

## 摘 要

随着计算机技术、多媒体技术、通讯技术在教育领域的不断推广，虚拟实验室在教育领域中的应用越来越广泛。虚拟实验室除了可以有效辅助高校开展科研工作，在实验教学方面也有显著的作用。本文针对现有虚拟实验室平台设计的不足，就虚拟实验室平台设计的关键技术进行了深入的研究，提出了一套可行的虚拟实验室平台构建方案，并根据方案设计，在平台中构建了一个虚拟实验室实例。

本文首先介绍了课题的研究背景、国内外研究现状；随后，本文分别从 Web 服务构建技术、单点登录模型、虚拟实验构建技术等方面对虚拟实验室平台设计中涉及的相关技术做了分析探讨；本文第三章在分析虚拟实验室平台功能需求的基础上，重点针对虚拟实验室平台通用服务和统一身份认证对虚拟实验室平台的体系结构以及接口做了详细的设计，并对平台涉及的相关数据库做了相应的设计；本文的第四章集中介绍了虚拟实验室平台功能开发的细节，应用 Web 服务技术实现了部分虚拟实验室通用服务，详细介绍了基于耶鲁大学 CAS 认证模型的统一身份认证模块在虚拟实验室平台服务器端的部署情况；为了验证本文对虚拟实验室平台设计的可行性，本文在第五章构建了基于大学计算机基础课程的计算机硬件组成虚拟实验室，分别从调用虚拟实验室平台通用服务、统一身份认证客户端部署、虚拟实验的仿真设计等方面加以实现。

**关键字** 虚拟实验室；通用服务；统一身份认证

## Abstract

Along with the constant popularization practices of computer, multimedia and telecommunication in the field of education, virtual laboratory has been widely applied. Virtual laboratory can not only be a useful tool to support the research work of universities, but also plays a significant role in experimental teaching. However, deficiencies still exist in the progress of designing and developing this new teaching method. The purpose of this thesis was to make intensive study on key technologies in designing virtual laboratory platform, demonstrate a feasible proposal for virtual laboratory platform and implement a virtual laboratory on the platform.

This thesis firstly introduced the background and research status both domestic and abroad. Following in this thesis, web services, single sign-on, technologies in implementing virtual experiments were referred. The third chapter analyzed the functional requirement of virtual laboratories, on top of that, common services and CAS (central authentication service) were explicitly designed according to the infrastructure of virtual laboratory platform, primary interfaces and database design were included as well. Related feature implementations were involved in the forth part, concerning details of utilizing web services and Yale CAS to configure the virtual laboratory platform. In order to validate the feasibility of the proposed platform design for virtual laboratory, a component of computer hardware based on fundamental of university computer was designed, which was mainly implemented by invoking common services, deploying CAS client, and fixing a well designed virtual experiment.

**Key Words** Virtual laboratory, Common services, Central authentication service

---

# 西南交通大学

## 学位论文版权使用授权书

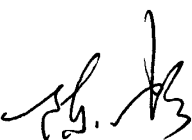
本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权西南交通大学可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复印手段保存和汇编本学位论文。

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“√”)

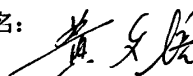
学位论文作者签名：



日期：

2010.5

指导老师签名：



日期：

2010.5

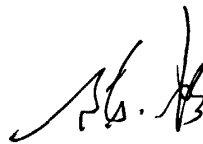
## 西南交通大学硕士学位论文主要工作（贡献）声明

本人在学位论文中所做的主要工作或贡献如下：

1. 在虚拟实验室平台通用功能的设计方面，本文提出了应用 Web 服务技术构建虚拟实验室通用服务的构想，在此基础上，实现了部分虚拟实验室平台通用功能。
2. 为了提升用户体验，方便用户在虚拟实验室平台中快速、有效地访问实验室资源，本文设计了基于耶鲁大学单点登录模型的虚拟实验室平台统一身份认证模块，并分别在平台服务器和客户端进行了部署。
3. 构建了基于平台的虚拟实验室，验证了验证通用服务、统一身份认证服务在虚拟实验室平台中应用的可行性，在此基础上，在虚拟实验仿真部分，本文根据大学计算机基础课程，设计实现了一个虚拟实验的仿真实例。

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说明。本人完全了解违反上述声明所引起的一切法律责任将由本人承担。

学位论文作者签名：



日期：

2010.5

## 第一章 绪论

### 1.1 研究背景

信息技术应用 in 提高信息传递速度和质量的同时，正快速广泛地渗透着人类社会的各个领域。在教育领域，信息技术已经引起了从教育观念、教育模式到教育过程的巨大变革。目前，伴随信息技术发展起来的 E-Learning<sup>[1]</sup>在教育领域中的应用日益广泛，特别是在 IT 学位课程的设置中，E-Learning 的支撑作用显得更加显著。<sup>[2]</sup>

J. W. Rickel 通过研究发现，单纯通过听觉获取知识，学生能获取总知识量的 25%；结合视觉与听觉，知识获取量可以达到 45%；而如果使用“在实践中学习”的方法，知识的获取量则高达 70%。<sup>[3]</sup>在现代远程教育中，理论课的许多课程需要借助各种多媒体实现基于网络的教学，而实验的出发点是学生亲自动手实践，因此相对于远程教育的理论教学模式而言，实验教学是近距离性的，这样一来，在远程教育中就出现了实验教学与远程教学模式不适应的状况。此外，目前教育网站中真正的教学内容相对匮乏，并且静态教程是主要的教学资源，动态教学内容仅占极小一部分，这已成为制约现代远程教育质量的一个重要因素<sup>[4]</sup>。

虚拟实验室 (Virtual Laboratory) 的概念最早在 1989 年由弗吉尼亚大学的 William Wulf 教授提出，用来描述一个计算机网络化的虚拟实验室环境<sup>[5]</sup>，是一个“无墙的研究中心”<sup>[6]</sup>。联合国教科文组织于 1999 年 8 月在美国依阿华州立大学召开的会议确定了“虚拟科学研究中心”和“虚拟研究实验室”科研和教学模式的构想，此次会议对“虚拟实验室”的定义如下：为了实现远程协作、实验研究或其他创新活动，通过分布式信息通讯技术产生并发布结果的电子工作室。<sup>[7]</sup>借助于计算机技术、多媒体技术、虚拟现实技术、通讯技术和数字化信息等技术构建而成的虚拟实验室，是一种基于计算机虚拟原型系统的全新科学研究与工程设计方法，是除理论与实物实验之外的第三种研究设计手段和形式<sup>[8]</sup>。虚拟实验室的开发利用不仅可以辅助高校有效开展科研工作，在实验教学方面也有显著的作用<sup>[9]</sup>，一个设备良好、管理先进的虚拟实验室平台是学习者快速、深刻地获取各种知识和技能的重要场所，对于实验教学改革具有重要的应用价值，在提高远程教育的教学质量方面 also 具有重要意义。

### 1.2 国内外发展现状

虚拟实验室的开发初衷是为了方便科研人员使用远程仪器设备，共享数据资源，达到远程进行科研工作的目的。虚拟实验室实质上是一个分布式计算机系统<sup>[10-11]</sup>，在

该系统中配备有具有遥控能力的网络化研究设备和数据采集平台, 以及支持协作活动的各种工具。<sup>[12]</sup>

在虚拟实验室概念提出至今的近 20 年来, 因其诱人的应用前景, 各国均在大力开发, 并取得了一些进展。目前, 虚拟实验室的建设在发达国家比较普及, 特别是在教育领域, 为数不少的一些大学均组建了与学科领域相关的远程虚拟实验室, 这些网络实验室已广泛地应用于教学中, 实现了教学方法的改革和创新, 例如, 美国卡耐基—梅隆大学将计算机控制的示波器、函数发生器等设备连接到网络, 实现了学生远程学习和操作; 密歇根大学的 Vicher 系统<sup>[13]</sup>将虚拟现实技术应用在化学工程教育的领域, 实现了催化剂钝化和非恒温反应等虚拟化学实验用于化学工程教育和设计; 加拿大 LICEF 研究中心则设计了一套通用的远程交互式虚拟电子工程实验软件<sup>[14 15 16]</sup>, 简化了电子工程虚拟实验在配置和设计方面的工作流程; 此外, 意大利帕瓦多大学建立的远程虚拟教育实验室、新加坡国立大学的远程示波器实验和压力容器实验<sup>[60 61]</sup>, 均被引入到课程学习中, 极大地促进了教学方式多样性的变革。

相对来说, 目前国内进行虚拟实验室研究的高校比较少, 比较成熟的网络虚拟实验室有清华大学的工程力学虚拟实验室<sup>[62]</sup>, 华中科技大学的测试工程网<sup>[17]</sup>, 浙江大学的大学物理虚拟实验室<sup>[63]</sup>等。

常见的虚拟实验室平台根据实现技术的不同可大致分为两大类: 一类是以纯软件方式开发; 另一类是软硬件结合的方式:

1. 以纯软件方式开发的虚拟实验室平台大多是采用了 HTML、CGI、Java Applet、Java Servlet 等与 Web 相关的技术, 如: 约翰霍普金斯大学的虚拟工程/科学实验室<sup>[18]</sup>, 由 R.Subramanian 等设计开发的虚拟生物学实验<sup>[19]</sup>, 均是以 Java Applet 作为客户端, Java Bean 实现实验组件的构建, XML (eXtensible Markup Language, 扩展性标识语言) 用于描述应用和实验数据; 卡罗莱纳州立大学的 LAAP (Learn Anytime Anywhere Physics)<sup>[30 31 32]</sup>, 则是使用 Java 语言开发的基于 Web 的探索式虚拟物理实验室, 使用 Java Applet 构建虚拟实验相关的器材和设备; 清华大学的工程力学虚拟实验室、浙江大学的大学物理虚拟实验室则使用 Flash 技术实现虚拟实验的过程控制和实验仿真。
2. 以软硬件结合的方式所开发的虚拟实验室在客户端一般也是采用 HTML、Java Applet 等技术, 不同的是, 在服务器端需要使用真实的硬件设备作支撑以完成虚拟实验的仿真和实验数据的处理, 如: L.Fabrega 等设计构建的网络虚拟实验室<sup>[20]</sup>以一台 Ethernet Switch 和多台运行 Linux 操作系统的计算机作为仿真设备, 用户远程输入 Linux 网络操作命令, 服务器端在接到客户端数据后, 使用本地的仿真硬件设备进行数据处理, 处理完数据后再将仿真结果通过网络返回给用户。

在前文的介绍中可以看到,在虚拟实验室的设计和应用方面,虽然已经出现了一些成熟、商业化的产品,但是,在不同专业和应用领域,由于虚拟实验仿真的复杂性、实验表现形式的多样性以及运行模式的差异性等因素,导致了目前尚未出现完全通用的虚拟实验室运行平台。此外,由于还没有国际化的标准对虚拟实验室的设计和应用方面做统一的规范,容易使现在的虚拟实验室沦为 Internet 世界中的孤立小岛,大量的资源被“锁”在各个小岛的中央数据库中,虚拟实验室间的信息交互存在很大困难。因此,虽然虚拟实验室在全球范围内呈快速发展的趋势,但是也存在诸多需要解决的问题:<sup>[21 22]</sup>

### 1. 缺少通用平台方面的应用研究以及广泛的资源合作

虚拟实验室的核心和精髓之一是资源的共享,而目前虽然很多高校都建立起了自己的虚拟实验室,但共享的深度和广度还远远不够。大多数虚拟实验室是针对特定的应用领域进行研究和设计的,缺少对于具有一定通用性和集成性的虚拟实验平台的研究和应用。此外,虚拟实验室大多是自行建立各校、各系甚至是各课程的虚拟实验室系统,分散而孤立,处于一种低水平的共享(仅仅是虚拟实验室系统内部的集中共享),缺乏实验室内部成员之间资源的共享,学校之间虚拟实验室的共享和交流更是障碍重重。

### 2. 虚拟实验内容的表现力和过程的交互性有待进一步增强

目前比较常用的虚拟仿真实验大部分采用 Java 语言或者 Flash 进行开发,由于 Java 的图形化开发工具不多,功能上的不尽完善,在一定程度上影响了实验内容的表现能力,而且在解决用户对虚拟实验交互性需求等方面还存在不足<sup>[50 51]</sup>,而 Flash 技术虽然动画制作方面表现卓越,由于程序设计能力有限,交互功能和仿真能力略显不足。

### 3. 异构系统的大量存在使虚拟实验室系统变得越来越复杂

计算机网络是典型的异构系统,随着网络技术的不断进步,不同时期的网络系统要求共存,不同的机型、操作系统、编程语言,使开发分布式系统软件十分困难。一个虚拟实验室系统移植到另一个网络平台时,不兼容的现象普遍存在,导致的结果很可能是重新开发新的满足需求的虚拟实验室系统,这样一来,虚拟实验室平台的功能重复开发工作显得非常普遍。

综上所述,研究构建将虚拟实验管理、虚拟实验室管理、实验教学、网上协作等作为公共支撑功能集成于一个通用的虚拟实验室平台,降低开发网上虚拟实验室的难度和工作量,实现大范围的资源共享,可以更好地推动虚拟实验的开发和应用<sup>[23 24 25 26]</sup>。

## 1.3 论文的研究内容与组织结构

虚拟实验室平台的设计既要考虑满足虚拟实验应用方面的要求,又要考虑对已有的虚拟实验的集成和未来功能、规模扩展方面的要求。针对以上问题,论文提出了虚

拟实验通用平台的设计方案，并围绕以下几个方面开展研究工作：

1. 功能的通用性。功能的通用性指的是虚拟实验室平台可以为不同类型的虚拟实验时提供通用功能服务。为了使平台通用功能可以被基于不同平台、不同语言开发设计而成的虚拟实验室调用，通过细致的研究分析，本论文将与学科领域无关的用户管理、实验教务管理和实验教学管理设计成 Web 服务，以减少相似功能模块开发的重复性。
2. 用户身份认证的便捷性。考虑到用户在进入虚拟实验室之前需要进行身份验证以及虚拟实验室需要各自建立用户信息管理机制的现状，本文在虚拟实验室平台的设计中提出了基于耶鲁大学 CAS 模型的统一身份认证方案，以减少用户在虚拟实验室间穿梭的门户障碍，提升用户体验。
3. 虚拟实验交互的友好性。为了吸引学生的注意力，使学生通过虚拟实验掌握更多的知识，发展更多的能力，本论文对虚拟实验的构建技术进行了分析研究，使用 3DS MAX 以及 VRML 设计了虚拟实验的仿真实例。
4. 部署维护的方便性。为减少部署与维护时的工作量，本论文对于虚拟实验室平台的架构采用 B/S 架构体系。

