件中配置好了, 所以“视觉设计者”能够独立于代码系统来根据他们的想法修改菜单系统。此外, 程序员不用为菜单系统写任何的代码。MSG 接受通过 XML 文件配置的菜单系统。开发者仅仅需要在一个 XML 文件中具体化每一个元素的属性和它们的关系。图 4 显示了一段一个菜单系统的配置文件。一个选项栏被用 StudentRecords 的名字声明。它包括很多选项条目, 其中一项是 ViewGrades。这个条目显示文本“View Grades”。它没有包含图片, 但支持一个链接“/app/viewGrades. do”。这个可选项是可见和可选择的。默认它是不可选择的, 因此它不是高亮的, 并且这个不包含其它任何的选项栏。配置文件同样将与其它特征一起定义可选项的大小。MSG 将菜单系统配置转换成一个菜单树的结构并且将它保存在菜单系统保存类中, 同时 HTML 代码将会自动生成。

```

<menu name="studentRecords">
  <menuItem name="viewGrades" text="View Grades"
    image="null" url="/app/viewGrades.do"
    bgrdColor="green" visible="y" enabled="y"
    selected="n" container="studentRecords"
    child="null" width="76" height="20"
    align="center" valign="top"
    linkClass="menuLinkColor"
    selectedLinkClass="menuActiveLinkColor">
  </menuItem>
  ...//other menu items
</menu>

```

Figure 4. menu system configuration

3.3 基于 WebGen 的方法

通过我们的 WebGen, 将会快速成型的开发一项 web 应用。设计者和开发者首先根据菜单系统将一个 web 应用分成一些 web 页面。每一个网页包含的数据模型都是一定的, 并且包含在数据文件中; 而生成标准和配置文件相一致。同时, 分级选项系统作为记录定义在菜单系统的具体文件中。然后 WebGen 将会生成 web 应用的原形。这个原形能够被引用和调整以适应当前的具体需要。从这点来说, 原形也被称之为实体模型。在这篇论文中, 我们将会交替的使用原形和实体模型这两个词。任何对于现在的数据模型和导航模型的改变都会集成到原形中经由 WebGen 重新运行。原形在取得客户同意后, 可以转让给开发商嵌入特定应用逻辑。在任何时候, WebGen 都能被应用到目前的原形系统中以包含以后对于数据模型和导航模型的改变。

4、实验

一个应用我们的方法的项目就是设计和开发一个 e-University 系统, 它是

一个学生, 教职员, 职员和管理员用来实现管理, 学生记录, 注册和金融服务的 web 应用。

当整个应用被提交后, 代码被用来检查分析。我们发现只有仅仅 9.62% 的代码是由开发人员写的, 而 90.38% 的代码都是由系统自动生成的。因此, 既然超过 90% 的代码都是由系统自动生成, 代码的质量将会得到保证, 开发的过程也将非常短, 同时相关的测试也会缩短。事实上, 当初这个项目是打算交给四个全职的开发者在四个月内完成的。当采用 WebGen 后, 这个项目仅由一名全职的开发者 (本文的第一作者) 和一名兼职的 JSP 设计和开发者就完成了。我们对于这个系统开始的实验表明 WebGen 能够提高开发速度的同时增强应用结果的可靠性。

我们的实验也有一些问题。使用我们的方法的最主要的优点是我们的 WebGen 能够自动的生成包括前端和后端的代码。然而, 我们要求前端的 JSP 页面和关系到系统后端的商业对象不能太复杂。因此, 需要一个高级的设计师来为大型的复杂系统分解和设计对象模型。总之, 我们认为更多的 WebGen 的内在功能还需要被提供以提高我们的方法支持大型的复杂的 web 应用的能力。

5、评估, 贡献和将来的工作

在本论文我们介绍并讨论了基于模板/汇总数据 (template/meta-data) 的部分代码生成系统支持 Web 应用的开发。它将无缝的集成 XML 和 J2EE 系列技术, 该框架将最大限度的挖掘基本工具技术的能力。此外, 完全独立的数据模型, 导航模型和显示模型反映了更多的一般概念过程, 这个过程将把技巧和方法从它们的基本技术选择中分离出来。在面对一个业务系统时, 在合成码和以后必须通过其他方式添加的代码之间去耦合而不是通过发生器来处理不断增长的变化和模型的改变, 因此将更进一步增强生成的应用的可扩充性, 可维护性和可重用性。

我们知道我们的方法需要在设计和开发阶段有严格的纪律性。然而, 在我们的实验中, 我们获得的好处大大超过了这个副作用。我们的方法引导开发者快速构建一个 web 应用的原形; 分析阶段和开发阶段清楚地分隔开。系统实体模型的获得使得顾客能够在开发的早期阶段就能参与到其中。WebGen 只要在开发者很少干预的情况下就能建立起一个可运行的实体模型系统。而且, 实体模型系统能够根据用户的反馈容易的进行修改。我们未来的工作将会包括一个交互的开发环境 (IDE) 来支持概念模型的编辑。现在开发者需要建立数据模型和配置作为一系列 XML 文件。IDE 能够自动的支持定义和维持这些文件, 同时提供一个图形化的编辑器来协助设计和维持概念模型。

致谢

我们深深地感谢评价者给我们的有价值的反馈。

6、参考文献

- [1] <http://ant.apache.org>.