整个论文的结构分为以下五个部分：

### 第一章：绪论

介绍本课题的研究背景，并对目前国内外的研究现状进行分析。

### 第二章：虚拟实验室平台设计相关技术

本章首先简要介绍 Web 服务模型，Web 服务开发框架 XFire，单点登录的模型分类、实现机制，虚拟实验构建技术，在本章的最后，介绍了本文对进行虚拟实验平台设计实现时的开发环境和工具。

### 第三章：虚拟实验室平台分析设计

本章首先提出了平台的总体设计原则，随后对虚拟实验室平台的功能需求做了分析，针对虚拟实验室通用功能模块、统一身份认证模块、虚拟实验仿真模块进行了详细的探讨，在此基础上，对虚拟实验室平台的体系结构进行了设计，重点对虚拟实验室平台通用服务和虚拟实验室平台统一身份认证的体系结构、业务流程、接口以及数据库设计作了详尽说明。

### 第四章：虚拟实验室平台的部署实现

本章首先分别详细阐述了虚拟实验室平台通用服务、虚拟实验室平台统一身份认证服务的实现细节，包括通用服务的接口设计及其实现、统一身份认证服务在服务器端的部署配置步骤。

### 第五章：基于虚拟实验室平台的实验室构建实例

为了验证虚拟实验室平台的实用价值，本章基于大学生计算机基础课程设计了基

于虚拟实验室平台的认识计算机主板实验，分别从统一身份认证客户端配置、虚拟实验室通用服务调用、虚拟实验设计开发、计算机硬件组成与其他实验室之间的交互等方面进行分析、设计、实现。

## 第二章 虚拟实验室平台设计相关技术

### 2.1 Web 服务

#### 2.1.1 Web 服务的起源和基本原理

在早期的软件开发中,为了方便系统维护和代码重用,常采用结构化的设计方式,这种方式往往将程序分成多个功能相对简单的模块,常用功能则以函数的形式实现,属性比较接近的函数则集合形成函数库,在调用的过程中,函数或函数库通常会受限于特定的程序开发语言或平台。随着面向对象程序设计思想的深入,原先的函数被封装成类,继承方法的使用增强了代码的重用与易维护性,但是,功能的调用仍受到程序开发语言和平台相关性的限制。为了实现基于不同开发语言程序间的相互调用,需要有更新的技术,组件技术则可以达到这个目的。从技术层面上来说,软件开发强调二进制级别的重用和集成以及程序语言的无关性、面向对象等,尽管 DCOM/COM+技术可以轻易跨越位于不同 Windows 平台中的应用系统,但是对于异构平台之间的交互仍然存在困难。

随着网络技术和 XML 应用的蓬勃发展,由于 XML 具有程序语言无关性和跨平台的特点,研究人员开始试图以 XML 描述在网络中进行数据的传输,从而解决了异构平台中的系统交互问题。随着这个理念的持续发展,Web 服务应运而生。Web 服务作为一种编程模型,是对象/组件技术在网络应用中的延伸,是通过标准的 Web 协议可编程访问的 Web 组件,以提高在可缩放的、松耦合的环境下交换信息的能力。W3C 组织对 Web 服务的定义是:Web 服务是一个用 URI(Uniform Resource Identifier,统一资源标识)标识的软件实体,其接口和绑定可以用 XML 协议定义、描述和发现,它支持通过 Internet 协议以及基于 XML 消息和松散耦合的方式与其它软件实体或服务直接通讯。<sup>[27]</sup>

在 Web 服务中,信息交换依靠 HTTP(Hyper Text Transfer Protocol,超文本传输协议)、XML、XSD(XML Schemas Definition,XML 结构定义)、SOAP(Simple Object Access Protocol,简单对象访问协议)和 WSDL(Web Service Description Language,Web 服务描述语言)等标准协议。<sup>[28]</sup>Web 服务将功能业务逻辑分解为服务,而这些服务是可以在网络上进行发布、发现和引用,Web 服务客户端则利用标准化的 XML 消息传递机制访问 Web 服务接口。Web 服务接口包含了与服务交互所需的细节,包括消息格式(详细描述操作的输入输出消息格式)、传输协议和位置,该接口隐藏了服务实现的具体细节,允许通过独立于软硬件平台、独立于开发语言的方式调用该服务。

Web 服务包括三种类型的角色：服务提供者 (Service Provider)，服务使用者 (Service Consumer)，服务注册中心 (Service UDDI)。图 2-1 展示了服务提供者、服务使用者、服务注册中心之间的交互过程。

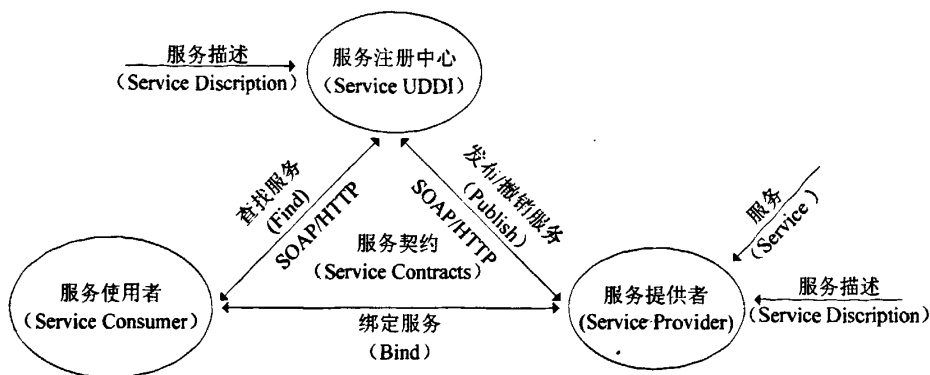


图 2-1 Web 服务模型<sup>[29]</sup>

三个角色间协作操作如下：

**发布/撤销服务：**为了使服务可被发现并使用，需要发布服务描述以使服务使用者发现和调用服务。

**发现服务：**服务请求者查询服务注册中心，查找满足需求的服务并定位服务地址。

**绑定服务：**服务请求者从服务注册中心查找到服务描述之后，服务使用者根据服务描述中的信息，调用服务提供者提供的服务。

### 2.1.2 XFire Web 服务构建框架

作为 Web 服务框架的后起之秀，XFire 从现有的框架中借鉴了许多优秀的理念，提供了各种绑定技术，支持多种传输协议，对 Web 服务体系中许多新的规范提供了支持。XFire 是一个免费的开源 SOAP 框架，它为 Web 服务的实现提供了一个实现环境，并且可以提供许多 Web 服务规范中的高级特征，这些特征在多数的商业或者开源工具都没有提供。简单来说，它具有以下一些特性<sup>[33]</sup>：

1. 支持重要的 Web 服务规范，如 SOAP、WSDL、WS-I Basic Profile、WS-Addressing、WS-Security 等；
2. 高性能的 SOAP 栈设计；
3. 可插拔的绑定，支持 POJO、XML Beans、JAXB1.1、JAXB2 以及 Castor；
4. 支持 JSR 181 规范，该规范通过 JDK 5.0 注解将 POJO 导出为 Web Service；
5. 支持多种传输协议，如 HTTP、JMS、XMPP、In-JVM 等；
6. XFire API 简洁明了，便于使用；
7. 支持 Spring、Pico、Plexus、Loom 等容器；
8. 支持 JBI (Java Business Integration, Java 业务整合) 规范；

- 9. 能够根据 WSDL 创建客户端和服务端端的存根代码；
- 10. 率先对 JAX-WS 提供了支持。

XFire 是完全基于流数据处理的框架，这意味着在工作进程中，XFire 不是将整个 SOAP 文档缓存在内存中，而是以管道的方式接收 SOAP 流数据，这种工作方式可以节省大量的内存占用。一般来说，在 SOAP 请求消息对 Web 服务发起真正调用之前，通常会经过传输、预转发、转发、策略实施、用户信息处理、预调用、服务调用等过程。

2.2 单点登录

单点登录（SSO，Single Sign On）是在相互信任的应用系统之间只需进行一次登录，并将这次主要的登录映射到其他应用中用于同一个用户的登录的机制<sup>[55-58]</sup>。单点登录的目的是在某个环境内，使用该单点认证（针对会话过程）作为访问其他应用程序、系统和网络的基础，从而消除了用户在一个会话中切换不同应用时的重复认证过程，对于用户来说，可以在各个系统间无缝的穿梭，同时减少密码记忆等安全信息，提升了用户体验。

2.2.1 单点登录的模型分类

单点登录模型根据认证信息存储地的不同可以分为三类：以服务器为中心的单点登录，以客户端为中心的单点登录，客户/服务器模式的单点登录<sup>[34]</sup>。这三种类型的单点登录模型在认证信息存储、认证过程等方面有所区别，如表 2-1 所示：

表 2-1 三种单点登录模型比较

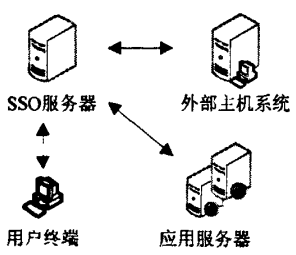
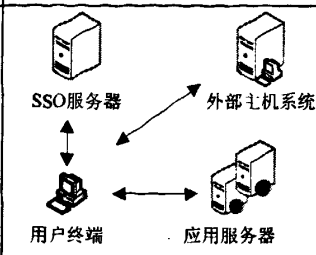
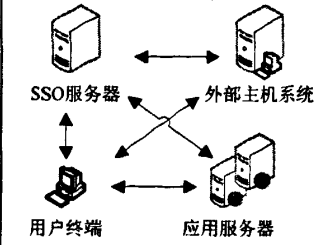
| 单点登录模型名称       | 认证信息存储地 | 模型图示                                                                                | 优缺点                                        |
|----------------|---------|-------------------------------------------------------------------------------------|--------------------------------------------|
| 以服务器为中心的单点登录模型 | SSO 服务器 |  | 用户合法性控制比较容易；服务器需要考虑负载均衡和备份；需要为端到端的集成进行定制开发 |

表 2-1 三种单点登录模型的比较（续）

| 单点登录模型名称            | 认证信息存储地             | 模型图示                                                                              | 优缺点                                |
|---------------------|---------------------|-----------------------------------------------------------------------------------|------------------------------------|
| 以客户端为中心的<br>单点登录模型  | 客户端                 |  | 不需要后端集成；容易造成非授权的访问；很难对用户进行细粒度的访问控制 |
| 客户/服务器模式的<br>单点登录模型 | 客户端<br>+<br>SSO 服务器 |  | 便于集成主机和网络设备；便于实现集中、细粒度的权限管理        |

2.2.2 单点登录的实现机制和实现技术

常见的单点登录实现机制包括<sup>[35]</sup>：

1. 基于经纪人（Broker-based）的实现，这种技术的特点是，有一个服务器集中进行身份认证和用户凭证信息的管理，经纪人则被用于电子身份的存取。中央数据库的使用减少了管理的代价，并为认证提供一个公共和独立的“第三方”，Kerberos、Sesame、IBM KryptoKnight 等就是这类实现机制的代表。
2. 基于代理人（Agent-based）的实现，在这种解决方案中，需要设计一个自动为不同的应用程序实现用户身份认证的代理程序。这个代理程序需要设计实现多个功能，如，使用口令表或加密密钥自动将认证的负担从用户端分离出去。代理人需要放在服务器端，在服务器的认证系统和客户端认证方法之间充当一个“翻译”，SSH（Secure Shell）就是此种机制的代表。
3. 基于令牌（Token-based）的实现，这是一种简单易用的方式，实现一个口令在多种应用当中使用，如 SecurID、WebID，FTP 和邮件服务器的登录认证等。
4. 基于网关（Gateway-based）的实现，在这种方案中，所有的响应服务局限在一个被网关隔离的受信任网段里，客户端通过网关进行认证后获得接受服务的授权。这种方案对企业中现有的网络环境要求比较严格，同时，要求已有的企业应用来适应单点登录系统，所以，这种方案的实际应用范围并不广。
5. 基于代理人和经纪人（Agent and Broker-based）的实现，该方案是 Agent-Based

和 Broker-Based 解决方案的结合,既具有前者的灵活性又能受益于后者中央式管理方案的优势。

而 SSO 在实现技术上则大体分为 Cookie<sup>[59]</sup>和 Session 两大类。Session 是一种服务器端机制,当客户端访问服务器时,服务器端为客户端创建一个唯一的 SessionID,以使在整个交互过程中始终保持状态,而交互的信息则可由应用自行指定。也就是说,用 Session 方式实现 SSO 时,不能在多个浏览器间实现单点登录,但可以跨域。<sup>[36]</sup>Cookie 则是一种客户端机制,它存储的内容主要包括:名字、值、过期时间、路径和域,路径和域合在一起就构成了 Cookie 的作用范围,因此 Cookie 方式可实现 SSO,但是域名必须相同。目前大部分 SSO 产品采用的是 Cookie 机制<sup>[37]</sup>,耶鲁大学的 CAS (Central Authentication Service) 也是如此。不同的是, CAS 可以很简单的实现跨域的 SSO,因为单点认证被控制在 CAS 服务器端,用户最有价值的 TGC (Ticket Granting Cookie) 只跟 CAS 服务器相关,而 CAS 服务器是唯一的,从而解决了 Cookie 不能跨域的问题。

目前已经有不少成熟的单点登录产品,主要的开源 SSO 软件包括:

1. WebAuth, 这是一种早期的 SSO 方案,支持 C++, Perl 等客户端。但是由于 WebAuth 使用了 Share Secret, 即 SSO 服务器端和客户端之间使用对称密钥 (Symmetric Key) 进行通信,因此在共享用户身份信息时存在一定的安全隐患。
2. OpenSSO, 它是基于 Sun Java System Access Manager<sup>[54]</sup>的一个开源的 SSO 项目,体系结构设计合理,功能比较强大,缺点是客户端支持不够广泛。
3. JOSSO, 这是另一个使用 Java 语言开发的单点登录产品, JOSSO 支持的客户端包括 Java, PHP 和 ASP。但是,由于 JOSSO 没有将后台认证与 SSO 分离,以及过分强调 JAAS, Axis 导致其在应用的广泛性上略有不足。此外, JOSSO 因为它使用了共享 Cookie 导致其不能支持跨域的单点登录。
4. CAS, 这是耶鲁大学开发的单点登录产品, CAS 设计理念先进、体系结构合理、配置简单、客户端支持广泛、技术成熟等等,目前 CAS 支持 (某些在完善中)非常多的客户端开发语言,包括 Java、.Net、ISAPI、Php、Perl、uPortal、Acegi、Ruby、VBScript 等。
5. CoSign, 这是 Michigan 大学的一个 SSO 项目, CoSign 跟 CAS 比较相似,都是基于 Kerberos 方式的协议,一个最大的不同是 CoSign 的 SSO Server 是基于 CGI,对 C/C++ 的群体应该是一个不错的选择。CoSign 协议的 UseCase 跟 CAS 很相似, CoSign 的客户端虽然也支持 J2EE/Apache/MSAPI(IIS), 但它的 Server 端使用 C 来编写,影响了在 Java 群体中的使用。

### 2.2.3 Yale CAS 模型介绍

Yale CAS 认证模型是耶鲁大学开发的一个比较典型的以服务器为中心的单点登录

模型，认证服务器集中存储用户的认证信息。该模型能为多个应用提供单点登录基础架构，同时可以为非 Web 应用但拥有 Web 前端的功能服务提供单点登录功能。在适用性上，由于它支持多种客户端，可以方便的在不同平台中进行部署。该模型的设计思路是，将多个 Web 应用系统的用户认证信息存储在一个身份认证服务器上，且该信息只用于验证用户身份的合法性，也就是说，用户凭证信息不会在应用系统间传递，由服务器为每个 Web 应用生成唯一不可篡改的票据，以保证 Web 应用的访问者身份是合法有效的。

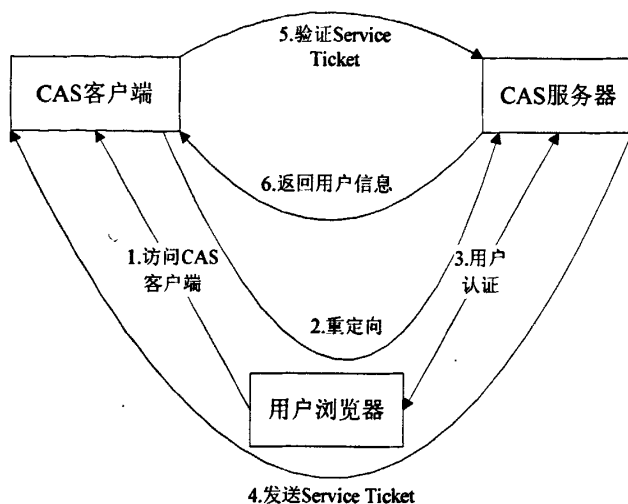


图 2-2 CAS 基础模式图<sup>[52]</sup>

图 2-2 为 CAS 的基础模式图。CAS 客户端首先过滤从用户浏览器传过来的每一个 Web 请求，在 Web 应用的受保护资源与用户之间构建起一道屏障。对于首次登录系统的情况，CAS 客户端会重定向用户请求到 CAS 服务器（图中的步骤 2），并传递该项服务信息（也就是要访问的目的资源地址），以便登录成功过后转回该地址。步骤 3 是用户认证过程，如果用户提供了正确的登录信息（Credentials），CAS 服务器会产生一个随机的相当长度、唯一、不可伪造的 ST (Service Ticket, 服务票据)，用户将使用这张票据访问指定服务应用。然后，缓存该 Ticket，并且重定向用户到 CAS 客户端（附带刚才产生的 ST），同时发送一个 TGC 给用户浏览器。下次当用户被其他应用重定向到 CAS 服务器时，CAS 服务器会主动获取到浏览器中 TGC，如果持有 TGC 且还没失效，那么进入步骤 4，CAS 服务器向该应用服务器发送 ST，达到了单点登录的效果；如果 TGC 失效，那么用户还是要重新认证（回到步骤 3）。最后，第 5 步和第 6 步是 CAS 客户端和 CAS 服务器之间完成一个对用户的身份核实，以确保 ST 的合法性。

## 2.3 虚拟实验构建技术

虚拟实验室平台应有具体的虚拟实验作为支撑，否则软件平台就是一个空架子，形

同虚设。在基于虚拟实验的开发过程中,虚拟环境的仿真、实验器材的造型建模、实验器材间的交互性操作是关键的一环。

虚拟实验中涉及到的设备、设施分成两类:一类是不需要交互操作的普通设施,另一类是可进行交互性操作的仪器设备。对于第一类实验器材可以利用建模工具进行模型构造,然后将其置于实验场景中即可。后一类仪器设备则应在构建逼真的外部模型的基础上,需要对虚拟仪器的各种开关、旋钮、接口等进行三维设计,此外,还需对各操作相关的数字或模拟量的显示进行编程处理。

在虚拟实验器材仿真部分,为了能够反映真实仪器设备的特性,通常采用 3DMAX 等建模工具对设备进行模型的构建,而为了增加沉浸感,常采用 VRML(Virtual Reality Modeling Language,虚拟现实建模语言)技术来构建三维场景,VRML 的动态交互能力使得三维场景可以根据用户的动作实时变化,使用户真正获得现场操作的感觉,同时,二维到三维的变化也使得用户看到的实验仪器和实验材料更加逼真。此外,Java3D、Flash 等技术也常被用于虚拟实验器材的模拟<sup>[38 39 40 41 42 43]</sup>。Flash 技术在动画制作方面表现卓越,具有运行效果出色、兼容性好的特点,但其程序设计能力有限,交互功能和仿真能力不够强,适宜做演示性实验;虽然 JAVA 技术现在支持三维场景的创建,但是使用 Java3D 创建虚拟场景并不容易,和 VRML 相比,它的开发周期较长,也没有 VRML 技术那么成熟。

### 2.3.1 虚拟现实建模语言 VRML

VRML 最初出现在 1994 年的瑞士日内瓦 W3 会议上,是一种描述因特网上交互式三维多媒体的标准文件格式<sup>[44]</sup>,是一项和多媒体通讯、因特网、虚拟现实等领域密切相关的技术,其基本目标是建立因特网上的交互式三维多媒体。VRML 在电子商务、教育、工程技术、建筑、娱乐、艺术等领域的广泛应用,使其成为目前构建虚拟现实应用系统的基础工具之一。

VRML 作为一种虚拟现实三维立体网络程序语言,可以在网络上创建逼真的三维场景,使虚拟世界的真实性和交互性得到更充分的体现<sup>[45]</sup>。其基本原理是用文本信息描述三维场景在互进行联网上传输,利用三维图形生成技术在本地上利用多传感交互技术以及高分辨显示技术,由 VRML 浏览器解释生成逼真三维虚拟场景,使用者通过使用特殊的头盔、数据手套传感设备,或利用键盘、鼠标等输入设备,进行实时交互,感知和操作虚拟世界中的各种对象,从而获得身临其境的感受和体会。

VRML 综合了现有三维软件的景象描述语言的优点,包含基本元素、顶点、线和面的定义;可以实现坐标变换的缩放、旋转和平移;拥有优化的数据结构;采用基于文件的运行机制,其程序不必经过任何编译、连接等处理;采用 ASCII 码格式的文本文件,在机器之间传送的是对象描述,而不是对象完整的图形化表示,因此可以大幅

减少在发布虚拟场景时占用的带宽。

但是, VRML 在构造三维场景需要录入较多规范的数据, 不能高效快捷地建立复杂物体的三维模型, 此时, 可以运用第三方的造型软件<sup>[46]</sup>, 现在很多三维软件都支持以 VRML 格式输出特定场景(如 UG, PRO/E, 3DS MAX 等都提供 VRML 的接口)。其中, 三维动画软件 3DS MAX 可以高效快捷地建立复杂物体的三维模型, 从而弥补 VRML 建立复杂的三维模型这方面的不足<sup>[47 48]</sup>, 但用 3DSMAX 制作的动画难于实现交互性及实时控制。把 VRML 与 3DSMAX 结合起来实现虚拟现实的动态交互可以达到事半功倍的效果。

### 2.3.2 建模工具 3DS Max

模型创建是开发虚拟实验的基础, 模型的好坏直接影响运行的效果和场景的沉浸度。3DS Max 提供多种建模技术, 开发者可以根据不同类型的虚拟实验仪器、设备或场景选择合适的建模技术来构建模型或设计场景。3DS MAX 是 Discreet 公司开发的(后被 Autodesk 公司合并)基于 PC 系统的三维动画渲染和制作软件。其前身是基于 DOS 操作系统的 3D Studio 系列软件, 在 Windows NT 出现以前, 工业级的 CG (Computer Graphics) 制作被 SGI 图形工作站所垄断。3DS Max 与 Windows NT 组合的出现极大地降低了 CG 制作的门槛。

以 Autodesk 3DS MAX2009 为例, 该工具除了具有实体造型、图像与三维动画制作于一体的强大功能外, 还集成了 VRML97 的功能, 包括内嵌、声音、时间感应器、细节级别、触动感应器、锚等 12 个 VRML 辅助工具。实体建模完成之后只要将文件导出为 VRML 的类型并进行保存就可以得到一个以 wr1 为后缀的文件, 在输出的 VRML 文件中可以支持 3DSMAX 的灯光、摄像机、动画等效果。

3DS MAX 广泛应用于广告、影视、工业设计、建筑设计、多媒体制作、游戏、辅助教学以及工程可视化等领域。应用 3D Max 构建模型或创建场景的关键技术对虚拟实验的开发具有重大意义。各种实验仪器、设备因其本身构造不同, 选择合适的建模方法构建模型不仅构建过程快速、方便, 而且构建完成的模型更加美观、逼真。适当的对模型和场景进行优化, 能生成更利于网络传输的文件。合理地运用这些技术, 能使 3D Max 制作的模型和场景更好地融入到各种虚拟现实制作工具中, 开发出符合要求的虚拟实验。

## 2.4 虚拟实验室平台开发环境、工具

虚拟实验室平台的开发主要分为以下几部分:

### 1. Web 服务的开发

开发环境: JDK 1.5.06

开发工具: MyEclipse 6.5

应用框架: XFire 1.2, Spring 1.2, Hibernate 3.0

服务器: Apache Tomcat 5.0

数据库: MS SQL SERVER 2000

操作系统: Windows XP

## 2. 统一认证模型的扩展与部署

统一认证模型服务器端版本: Yale Cas Server 3.3.3

统一认证模型客户端版本: Yale Cas Client 3.1.6

部署环境: JDK 1.5.06

服务器: Apache Tomcat 5.0

操作系统: Windows XP

数据库: MS SQL SERVER 2000

## 3. 虚拟实验的构建

开发工具: Autodesk 3DS MAX 2009, ParallelGraphics VrmIpad v2.1

应用插件: ParallelGraphics Cortona VRML Client 5.1

操作系统: Windows XP

## 本章小结

本章从介绍了分别从 Web 服务技术的起源以及基本原理, Web 服务构建框架 XFire 等方便细致阐述了在虚拟实验室平台设计中使用 Web 服务技术以及 XFire 框架的优越性; 结合现实应用中用户需要多次登录不同虚拟实验室平台的现状, 本章从介绍单点登录的模型分类以及实现机制入手, 细致阐述了 CAS 单点登录模型, 随后在虚拟实验室平台设计中引入了单点登录设计; 虚拟实验构建技术对虚拟实验室平台的设计而言是不可或缺的组成部分, 因此, 本章分别介绍了虚拟实验建模语言 VRML 和建模工具 3DS Max; 在本章的最后, 简单描述了在本论文的后续部分针对虚拟实验室平台设计在开发过程中涉及的开发环境和开发工具。

## 第三章 虚拟实验室平台分析设计

### 3.1 虚拟实验室平台的总体设计原则

虚拟实验室平台作为一个集虚拟实验教务、教学管理, 虚拟实验仿真为一体的功能强大的网上运行系统, 与传统的教学模式相互联系、相互补充。为了避免在各类虚拟实验室开发过程中通用功能的重复开发, 结合虚拟实验室之间的数据流通、保密数据的维护、用户体验的提升等因素, 本论文对虚拟实验室平台的总体设计原则如下:

1. 虚拟实验室平台采用统一的用户认证机制, 方便用户在虚拟实验室平台中无障碍地穿梭于各个虚拟实验室之间。统一用户认证只负责验证虚拟实验室平台用户身份的合法性, 系统功能的授权则由特定的虚拟实验室进行控制;
2. 与学科领域无关的虚拟实验室用户管理、实验教务管理、实验教学管理模块使用虚拟实验室平台提供的服务接口实现, 虚拟实验室平台对各个实验教务相关的信息进行统一管理, 虚拟实验室提供虚拟实验接口, 虚拟实验涉及的实验流程、数据存储及其相关接口的设计及实现, 则需要在具体的虚拟实验开发中应用相关技术实现;
3. 结合虚拟实验应用实际需要采用不同的虚拟仿真技术, 构建直观、交互性强的虚拟实验。

也就是说, 在整个虚拟实验室平台中, 用户身份认证中心是唯一的, 所有的用户登录都在统一的身份认证中心进行, 虚拟实验室应用不再处理用户的登录事宜。为了保证用户信息传递的安全性, 统一身份认证中心和所有的虚拟实验室应用间需要建立信任关系, 在统一身份认证中心对用户身份合法性的判断后, 用户信息将被传递给具体的虚拟实验室应用。而对于虚拟实验室平台所提取的通用功能, 在调用时, 需要由平台提供的通用功能服务接口完成服务的调用。

### 3.2 虚拟实验室平台的功能需求分析

在构建虚拟实验室平台之前, 首先需要分析平台的功能需求, 提取与应用领域无关的共性功能, 经过研究分析, 虚拟实验室的共性功能主要涉及用户管理、实验教务管理、实验教学管理三部分。此外, 根据上一节的平台设计原则, 虚拟实验室平台统一身份认证模块, 虚拟实验的仿真也是不可缺少的重要组成部分。

### 3.2.1 统一身份认证模块

在传统的虚拟实验室平台设计中,用户在进入实验室之前,需要先通过该实验室的身份认证,每一个虚拟实验室系统都需要建立独立的用户信息数据库以验证用户身份的合法性。对于同时需要使用多个虚拟实验室系统的用户,每进入一个虚拟实验室系统就需要登录,这无疑会耗费大量的时间。另一方面由于没有统一的身份认证系统设计,在一定程度上造成了现有成熟虚拟实验室较多,虚拟实验室之间资源共享程度不高的现状。