- [2] Baresi, L., Garzotto, F., & Paolini, P. From web sites to web applications: new issues for conceptual modeling. Proceedings of WWW Conceptual Modeling Conference, (Salt Lake City, 2000).
- [3] Bochicchio, M., and Palano, R. Prototyping web applications. in Proceedings of the 2000 ACM Symposium on Applied Computing (SAC) (Como, Italy, 2000), 978–983.
- [4] Booch, G., Jacobson, I., and Rumbaugh, J. The Unified Modeling Language User Guide. Addison-Wesley, Reading MA, 1998.
- [5] Ceri, S. Fraternali, P., and Bongio, A. Web Modeling Language (WebML) : A Modeling Language for Designing Web Sites. Proceedings of 9th International World Wide Web Conference, (Amsterdam, Netherlands, May 2000), 15–19.
- [6]
http://www.kapitec.com/Produits/CodeCharge/en/codecharge_std.htm.
- [7] Conallen, J. Modeling web application architectures with UML. Communications of the ACM, 42, 10, (October 1999), 63–70.
- [8] <http://www.beust.com/cedric/ejbgen/#introduction>.
- [9] Fraternali, P., and Paolini, P. Model-driven development of web applications: the AutoWeb system. ACM Transactions on Information Systems (TOIS), 18, 4, (October 2000), 323–382.
- [10] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. Design Patterns. Addison Wesley, 1994.
- [11] Gaedke, M., and Rehse, J. Supporting compositional reuse in component-based web engineering. in Proceedings of the 2000 ACM Symposium on Applied Computing (SAC) (Como, Italy, 2000), 927–933.
- [12] Garzotto, F., Paolini, P., and Schwabe, D. HDM - a model based approach to hypermedia application design. ACM Transactions on Information Systems (TOIS) 11, 1, (January 1993), 1–26.
- [13] Grundy, J., et al. Extending a persistent object framework to enhance enterprise application server performance. In Proceedings of the 13th Australasian conference on database technologies, (Melbourne, Victoria, Australia, 2002), 57–64.
- [14] <http://www.imatix.com/html/gslgen/index.htm>.
- [15] Monson-Haefel, R. Enterprise JavaBeans. O’ Reilly & Associates,

- Inc. ISBN: 0-596-00226-2. 3rd edition, 2001.
- [16] <http://java.sun.com/j2ee>.
- [17] <http://www.jeewiz.co.uk/index.html>.
- [18] <http://www.jostraca.org/links.html>.
- [19] <http://www.koolfrog.com>.
- [20] Lei, Y., Motta, E., and Domingue, J. IIPS: An intelligent information presentation system. in Proceedings of the 7th International Conference on Intelligent User Interfaces, (San Francisco CA, 2002), 200–201.
- [21] McLaughlin, B., and Loukides, M. Java and XML. O'Reilly Java Tools, 2001.
- [22] Schwabe D., and Rossi G. An Object Oriented Approach to Web-Based Application Design, Theory and Practice of Object Systems, 4, 4, J. Wiley, 1998.
- [23] http://jakarta.apache.org/Struts_
- [24] <http://java.sun.com/docs/books/tutorial/uiswing/mini/index.html>.
- [25]
http://industry.java.sun.com/solutions/products/by_product/0,2348,ejb-5791-28,00.html.
- [26] Zhang, J., and Buy, U. A framework for the efficient production of web applications. Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC), (Kemer-Antalya, Turkey, June 30–July 3, 2003), 419–42.

致 谢

大学四年的生活很快就要过去了，四年中，我学到了很多知识，提高了自己在各方面的能力，这些知识和能力的获得和老师的辛勤栽培、同学的关心帮助是分不开的，在此，谨向他们一一表示感谢：

首先，我要感谢我的导师李罡老师，我很早就被他那诙谐幽默的讲课风格，诲人不倦的教学态度所吸引，在进行毕业设计和论文写作期间，又幸得李罡老师担任我的导师，李罡老师以其严谨的工作作风，孜孜不倦的治学要求和关怀备至、亲切自然的态度给予了我很大的帮助，使我的论文能够按时地并保质保量的完成，可以说李罡老师是我遇到的最好的老师之一。

其次，要感谢在我四年大学学习生活中给予我教导和帮助的天津大学的各位老师，正是这些老师在四年中传授给我的知识和能力保证了我的论文的顺利完成，而这些知识和能力也必将成为我以后学习和工作的基础和后盾。

同时，还需要感谢我的同学们在这四年中给我的关心和帮助，特别是在最后论文写作期间给了我很大的帮助。

最后，要感谢我远方的父母和家人，感谢他们在学业和生活上给我的理解、支持和鼓励，使我能够安心的完成学业。