统一身份认证模块在用户向虚拟实验室应用发出请求时,将实际用户映射成一个电子身份,如果请求中的用户信息是合法的,则向用户浏览器发送一个 Cookie,同时将该身份标识安全、高效地传递到虚拟实验室系统的 Web 应用中,并为该虚拟实验室的 Web 应用生成唯一不可伪造的票据,这样一来,用户和 Web 应用之间就建立了通信关系,随后,Web 应用根据用户的请求进行相关信息处理。而此后如果用户再次向其他虚拟实验室 Web 应用发出请求时,统一身份认证模块会自动检测到用户浏览器中 Cookie 信息,由于该 Cookie 保存有用户凭证信息,平台将不再要求用户进行登录操作而直接与该虚拟实验室 Web 应用进行通信,而统一身份认证模块在监测到用户发出的请求时,仍然会为该 Web 应用生成唯一的应用服务票据,该过程对于用户而言是透明的。为了保证用户信息在统一身份认证服务器以及虚拟实验室系统之间传输的安全性,本模块与外界的信息交互均在 HTTPS 协议层之上,服务器与虚拟实验室之间采用双向身份认证模式。图 3-1 为统一身份认证模块在虚拟实验室平台中所处的网络位置。

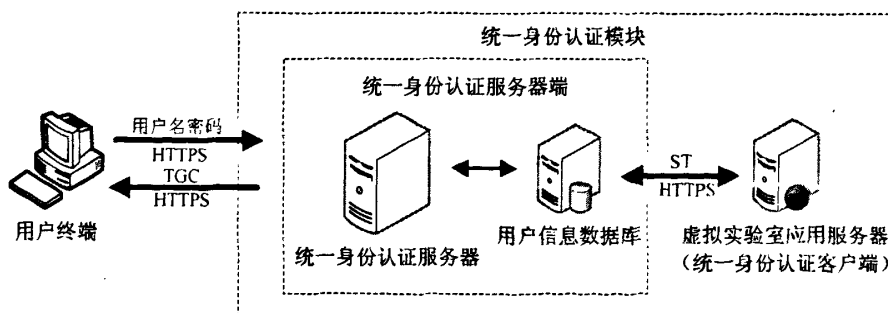


图 3-1 统一身份认证模块示意图

### 3.2.2 通用服务模块

本节将详细介绍虚拟实验室应该具备并且与学科应用领域无关的通用服务功能,根据上文分析,该模块根据管理内容的不同可以分为三个子模块:用户管理模块、实验教务管理模块、实验教学管理模块。

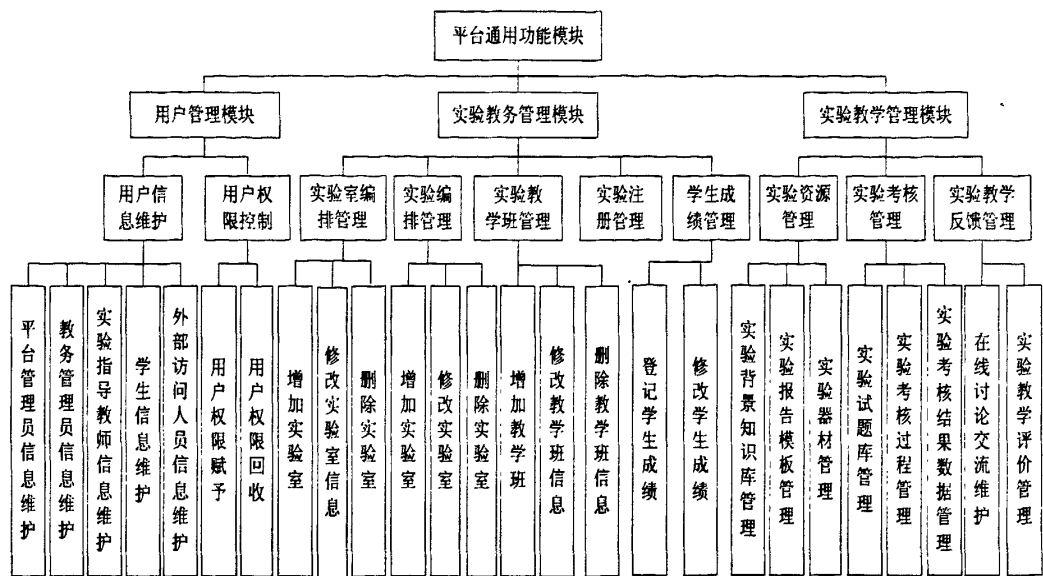


图 3-2 虚拟实验室平台通用服务模块功能结构图

图 3-2 为平台通用服务模块的功能结构图，下文将分别从用户管理、实验教务管理、实验教学管理对虚拟实验室平台的通用服务功能做详细介绍。

3.2.2.1 用户管理模块

在虚拟实验室平台中，用户角色可以分为平台管理员、教务管理员、实验指导教师、学生、外部访问人员等五类。用户基本信息维护主要进行平台用户的添加、审核、修改、删除等操作，用户数据的录入主要包括从数据文件中批量导入和单个录入两种形式，由于虚拟实验室平台的教务管理员用户、实验指导教师用户以及学生用户与普通高校教务网站用户具有很大的重叠性，用户数据管理模块在设计中需要提供教务系统用户数据库访问接口，以减少数据存储空间以及维护数据所带来的工作量。

除了需要对平台用户的用户名、密码、所在单位、联系方式等基本信息进行维护外，功能访问权限的控制也是一个非常重要的环节，在虚拟实验室平台中，用户权限控制在保障平台数据安全的同时，可以简化平台的资源管理。一般来说，平台管理员、教务管理员、实验指导教师、学生用户直接参与虚拟实验室应用，而外部访问人员在虚拟实验室平台中只能查看平台对外公布的信息，实验室的具体应用不对这类用户开放，但是，随着虚拟实验室开放进程的加快，外部访问人员在平台中的权限也将不断增多，不过在本节的权限授权控制部分，具体的实验室应用仍然不对该类用户开放。

依据 RBAC (Role Based Access Control, 基于角色访问控制机制)<sup>[64 65]</sup>的基本思想，对于不同的角色类型，在虚拟实验室平台中的操作权限有所区别，结合上文对于虚拟实验室平台通用功能的研究分析，基于角色访问的虚拟实验室平台权限控制机制如表

3-1 所示:

表 3-1 基于角色访问的虚拟实验室权限控制机制分析

| 权限描述<br>功能名称                      | 角色<br>名称 | 平台<br>管理员                            | 实验<br>教务管理员                                         | 实验<br>指导教师                        | 学生                                    | 外部<br>访问人员        |
|-----------------------------------|----------|--------------------------------------|-----------------------------------------------------|-----------------------------------|---------------------------------------|-------------------|
| 用户信息维护<br>(用户名、密码、<br>所属单位、联系方式等) |          | 具备平台<br>所有用户<br>信息维护<br>权限           | 具备教务员<br>个人信息、管<br>辖范围内的实<br>验指导教师、<br>学生信息维护<br>权限 | 具备个<br>人信息维<br>护权限                | 具备个<br>人信息<br>维护权<br>限                | 无该项<br>功能操作<br>权限 |
| 用户权限控制                            |          | 具备平台<br>所有角色<br>类型的权<br>限赋予、回<br>收权限 | 具备管辖范<br>围内的实验指<br>导教师、学生<br>的权限赋予、<br>回收           | 教学班<br>内学生权<br>限的赋予、回收            | 无该项<br>功能操<br>作权限                     | 无该项<br>功能操作<br>权限 |
| 实验室编排管理                           |          | 具备平台<br>所有实验<br>室的编排<br>管理权限         | 具备管辖范<br>围内的实验室<br>编排管理                             | 无该项<br>功能操作<br>权限                 | 无该项<br>功能操<br>作权限                     | 无该项<br>功能操作<br>权限 |
| 实验编排管理                            |          | 具备平台<br>所有实验<br>的编排管<br>理权限          | 具备管辖范<br>围内的实验编<br>排管理                              | 无该项<br>功能操作<br>权限                 | 无该项<br>功能操<br>作权限                     | 无该项<br>功能操作<br>权限 |
| 实验教学班管理                           |          | 具备平台<br>所有教学<br>班信息的<br>管理权限         | 具备管辖范<br>围内的教学班<br>信息管理                             | 具备对<br>所在实验<br>教学班的<br>信息管理<br>权限 | 无该项<br>功能操<br>作权限                     | 无该项<br>功能操作<br>权限 |
| 实验注册管理                            |          | 具备平台<br>所有学生<br>的实验注<br>册管理权<br>限    | 具备管辖范<br>围内的学生实<br>验注册管理                            | 具备对<br>所在教学<br>班的学生<br>注册管理<br>权限 | 仅参与<br>实验注<br>册环节，<br>不具备<br>管理权<br>限 | 无该项<br>功能操作<br>权限 |

表 3-1 虚拟实验室用户角色访问控制分析（续）

| 权限描述<br>功能名称 | 角色<br>名称 | 平台<br>管理员        | 实验<br>教务管理员      | 实验<br>指导教师         | 学生                       | 外部<br>访问人员 |
|--------------|----------|------------------|------------------|--------------------|--------------------------|------------|
| 学生成绩管理       |          | 具备平台所有学生的成绩管理权限  | 具备管辖范围内的学生成绩管理权限 | 具备对所在教学班学生的成绩管理权限  | 具备个人成绩查询、成绩申诉等权限         | 无该项功能操作权限  |
| 实验资源管理       |          | 具备平台所有实验资源的管理权限  | 具备管辖范围内的实验资源管理权限 | 具备所在教学班的实验资源管理权限   | 具备实验资源下载、查看的权限，不具备资源管理权限 | 无该项功能操作权限  |
| 实验考核管理       |          | 具备平台所有实验考核管理权限   | 具备管辖范围内的实验考核管理权限 | 具备所在教学班的实验考核管理权限   | 参与实验考核环节，不具备管理权限         | 无该项功能操作权限  |
| 实验教学反馈管理     |          | 具备平台所有实验教学反馈管理权限 | 具备管辖范围内的学生成绩管理权限 | 具备所在教学班的实验教学反馈管理权限 | 参与实验教学反馈环节，不具备管理权限       | 无该项功能操作权限  |

### 3.2.2.2 实验教务管理模块

本模块的主要任务是根据教学大纲进行实验室和实验的编排管理，并对学生的实验注册信息进行处理，此外，该模块还涉及实验教学班管理、学生实验成绩管理等。

实验室的编排管理的一般流程是，首先由教务管理员根据教学任务确定实验室的研究课题，然后结合实际情况向平台管理员提出虚拟实验室的编排计划，如为新学科开设新的实验室，改进部分已有的实验室信息以满足不断变化的教学需要，或者剔除少数不再满足教学要求的实验室等，平台管理员审核通过实验室编排计划则完成实验

室的编排工作。

实验的编排需依托某一特定的实验室研究课题进行规划，实验数量的增减由教学计划、学生的知识需求等因素决定。实验的编排管理工作由教务管理员负责，教务管理员分别从实验目的、实验内容等方面进行实验的编排，完成初步编排工作后教务管理员需向平台管理员提交编排计划以便平台管理员进行最后的审核，审核通过则完成实验的编排工作。

教务管理员根据实验室的课题性质和学生的培养计划可以制定初步的教学班开放计划，并以教学班为单位为其指定实验指导教师。学生的实验注册情况是决定实际教学班级数的重要因素，教务管理员在不影响实验教学质量的基础上根据学生注册数动态增减教学班级数量。

学生根据培养计划与实验课程安排情况进行注册操作，实验指导教师决定是否接收学生的注册申请，注册成功与否取决于实验教学班指导教师的工作量、本实验教学课题是否为该学生的优选课程等因素。实验注册管理允许教务管理员对学生的注册信息做弹性修改，如在条件允许情况下，接受学生提交调班申请等。

学生成绩是体现学生实验技能的重要形式，学生成绩的管理则是实验教务管理的核心内容之一。对于学生用户来说，除了可以通过该模块查看个人成绩，还能进行成绩申诉。实验指导教师则可以通过该模块登记、修改学生成绩，以及将成绩数据导出为平台支持格式的文件。教务管理员则可对学生成绩分布情况调整实验考核的难易度，也可以以此为依据修改此后的教学计划。

### 3.2.2.3 实验教学管理模块

本模块主要对实验教学密切相关的实验资源，实验考核以及实验教学反馈进行管理。

实验资源包括实验背景知识、实验器材、实验模板、实验报告等内容。实验教务管理员统一设置实验场景，包括实验目的的、实验模板的设计，实验器材的添加、删除，以及部分实验背景知识的上传。实验指导教师可以在线编写电子教案并根据需要发布、修改和删除电子教案，实验指导教师可以设置电子教案的访问权限，如是否允许外部访问人员查看、是否允许学生用户下载等。该模块支持学生用户在线查看、下载部分不受保护的实验资源。

在实验考核管理模块中，试题库由实验教务员和实验指导教师共同管理，实验指导教师可以上传新的考核试题，在新试题发布之前，实验教务员将进行试题审核工作，审核通过后新试题才进入实验试题库。实验指导教师在对学生进行考核时，负责试题内容的选定，并设定考试时间，允许教师通过某些手段查看学生考试进程。考核结束

后,教师根据学生作答情况在成绩管理模块中录入成绩。

实验教学反馈模块包括在线交流和教学评价两部分。在线交流允许教师可以组织学生参加答疑讨论,也可在此开展头脑风暴类的集体学习活动。学生通过教学反馈实现对实验教学内容、实验资源、实验指导教师教学情况的评价,实验指导教师和实验教务管理员负责对评价信息作回应;实验指导教师则可以针对学生的考核结果对学生提出新的学习要求。

### 3.2.3 虚拟实验仿真模块

该模块主要借助多媒体、仿真软件和虚拟现实等技术,实现在计算机、网络上营造可辅助、部分替代甚至全部替代传统实验各操作环节<sup>[49]</sup>,为用户呈现与真实的环境相仿的虚拟实验。虚拟实验的仿真需要从以下三个方面综合考虑:

1. 多感知性,所谓多感知性指的是除了一般计算机能带给用户视觉、听觉的效果之外,还应结合触觉感知、味觉感知、嗅觉感知等。理想的虚拟实验应该能满足人所具有的感知功能。
2. 沉浸感,又称临场感,它是指用户对于虚拟实验器材、虚拟环境所感受到的真实程度,理想的虚拟实验环境应该达到使用户难以分辨真假的程度。
3. 交互性,指的是用户在环境中对器材的可操作程度,以及从环境中得到反馈的实时性。

为了提高用户体验,虚拟实验环境的创建、虚拟实验器材的仿真以及实验操作的交互性和实验结果的仿真性是该模块的关键,而虚拟实验过程控制则是整个虚拟实验的灵魂。此外,由于虚拟实验仿真模块与实验教务管理模块在信息交互上存在一定的关联,因而该模块需要为平台通用服务模块提供访问接口,方便两个模块间的信息交互,该模块的功能软件位置如图 3-3 所示:

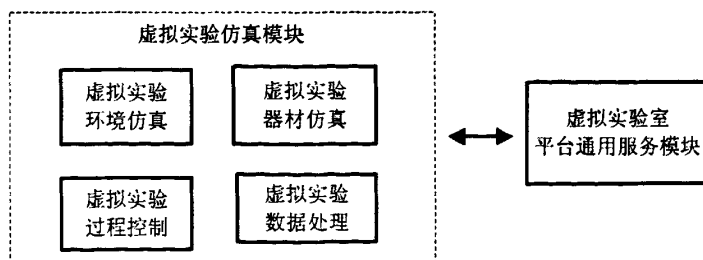


图 3-3 虚拟实验仿真模块软件位置图

3.3 虚拟实验室平台的体系结构及接口设计

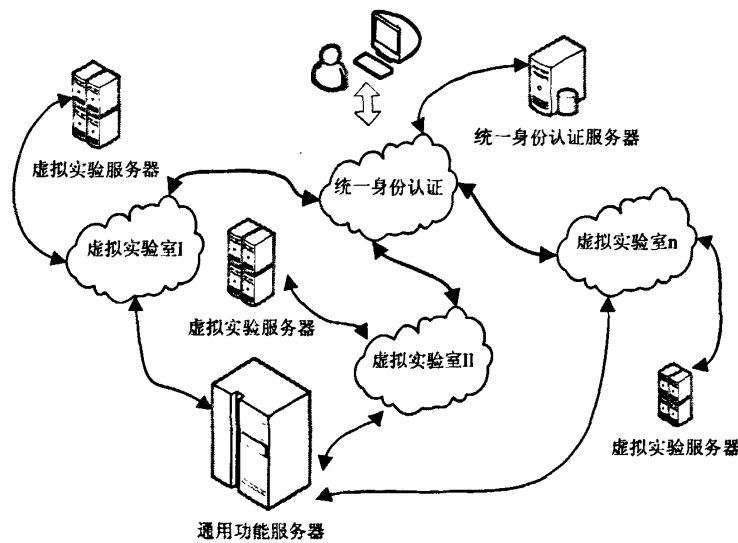


图 3-4 虚拟实验室平台的总体架构图

图 3-4 为虚拟实验室平台的总体架构图，本节将分别从统一身份认证和通用功能服务进行体系结构和接口的设计。

3.3.1 统一身份认证设计

统一身份认证在虚拟实验室平台中主要实现用户身份的合法性认证，用户通过身份认证后，即可访问所有与统一身份认证服务建立信任关系的应用系统。作为一个异构的应用系统，统一身份认证的设计需要支持多种客户端类型，而前文已经提及，Yale CAS 具有较好的适用性，因此，本文采用 Yale CAS 模型对平台进行统一身份认证设计。

由于统一身份认证存在一种门槛效应，黑客只需截获 CAS 服务器与数据库进行身份验证时的信息，或者 CAS 服务器发送到用户服务器端的 TGC，就能访问所有与该用户相关的所有应用系统。另一方面，由于 CAS 服务器应用 SSL 将 TGC 发送给终端用户，如果在数据传输发送方和接收方之间采取相对严格的双向身份认证方式，那么在数据传输过程中要截取 TGC 难度非常大。由于 CAS 模型中并没有采取任何措施保证用户身份信息的安全传输和认证，如果 CAS 服务器需要访问远程的用户数据库来验证用户信息（如在统一身份认证开发之前，个别 Web 应用已对用户数据库完成设计），用户信息极有可能被窃取，这样一来，CAS 服务器和用户数据库之间信息交互的安全性就显得尤为重要。

下文将分别从以下三个方面进行逐一介绍：平台统一身份认证应用流程，基于双

向认证的服务器、客户端结构设计，服务器端用户信息安全性设计。

### 3.3.1.1 统一身份认证应用流程设计

根据实际应用的需要，统一身份认证模块在虚拟实验室平台中的应用可以分为两个环节：

1. 控制虚拟实验室平台用户单点登录过程；
2. 实现虚拟实验室平台用户单点登出。

基于 Yale CAS 模型的单点登录在平台统一身份认证模块的应用流程如图 3-5 所示：

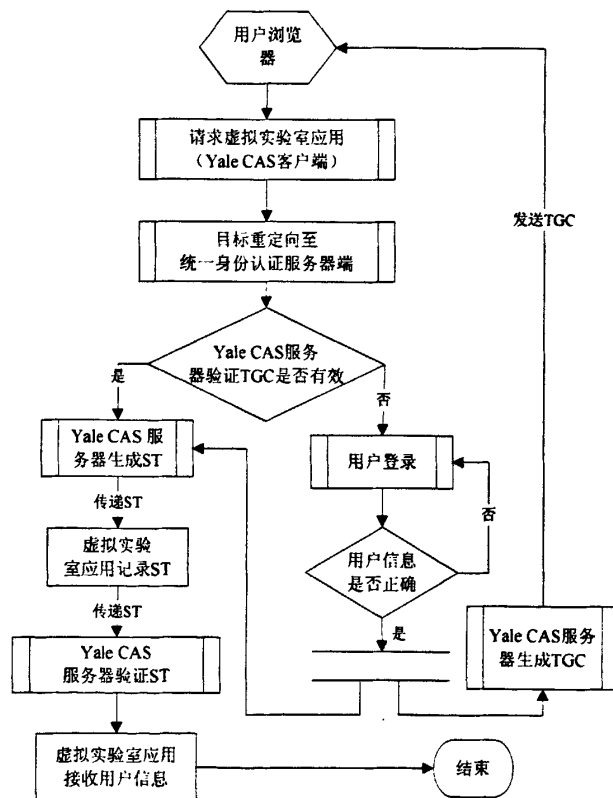


图 3-5 虚拟实验室平台统一身份认证过程示意图

1. 当用户第一次登录虚拟实验室平台时，浏览器将重定向至统一身份认证服务器，显示统一用户登录页面，提示用户输入用户名和密码等相关用户身份验证信息。
2. 用户键入用户名和密码后，用户的认证信息提交到 CAS 服务器。
3. 统一身份认证服务器到用户数据库或者 LDAP( Lightweight Directory Access Protocol)服务器中核对用户身份信息。
4. 如果用户信息验证成功，统一身份认证服务器将生成一个包含 TGC 的回执并将

其发送到用户浏览器,同时 CAS 服务器会生成一张服务票据 ST,以证明用户对该虚拟实验室应用具有使用权。

5. CAS 服务器将浏览器重定向到用户刚才请求的服务页面。应用系统将服务票据 ST 及其服务名一起提交到 CAS 服务器请求验证;
6. 如果 ST 验证通过,CAS 服务器就将用户信息传送给应用系统,应用系统同用户浏览器间从而建立了会话,可以使用该虚拟实验室应用。

在实际虚拟实验室平台应用中,比较常见的情况是,用户可能同时使用多个实验室,在传统的设计中,实验结束或者用户不再使用该实验室资源时,基于安全性的考虑往往需要进行注销操作,而逐一对各个实验室进行退出或注销的设计,可能导致用户在整个退出过程中,只注销了在某些实验室中的登录信息,而在其他实验室中,该用户仍处于登录状态,下一位使用计算机的用户将有可能篡改实验用户的信息,这样的设计可能会引发安全性方面的问题。鉴于以上问题的考虑,在虚拟实验室平台的设计中,退出实验室的方案使用单点登出技术,应用的模型是耶鲁大学的 CAS 模型,需要说明的是,该模型 JA-SIG CAS Server 3.1.2 版本之后才支持单点登出功能,基于 Yale CAS 的单点登出在虚拟实验室平台中的应用流程如图 3-6 所示:

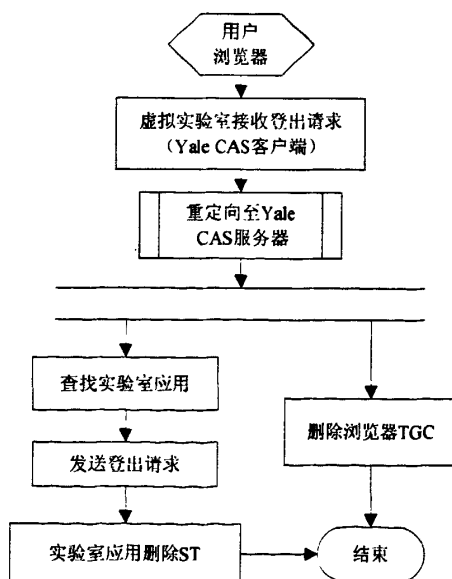


图 3-6 虚拟实验室平台单点登出流程

具体流程如下:

1. Web 浏览器向应用服务器(虚拟实验室)发出要求登出请求,应用服务器重定向到 CAS 服务器;
2. CAS 服务器接收请求,检测用户浏览器的 TGC,删除对应的 Session;
3. CAS 服务器找到所有通过该 TGC 登录的其他实验室应用服务器并发送回调请求;

4. 所有收到请求的应用服务器解析回调请求, 取得 SessionId, 根据 SessionId 取得 Session 后, 删除 Session, 单点登出过程结束。

### 3.3.1.2 客户机/服务器双向身份认证设计

由于统一身份认证服务器和客户端的信息交互在 SSL 层上进行, 而 SSL 连接分为握手和数据传输两个阶段, 一旦服务器和客户机成功完成握手过程, 数据信息将经过一系列加密算法保护后开始在服务器和客户端之间进行传输, 由此可见, 有效安全的服务器和客户机握手是保障数据信息传输的基石, 一旦有伪装的不法用户在该过程与服务器或客户端握手成功, 信息数据在传输的过程中就极有可能被窃听甚至篡改, 后果不堪设想。

一般来说, 在握手阶段, 服务器与客户机主要完成以下三个任务:

1. 就一组用于保护数据的算法达成一致;
2. 就一组选定算法选定加密密钥;
3. 进行身份认证。

SSL 认证包括双向认证模式与单向认证模式, 两者的主要区别在于握手阶段的身份认证部分, SSL 提供了以电子证书为基础的身份认证机制, 认证主要进行证书真实性验证和证书所有者的确认, 单向认证只对服务器进行身份验证, 而双向认证模式增加了对客户机的认证过程。客户机认证的思想是客户机使用其私钥对某些内容进行数字签名, 由于数字签名可以保障传输内容的机密性、完整性、不可抵赖性, 因而可以证明证书确实为该客户机所有。其设计的初衷是为了满足某些服务器的访问权限的要求, 服务器根据不同的客户机证书实现对用户身份的认证, 进而提供相应权限的服务。由于在 Yale CAS 模型中, 并没有对客户机对服务器的访问做限制, 这样的后果是, 黑客很容易伪装成客户机对服务器进行访问, 双向认证模式则可以使服务器在辨别客户机真实身份的同时, 有效减弱服务器遭受攻击的可能性。

统一身份认证服务器和客户机在握手之前, 需要各自生成数字证书文件, 证书生成的一般流程是:

1. 证书申请者产生密钥对;
2. 证书申请者将公共密钥及部分个人身份信息传送给身份认证中心(Certificate Authority, CA), 请求申请数字证书;
3. 认证中心核实信息发送方身份后, 为其颁发数字证书, 该证书内包含证书请求者的个人信息以及其公钥信息, 同时还附有认证中心的签名信息。

统一身份认证服务器和客户机在握手阶段的双向身份认证的过程如下:

1. 客户机向统一身份认证服务器发出问候信息请求建立连接, 该消息包含了随机

- 字节流, 客户机支持的 SSL 或 TLS 的最高版本号, 支持的加密方法;
2. 服务器收到问候消息后, 记录客户机随机数, 向客户机发送问候消息, 此消息包含与客户机提供的版本号以及加密算法后选择的证书版本号、加密算法、压缩算法, 会话编号, 服务器随机字节流, 该过程完成算法协商;
  3. 服务器向客户机发送包含服务器端数字证书的报文, 该报文包含了服务器端数字证书的公钥信息, 客户机将用该公钥加密实际的会话密钥;
  4. 服务器向客户机发送客户机数字证书请求, 该请求包括服务器可接受的客户机证书类型, 以及服务器所信任的 CA 列表;
  5. 服务器向客户机发送问候过程结束的报文;
  6. 客户机记录服务器随机数、会话编号, 验证服务器证书的合法性;
  7. 客户机向服务器发送合法的数字证书, 如果客户机没有满足条件的证书, 则向服务器发送无证书警告, 服务器收到警告后宣布本次握手失败。但是, 该警告仅仅起到警示作用, 服务器可以继续选择其他认证方式或者中断本次通信;
  8. 客户机提取服务器端数字证书的公钥, 然后根据服务器的公钥生成随机预制密码串, 并用服务器数字证书的公钥对该密码串进行加密;
  9. 客户机向服务器发送封装有上一步加密后的密钥交换报文, 其中密钥信息是关于双方在会话中使用的对称密钥算法的, 由于使用了服务器端的公钥进行加密, 可以有效防止第三方窃听;
  10. 客户机发送证书验证命令, 该命令包括经客户机密钥加密后双方握手的摘要信息;
  11. 服务器计算摘要并使用先前接收的客户机证书中的公钥检查摘要信息, 因为只有证书的私钥持有者才能对消息进行签名, 从而可以断定证书发送方的真实身份;
  12. 客户机发送修改密钥格式协议 (Change Cipher Spec) 报文, 表明客户机之后发送的数据都会新的加密算法算法、对称密钥进行加密, 并用该加密方法向服务器发送确认握手的完整性的结束消息, 该消息包含了之前整个连接过程所有握手消息的 MAC 值 (Message Authentication Code, 消息验证码);
  13. 服务器接收客户机发送的结束握手消息, 解密该消息, 比较解密得到的 MAC 值与自身计算得到的 MAC 值, 如果两值相同, 则向客户机发送修改密钥格式协议报文, 该报文表明服务器之后发送的数据都会被加密, 并使用新的加密算法算法、对称密钥向客户机发送确认握手的完整性的结束消息;
  14. 握手结束, 客户机与服务器开始应用层数据的加密传输。

### 3.3.1.3 用户信息的存储访问机制

由于虚拟实验室平台用户数据与高校教务系统用户数据具有一定的重叠性,因此,平台用户信息的访问机制可以分为以下两种:

1. 使用轻量级目录访问协议 LDAP, 由于该协议直接运行于 TCP/IP 等可靠传输协议之上,可以有效防止统一身份认证服务器在进行用户合法性验证过程中的信息窃听。采取 LDAP 作为统一身份认证的用户数据库的另一个优势在于,直接从 LDAP 服务器读数据时比从关系型数据库读取数据快一个数量级,可以处理大量用户并发访问。此外,目录树结构也便于对人员组织架构的管理,这种方式对于已建有高校教务系统用户数据库的情况非常适用,教务系统首先需要向身份认证中心提供唯一的应用标识(如应用的 URL),统一身份认证服务器根据应用标识为其注册一个唯一的应用认证接口,随后在 LDAP 服务器上初始化用户映射表,初始化的主要工作是导入教务系统中与认证相关的主要信息(用户名、密码),如图 3-7 所示。

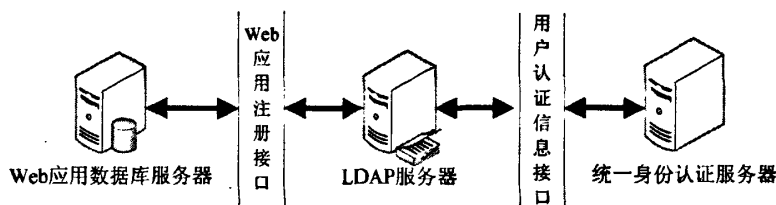


图 3-7 使用 LDAP 映射用户数据的结构图

2. 直接使用本地数据库存储用户信息,这种方法比较简单,统一身份认证服务器在接收到用户传递过来的信息后直接根据本地数据库中保存的用户信息进行合法性验证,这种存储机制适用于虚拟实验室平台独有用户数据的存储。在正常工作的情况下,虽然 SSL 能保护客户机和服务器端数据交换的安全性,而 SSL 通常需要与 HTTP 协议共同合作实现数据的传输,而 HTTP 作为无对象的连接,在用户数据传输过程中容易遭受中间人攻击 (man in the middle),攻击者对客户机冒充成服务器,对服务器冒充成客户机。一旦攻击者被统一身份认证服务器误认为是客户机用户就能直接访问数据库,此时用户的保密数据极有可能被窃取。因而,如果采取直接将用户信息保存在本地数据库中,那么为数据库设置加密机制是非常有必要的。本文第四章对虚拟实验室平台统一身份认证模块的实现将采用本方法。

### 3.3.2 通用服务设计

#### 3.3.2.1 通用服务软件体系结构设计

WEB 为虚拟实验室平台通用功能的实现提供了一种基础平台, 基于 WEB 的 B/S (浏览器/服务器) 计算模式是实现网上虚拟实验系统的一种基本模式<sup>[53]</sup>。所谓“B/S”是指 Browser/Server (浏览器/服务器), 即客户端是标准的浏览器, 服务器端为标准的 Web 服务器, 应用服务器响应浏览器的请求。相对于 C/S (Client/Server) 软件体系结构, 应用 B/S 结构的优点主要在于:

1. 无须在不同的客户端上安装不同的客户应用程序, 使安装过程更加简便, 简化了客户端。
2. B/S 应用程序模型是在服务器上部署和更新的, 消除了将应用程序的任何部分显式部署到客户计算机并加以管理的必要性, 具有易于安装, 方便部署与维护等特点, 可以解决许多与应用程序部署和维护相关联的问题。

基于以上两方面的考虑, 结合虚拟实验室平台通用功能的分析, 对于各个相对独立的实验室, 由虚拟实验平台提供基于 B/S 的通用功能 (如教务管理) 的服务接口。虚拟实验室平台通用功能模块部分的体系结构设计如图 3-8 所示。这些通用功能的业务逻辑和数据访问采用 Web 服务技术来实现, 可以减少重复的开发和重叠的模块, 增强代码的可重用性。根据上文对教务管理平台的功能分析, 可将该平台提供的各个功能分成多个 Web 服务: 实验室、实验、班级、教师、学生、实验相关资源、实验报告、实验成绩等。

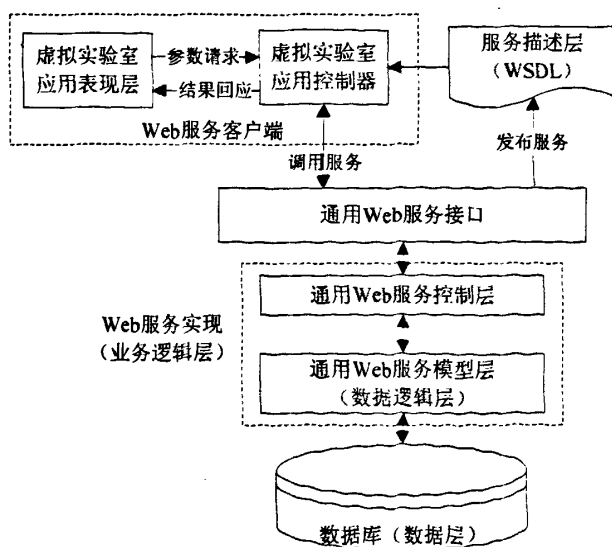


图 3-8 虚拟实验室平台通用功能模块的软件体系结构

3.3.2.2 通用服务调用流程

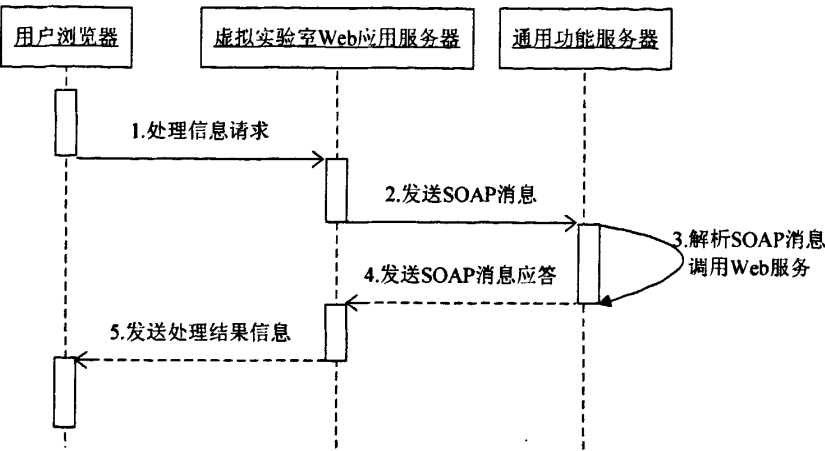


图 3-9 通用功能服务调用时序图

图 3-9 为通用功能服务调用时序图，调用流程如下：

- 1. 用户终端向虚拟实验室应用服务器发送的某个通用功能的应用请求；
- 2. 虚拟实验室应用服务器向通用功能服务器发送 SOAP 请求消息，该请求会被嵌入在一个 HTTP POST 格式的消息中；
- 3. 通用功能服务器根据服务端口以及服务描述的设置，解析收到的 SOAP 请求中的方法调用和参数格式；
- 4. 通用功能服务器调用 Web 服务，生成 SOAP 应答；
- 5. 虚拟实验室应用服务器接收到 SOAP 应答后，通过 HTTP 将处理结果呈现给用户。

3.3.2.3 通用服务接口设计

根据上文的分析设计，平台为虚拟实验室应用提取了通用功能服务，通用功能服务器提供了具体功能的业务逻辑以及功能实现。虚拟实验室应用在调用通用功能服务时，需要借助通用服务提供的接口，可以说，通用服务接口是虚拟实验室应用和通用功能服务器间实现服务请求与应答的桥梁。本节只对主要的四个主要接口进行设计：Istu, Itea, Ilab, Iclass。需要指出的是，接口的具体实现细节在本文第四章将详细介绍。

表 3-2 为虚拟实验室平台通用服务部分接口信息，以及每个接口提供的相关功能以及接口类：

表 3-2 通用功能主要接口及函数列表

| 接口名称   | 函数类型    | 函数名                           | 功能          |
|--------|---------|-------------------------------|-------------|
| Istu   | List    | getStudent()                  | 得到学生列表      |
|        | boolean | deleteStudentByID(Integer id) | 根据学号删除学生    |
|        | boolean | updateStudent(Student stu)    | 更新学生信息      |
|        | boolean | saveStudent(Student stu)      | 增加学生        |
|        | Student | getStudentByID(Integer id)    | 根据学号得到学生对象  |
| Itea   | List    | getTeacher()                  | 得到教师列表      |
|        | boolean | deleteTeacherByID(Integer id) | 删除教师        |
|        | boolean | updateTeacher(Teacher tea)    | 更新教师信息      |
|        | boolean | saveTeacher(Teacher tea)      | 增加教师        |
|        | Teacher | getTeacherByID(Integer id)    | 根据编号得到教师对象  |
| Ilab   | List    | getLab()                      | 得到实验室列表     |
|        | boolean | deleteLabByID(Integer id)     | 删除实验室       |
|        | boolean | updateLab(Lab lab)            | 更新实验室信息     |
|        | boolean | saveLab(Lab lab)              | 增加实验室       |
|        | Lab     | getLabByID(Integer id)        | 根据编号得到实验室对象 |
| Iclass | List    | getClasses()                  | 得到班级列表      |
|        | boolean | deleteClassesByID(Integer id) | 删除班级信息      |
|        | boolean | updateClasses(Classes cla)    | 更新班级信息      |
|        | boolean | saveClasses(Classes cla)      | 增加班级        |
|        | Class   | getClassesByID(Integer id)    | 根据编号得到班级对象  |

3.4 虚拟实验室平台的数据库设计

由于在虚拟实验室平台中，虚拟实验室涉及的数据信息与统一身份认证的用户合法性验证信息相对独立，因此，本文将上述数据分别存放于不同的数据库中，使用 MS SQL SERVER 2000 进行数据表的设计。

3.4.1 统一身份认证用户数据库表设计

由于统一身份认证采用基于用户名/口令的认证方式来确定用户身份，因此，对于该部分的数据库最为核心的数据库表是包含了用户名、密码的数据库表，该表只有两个字段：username，password，为了保证用户信息的唯一性，本论文对 user 表中 username

的构成做了以下规定: username 由国家代码、学校代码、学号组合的形式生成, 如 861061420034392, 86 为国家代码, 10614 为学校代码, 20034392 为学生学号。图 3-10 为统一身份认证用户凭证信息数据库的表结构。

| 列名       | 数据类型    | 长度 | 允许空                      |
|----------|---------|----|--------------------------|
| username | varchar | 50 |                          |
| password | varchar | 50 | <input type="checkbox"/> |

图 3-10 统一身份认证用户数据库表

### 3.4.2 虚拟实验室通用功能数据库表设计

由于统一身份认证只负责验证用户身份的合法性, 对于虚拟实验室功能使用权限由虚拟实验室本身进行分配, 因此, 有必要对虚拟实验室平台的用户分成若干组(平台管理员, 教务管理员, 实验指导教师, 学生, 外部访问人员), 相应地, 在虚拟实验室平台通用功能的数据库设计中, 需要对用户组进行分类设计。

虚拟实验室通用服务的数据库表关系如图 3-11 所示。需要指出的是, 虚拟实验室中的用户组中的用户名信息(图 3-11 中数据库表 admin、teacher、student 的 admin\_Id, tea\_Id, stu\_Id)与统一身份认证部分(数据库表 user 中的 username)保持一致。其中, student 表中的学号、teacher 表中的教工编号、classes 表中的班级编号分别作为表 enroll 表中相应字段的外键, teacher 表中的教工编号作为 class 表中教室编号的外键, lab 表中的实验室编号作为 classes 表中实验室编号的外键。

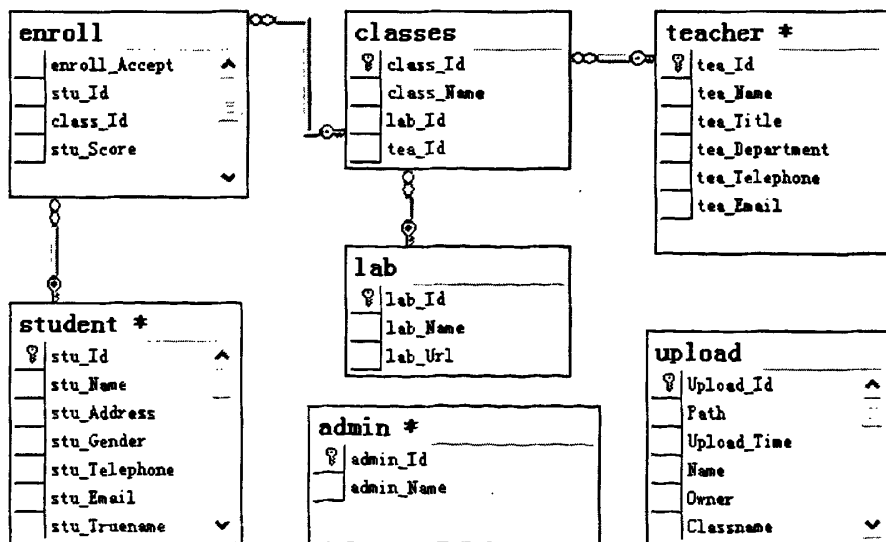


图 3-11 虚拟实验室平台通用功能模块部分数据库表关系图

## 本章小结

本章首先提出了平台构建的总体设计原则，随后从虚拟实验室平台的功能分析入手，对虚拟实验室平台的功能需求做了细致的分析；在功能分析的基础上，对虚拟实验室平台中尤为重要的统一身份认证、通用功能进行了架构设计，并对通用功能部分接口作了设计；根据前文的分析设计，为用户信息数据、虚拟实验室通用功能部分所需的部分数据库表进行了设计，并建立了对应的数据库表关系图。

## 第四章 虚拟实验室平台的部署实现

### 4.1 统一身份认证服务器的部署

虚拟实验室平台统一身份认证模块包括两个部分，认证服务器和客户端（虚拟实验室应用系统），本节只研究认证服务器端的实现，客户端部分的部署实现放到本文第五章进行研究。统一身份认证模块的服务器端完成以下几个方面的工作：

1. 验证客户端服务器的真实身份；
2. 验证用户信息的合法性；
3. 为 Web 应用生成 ST；
4. 生成 TGC 并将其传送到用户浏览器；
5. 注销用户登录信息。

下文重点介绍 Yale CAS 的统一身份认证服务器端用户合法性验证的实现以及服务器端 SSL 服务的开启。

#### 4.1.1 SSL 服务的开启

在前文对虚拟实验室平台的设计中，无论是统一身份认证还是通用功能的调用，信息交互都需要建立在 SSL 层之上，SSL 作为保护基于 WEB 事务的手段，需要在统一身份认证服务器与虚拟实验室服务器之间、虚拟实验室通用功能服务器端与通用功能服务客户端之间建立信任关系。SSL 指定了在应用程序协议（如 HTTP、Telnet、FTP）和 TCP/IP 之间提供数据安全性分层的机制，它采用公开密钥技术，为数据传输层提供数据加密、服务器认证、消息完整性以及可选的客户机认证。<sup>[53]</sup>

目前国内用户可选择购买的 SSL 证书有两种，一种是由美国的 Thawte、GeoTrust、Verisign 等公司颁发的，这类证书可以被浏览器认可；一种是国内各种认证中心颁发的，但不被浏览器认可，需要另外安装根证书。本论文使用自签名的证书作为客户端与服务器端安全通信的凭证，服务器为 Apache Tomcat 5.0，自签名证书使用 JDK 1.5.06/bin 中的 keytool.jar 生成，具体流程如下：

1. 生成服务器端密钥库

由于在开发过程中，计算机中已经安装了 JDK 1.5.06 并设置了环境变量，因此，可以直接运行 cmd 命令，并通过命令行进入 Tomcat 的安装路径，然后输入命令：  
keytool -genkey -alias tomcat -keyalg RSA -keypass changeit -storepass changeit -keystore server.keystore -validity 3600，其中，changeit 是密码，3600 是该证书的

有效期限为 3600 天，默认情况下证书的有效期为 3 个月。按回车键后根据提示进行相关内容输入，如图 4-1 所示：



图 4-1 密钥库的生成

完成此步骤后，会在 Tomcat 目录中生成 server.keystore 密钥库，该文件保存了服务器端的证书库。需要注意的是，由于在本虚拟实验室平台开发过程中，认证模型的服务器和客户端部署在同一台计算机上，因此，使用的 Common Name (CN，图中提示的“您的名字与姓氏是什么”)为“localhost”，也可以使用计算机名代替，CAS 不建议使用 IP 地址作为 CN，而一般情况下，由于服务器与客户端通常不会是同一台主机，因此，普遍的情况是将 CN 的值设成域名地址。

## 2. 生成服务器端的证书文件并导出该文件：

该部分的命令是：keytool -export -trustcacerts -alias tomcat -file server.cer -keystore server.keystore -storepass changeit，该命令将拷贝生成的 server.crt 文件至 %JAVA\_HOME%\jre\bin 目录下。图 4-2 显示了本步骤和下一步骤的操作信息。



图 4-2 数字证书的导出导入

### 3. 将服务器端生成的证书导入到 JRE 证书信任库中:

该部分的命令式: `keytool -import -trustcacerts -alias tomcat -file server.cer -keystore %JAVA_HOME%\jre/lib/security/cacerts`, 按回车输入密码后, 会显示该证书的信息。选择信任证书则完成了证书的导入。

### 4. 配置 CAS 服务器端 Tomcat 的 SSL 项:

将生成的 `server.keystore` 复制到 `%TOMCAT_HOME%\conf` 下。找到 `%TOMCAT_HOME%\conf\server.xml` 文件, 解开 SSL 的注释, 配置如下:

```
<Connector port="8443" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" disableUploadTimeout="true"
    acceptCount="100" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="server.keystore"
    keystorePass="changeit"/>
```

## 4.1.2 用户身份合法性验证的实现

在对用户信息进行合法性验证时, 由于用户凭证信息在客户端经过加密后通过 HTTP 协议传输送到认证服务器端, 密文只能用保存在认证服务器端的密钥才能解密得到正确的明文, 也就是说, 认证服务器在验证时, 首先对用户凭证信息进行解密, 然后将解密后的用户凭证与设置了保密机制的数据库数据进行核对, 如果用户名和口令与数据库的用户信息中相匹配, 该用户通过身份的合法性验证。

在耶鲁大学的 CAS 模型中, 服务器端默认使用 `SimpleTestUsernamePasswordAuthenticationHandler` 方法实现用户的身份认证, 只要用户名和密码相同就能通过认证, 但是这种身份验证在实际应用中是不能满足需求的, 而 CAS 服务器提供扩展认证的核心 `Authentication Handler` 接口, 该接口可以用于验证登录时的用户凭证信息。为了使用耶鲁大学已有的 CAS 服务器端, 本文应用 Tomcat 作为服务器, 并在该服务器中导入 Yale Cas Server 3.3.3 的 `cas.war`, 具体的目标位置是 Tomcat 目录下的 `webapps` 文件夹。Tomcat 服务器重新启动后, `cas.war` 将被自动编译成一个工程。根据本文 3.4.1 节对于用户身份信息数据库表的设计, 统一身份认证的数据库采用 MS SQL SERVER 2000, 为了使统一身份认证服务器支持 JDBC 认证方式, 需要为编译后的工程添加以下包文件: `spring-jdbc-2.5.6.jar`, `commons-dbcp-1.2.2.jar`, `commons-pool-1.5.4.jar`, `commons-collections-3.2.1.jar`, `msbase.jar`, `mssqlserver.jar`, `msutil.jar`。同时在 CAS 的服务器端的部署文件中添加 JDBC 认证, 需要说明的是, 用户凭证信息存放于数据库 `vlab` 中的数据表 `users` (前文已对该表的数据项信息作了说明) 中。以下是需要在统一身份

认证服务器的部署文件 `deployerConfigContext.xml` 添加的核心代码:

```
<bean class="org.jasig.cas.adaptors.jdbc.QueryDatabaseAuthenticationHandler">
  <property name="sql" value="select password from users where username=?"/>
  <property name="dataSource" ref="dataSource" /></bean>
  <property name="driverClassName">
    <value>com.microsoft.jdbc.sqlserver.SQLServerDriver</value>
  </property>
  <property name="url">
    <value>jdbc:microsoft:sqlserver://localhost:1433;databasename=vlab</value>
  </property>
  <property name="username"><value>username</value></property>
  <property name="password"><value>password</value></property>
</bean>
```

由于对用户信息数据库做了加密设置, 还需要在上述配置文件中添加加密设置的相关方法:

```
<bean id="myPasswordEncoder"
  class="org.jasig.cas.authentication.handler.DefaultPasswordEncoder"
  <constructor-arg value="MD5"/>
</bean>
```

## 4.2 通用服务的实现

本文 3.3.3 节已对虚拟实验室通用功能服务接口作了设计, 而在实现虚拟实验室通用服务时, 本论文使用 XFire 框架进行构建。为了实现虚拟实验室平台通用服务, 首先需要创建 Web Service 工程, 本文使用 MyEclipse 6.5 中新建了一个基于 XFire 1.2 的 Web Service 工程 vlabS, 结合上文对于通用功能接口的设计, 构建如下服务: `stu`, `tea`, `class`, `lab`, 分别对应接口: `Istu`, `Itea`, `Ilab`, `Iclass`。由于每个服务接口只提供类的抽象申明, 需要有与之对应的实现类进行具体的业务逻辑操作, 本文分别定义了 `stuImpl`, `teaImpl`, `labImpl`, `classImpl` 四个类具体实现接口服务, 上述实现类的业务层由 Spring 框架进行管理, 数据的持久化则通过 Hibernate 完成。

### 4.2.1 通用服务的构建

在新建服务中时有两种选择: 一种是 Bottom-up、另外一种 Top-down。这两种方式分别对应了自下而上和自上而下两种新建 Web Service 的模式。如果已经有了实现的功能类, 想要把它封装成 Web Service, 那么应当选择 Bottom up 模式; 如果首先定义了 Web Service 的描述文件 (WSDL), 接下来实现 WSDL 的功能, 则选择 Top down

模式。本论文选择自下而上的构建方式。

新建完上述四个服务后，在 vlabS 工程的配置文件 service.xml 中将有如下定义：

```
<service>
    <name>stu</name>
    <serviceClass>com.services.Istu</serviceClass>
    <implementationClass>com.services.stuImpl</implementationClass>
    .....
</service>
<service>
    <name>tea</name>
    <serviceClass>com.services.Itea</serviceClass>
    <implementationClass>com.services.teaImpl</implementationClass>
    .....
</service>
<service>
    <name>lab</name>
    <serviceClass>com.services.ILab</serviceClass>
    <implementationClass>com.services.labImpl</implementationClass>
    .....
</service>
<service>
    <name>class</name>
    <serviceClass>com.services.Iclass</serviceClass>
    <implementationClass>
        com.services.classImpl
    </implementationClass>
    .....
</service>
```

#### 4.2.2 通用服务数据库表的持久化

由于之前已经设计了通用功能服务相关的数据库，此时，只需使用 Hibernate 逆向工程，即可将关系数据库变成 Java 的对象，这样做的目的在于方便数据库操作，另外，由于使用了第三方缓存和数据库连接池，在性能上比传统的数据库访问有更大的优势。通过使用 Hibernate 3.0 对数据库 vlab 中的表进行逆向工程操作后，即生成了相应的 java

对象以及映射 xml 文件，以表 Student 为例，对应的 java 文件定义如下：

```
public class Student implements java.io.Serializable {
    private String stuId;
    .....
    private Set enrolls = new HashSet(0);
    public Student() {}
    public Student(String stuId) { this.stuId = stuId; }
    public Student(String stuId, String stuName, String stuPassword, String stuAddress,
String stuGender, String stuTelephone, String stuEmail, String stuTruename, Set enrolls) {
        this.stuId = stuId;
        .....
        this.enrolls = enrolls;
    }
    public String getStuId() {return this.stuId; }
    public void setStuId(String stuId) {this.stuId = stuId; }
    .....
    public Set getEnrolls() {return this.enrolls; }
    public void setEnrolls(Set enrolls) {this.enrolls = enrolls; }
}
```

其映射文件定义如下：

```
<hibernate-mapping>
    <class name="com.hibernate.Student" table="student" schema="dbo"
catalog="vlab" lazy="false">
        <id name="stuId" type="java.lang.String">
            <column name="stu_Id" length="50" />
            <generator class="assigned"></generator>
        </id>
        .....
        <set name="enrolls" inverse="true">
            <key><column name="stu_Id" length="50" /></key>
            <one-to-many class="com.hibernate.Enroll" /></set>
        </class>
</hibernate-mapping>
```

### 4.2.3 通用服务业务层的实现

在上一节使用 Hibernate 进行数据库表的持久化操作时, 如果选择了 Java Data Access Object (DAO), 并且 DAO 类型为 Spring DAO, 那么就可以直接得到数据库表的数据访问对象, 结合上文对通用服务接口的设计, 该数据访问对象需要实现接口中涉及各个方法类, 以 StudentDAOImp 为例, 该对象的定义如下:

```
public class StudentDAOImp implements StudentDAO {
    private static Log log = LogFactory.getLog(StudentDAOImp.class);
    public List getStudent() {
        // TODO Auto-generated method stub
        try {
            Session s = HibernateSessionFactory.getSession();
            Transaction tx = null;
            tx = s.beginTransaction();
            List results = s.createQuery("from Student stu").list();
            tx.commit();
            s.close();
            if (results != null && results.size() > 0) {
                return results;
            }
        } catch (HibernateException e) {
            log.fatal(e);
        }
        return null;
    }
    public boolean deleteStudentByID(String id) {.....}
    public Student getStudentByID(String id) {.....}
    public boolean updateStudent(Student stu) {.....}
    public boolean saveStudent(Student stu) {.....}
}
```

### 4.2.4 通用服务的发布

将上文构建的服务使用 MyEclipse 6.5 工具发布后, 可在浏览器中输入:  
[http://localhost:8080/vlabS/services/\\*\\*\\*\\*?wsdl](http://localhost:8080/vlabS/services/****?wsdl) 查看相应的 WSDL 文档 (\*\*\*\*为服务名, 如

stu)，下面以服务 stu 为例，简要介绍 stu 服务及其发布生成的 WSDL 文档。

stu.wsdl 该文档包括 type、message、portType、binding、service 五部分，都在标签 definitions 中进行描述，其 WSDL 文档如图 4-3 所示。

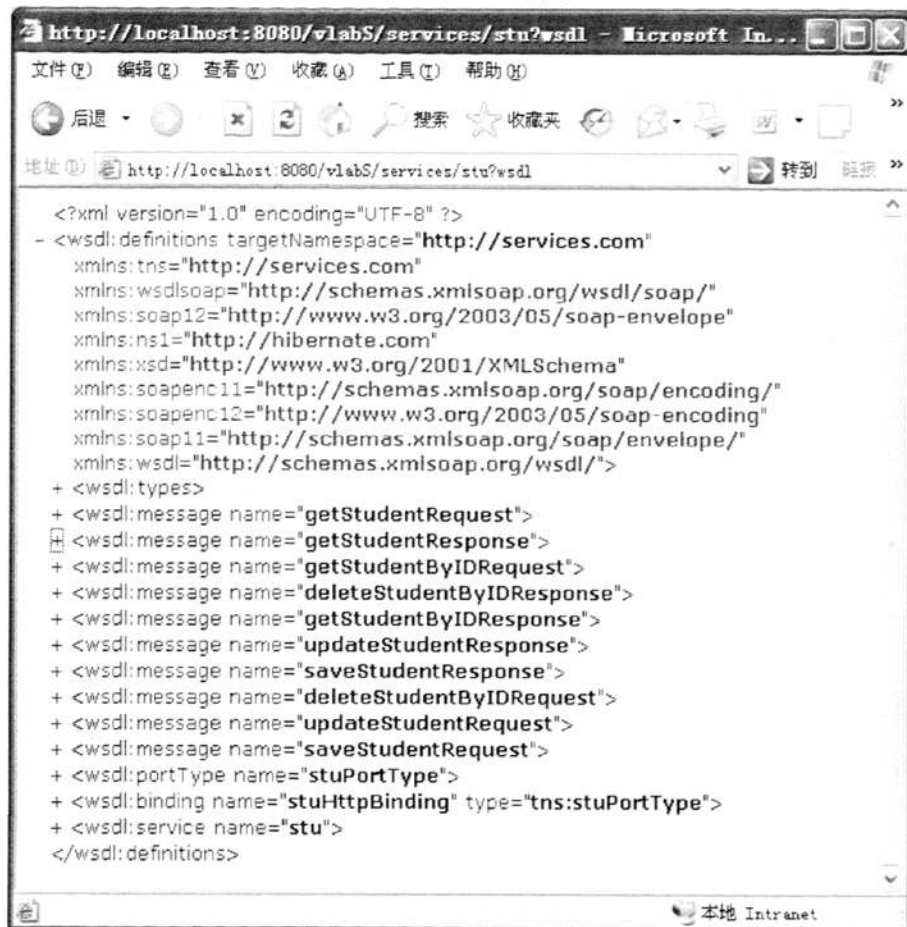


图 4-3 stu.wsdl 文档

其中，Type 用于定义独立与机器和语言的类型定义，如：

```
<xsd:schema
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
attributeFormDefault="qualified"
```

```
elementFormDefault="qualified" targetNamespace="http://services.com">
```

```
<xsd:element name="getStudentById">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element maxOccurs="1" minOccurs="0" name="in0" nillable="true"
type="xsd:int" />
```

```
</xsd:sequence>
```

```
</xsd:complexType>
```

```
</xsd:element>
```

```
.....
```

```
</xsd:element>
```

Messages 包括函数参数或文档描述, 如:

```
<wsdl:message name="getStudentByIDRequest">
```

```
  <wsdl:part name="parameters" element="tns:getStudentByID" />
```

```
</wsdl:message>
```

```
<wsdl:message name="getStudentByIDResponse">
```

```
  <wsdl:part name="parameters" element="tns:getStudentByIDResponse" />
```

```
</wsdl:message>
```

PortTypes 引用消息部分中消息定义来描述函数签名 (操作名、输入参数、输出参数), 如:

```
<wsdl:portType name="stuPortType">
```

```
<wsdl:operation name="getStudentByID">
```

```
<wsdl:input
```

```
  name="getStudentByIDRequest"
```

```
  message="tns:getStudentByIDRequest" />
```

```
<wsdl:output
```

```
  name="getStudentByIDResponse"
```

```
  message="tns:getStudentByIDResponse" />
```

```
</wsdl:operation>
```

```
</wsdl:portType>
```

Binding 中绑定实现 PortTypes 部分的操作, 如:

```
<wsdl:binding name="stuHttpBinding" type="tns:stuPortType">
```

```
<wsdlsoap:binding
```

```
  style="document" transport="http://schemas.xmlsoap.org/soap/http" />
```

```
<wsdl:operation name="getStudentByID">
```

```
  <wsdlsoap:operation soapAction="" />
```

```
<wsdl:input name="getStudentByIDRequest">
```

```
  <wsdlsoap:body use="literal" />
```

```
</wsdl:input>
```

```
<wsdl:output name="getStudentByIDResponse">
```

```
  <wsdlsoap:body use="literal" />
```

```
</wsdl:output>  
</wsdl:operation>  
.....
```

```
</wsdl:binding>
```

Services 用于确定绑定的端口地址。如：

```
<wsdl:service name="stu">  
  <wsdl:port name="stuHttpPort" binding="tns:stuHttpBinding">  
    <wsdlsoap:address location="http://localhost:8080/vlabS/services/stu" />  
  </wsdl:port>  
</wsdl:service>
```

## 本章小结

本章从用户身份合法性验证和服务端 SSL 服务的开启介绍了用户统一身份认证服务器端的部署细节；4.2 节详细介绍了虚拟实验室平台通用服务接口的创建，数据库表的持久化，业务层实现以及服务的发布细节。

## 第五章 基于平台的虚拟实验室构建

### 5.1 计算机硬件组成实验室背景介绍

大学计算机基础是一门系统介绍计算机基本概念, 强调文化与信息意识的基础课程。该课程涉及计算机教育领域的计算机基本知识、计算机体系结构、微机硬件基础、操作系统技术基础知识、软件开发基础、数据库应用基础、多媒体技术应用基础和网络技术基础等内容。教学目标是使学生通过对本课程的学习, 能概括计算机软、硬件技术与网络技术的基本概念, 阐明软件设计与信息处理的基本过程, 概述典型计算机系统的基本工作原理, 能独立安装、设置与操作典型的计算机平台。

大学计算机基础课程较全面、概括性地讲述计算机科学与技术学科中的一些基础性知识和重要概念, 主要包括:

1. 计算机系统基础: 计算机的发展简史, 计算机的基本原理、特点、应用, 计算机硬件系统的基本组成, 计算机软件系统概述等;
2. 信息的表示与计算机基本工作原理: 计算机运算基础, 计算机的指令和指令系统等;
3. 操作系统基础: 操作系统基本原理、功能、种类及发展简况, 典型操作系统的安装、配置和使用等;
4. 计算机网络初步: 计算机网络基本知识, 网络与分布式计算简介, 典型信息服务的使用等;
5. 软件开发与信息处理技术: 程序设计与程序设计语言的基本概念, 程序设计的基本过程, 软件开发方法, 数据库管理系统基本概念等;
6. 信息系统安全与社会责任: 信息安全的基本概念与状况、计算机病毒、网络黑客及网络攻防、系统安全规划与管理, 社会责任与职业道德规范等。

由此可见, 认识计算机的硬件结构是大学生计算机基础课程的基础。集中授课和上机实践是计算机基础教学中的两个重要环节, 上机实践主要进行计算机基本应用技能的练习, 在集中授课环节, 由于条件限制, 教师往往无法展示具体的计算机硬件, 而只能通过图片的形式来演示, 学生能够直接观察到的也仅仅是键盘、鼠标和显示器等外部设备, 而对于主板上的 CPU 插座、CPU 调压器、主板芯片组、存储器插座、总线插槽等细节, 学生只能获取概念层面的知识, 而无法知道这些部件究竟是什么样子, 以及如何排列等。为了让学生对计算机硬件有直观的认识, 本章将结合虚拟实验室平台, 构建基于大学计算机基础课程的计算机硬件组成虚拟实验室。

## 5.2 计算机硬件组成虚拟实验室的实现

首先需要在 MyEclipse 6.5 中, 新建 Web 工程 vlab, 同时为该工程添加 XFire 1.2, Struts, Spring 1.2, Hibernate 3.0 框架。

### 5.2.1 统一身份认证客户端配置

由于计算机硬件组成虚拟实验室作为虚拟实验室平台的主体部分之一, 同时也是统一身份认证模块的客户端, 而统一身份认证模块采用基于 SSL 的双向认证模式, 因此, 在客户端也需要开启 SSL 服务, 其流程与服务器端开启方式类似, 此处不再赘述。作为统一身份认证模块客户端的虚拟实验室系统, 为例确保其支持单点登录、登出, 需要对其进行相应的部署。首先, 需要为工程 vlab 添加包 Yale Cas Client 3.1.6 之后, 然后在该工程的 web.xml 添加过滤器, 核心代码如下:

```
<filter>
    <filter-name>CAS Single Sign Out Filter</filter-name>
    <filter-class>
        org.jasig.cas.client.session.SingleSignOutFilter
    </filter-class>
</filter>
<filter-mapping>
    <filter-name>CAS Single Sign Out Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

<filter>
    <filter-name>CAS Filter</filter-name>
<filter-class>edu.yale.its.tp.cas.client.filter.CASFilter</filter-class>
    <init-param> <param-name>edu.yale.its.tp.cas.client.filter.loginUrl</param-name>
        <param-value>https://localhost:8443/cas/login</param-value>
    </init-param>
    <init-param>
<param-name>edu.yale.its.tp.cas.client.filter.validateUrl</param-name>
<param-value>https://localhost:8443/cas/serviceValidate</param-value>
    </init-param>
    <init-param>
<param-name>edu.yale.its.tp.cas.client.filter.serverName</param-name>
    <param-value>127.0.0.1:8080</param-value>
```

```
</init-param>
</filter>
<filter-mapping>
    <filter-name>CAS Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

通过以上配置,用户首次登陆虚拟实验室平台,或者注销后再次访问平台资源时,浏览器会重定向到虚拟实验室平台统一入口,如图 5-1 所示:



图 5-1 虚拟实验室平台统一身份认证入口

Tomcat 服务器端监测信息如图 5-2 所示:

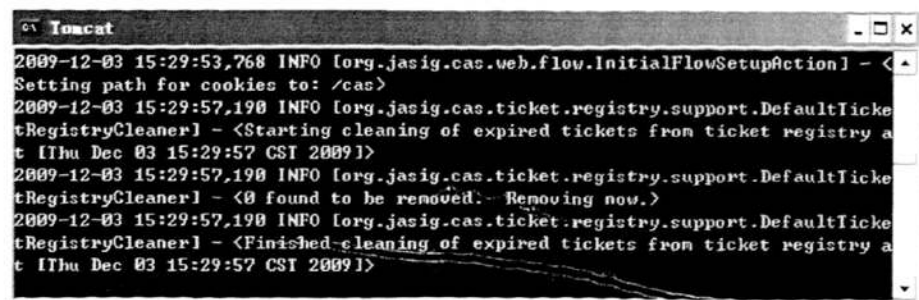


图 5-2 Tomcat 服务器对 CAS 服务器的过滤监测

### 5.2.2 通用服务的调用

由于虚拟实验室通用功能可以通过调用第四章设计实现的 vlabS 服务, 因此, 在本实验室的构建中, 表现层, 使用 Struts, 业务层则通过调用 vlabS 提供的服务实现。下面以维护学生信息为例, 简单介绍具体的设计及相关调用关系。表现层的页面跳转关系如图 5-3 所示:

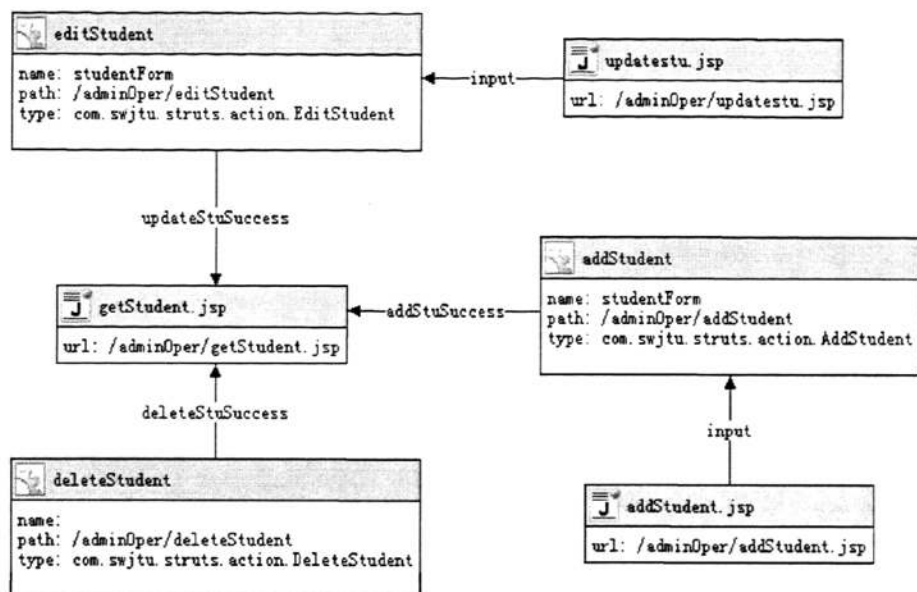


图 5-3 学生信息维护页面跳转图

对于业务层, 则需要根据 vlabS 提供的 WSDL 地址即可新建 Web Service Client。如, 由 <http://localhost:8080/vlabS/services/stu.wsdl> 构建的客户端完成后, 会生成 `stuPortType.java` 和 `stuClient.java`, 其中 `stuPortType.java` 是一个接口类, 定义了根据 WSDL 文档描述的各个功能的名称、名字空间信息, 该部分代码片段如下所示:

.....

```
@WebService(name = "stuPortType",
             targetNamespace = "http://services.com")
@SOAPBinding(use = SOAPBinding.Use.LITERAL,
             parameterStyle= SOAPBinding.ParameterStyle.WRAPPED)
public interface stuPortType {
    @WebMethod(operationName = "getStudentByID", action = "")
    @WebResult(name = "out", targetNamespace = "http://services.com")
    public Student getStudentByID(
        @WebParam(name = "in0",
                 targetNamespace = "http://services.com")
```

```
Integer in0);
.....
}
stuClient.java 则是对 stu 服务的具体调用, 该部分代码片段如下:
.....
public class stuClient {
.....
    public stuClient() {
        create0();
        Endpoint stuHttpPortEP =
service0.addEndpoint(new QName("http://services.com","stuHttpPort"),
                        new      QName("http://services.com","stuHttpBinding"),
                        "http://localhost:8080/vlabS/services/stu");
        endpoints.put(.....);
        Endpoint stuPortTypeLocalEndpointEP=
service0.addEndpoint (.....);
        endpoints.put(.....);
    }
.....
}
```

虚拟实验室平台提供的其他服务都可以通过类似的操作被调用, 此处不加以赘述。

### 5.2.3 计算机硬件组成虚拟实验的仿真实现

主板作为计算机最基本的也是最重要的部件之一, 认识主板的构造以及各类接口的位置是进行各类计算机基础实验的前提, 因此, 本论文设计了认识主板实验, 目的在于让学生熟悉常见微机主板的基本结构及布局, 了解主板所用元器件的外形特征。

#### 5.2.3.1 基于 3DS MAX 的计算机主板仿真建模

本论文以超微主板为原型, 使用 Atudesk 3DS MAX 2009 作为开发工具, 制作了主板模型。3DS MAX 是目前应用最广泛的三维建模、动画、渲染软件之一, 该软件的可以通过中的颜色、透明度、光亮度、材质贴图等参数来实现三维彩色的效果。

首先, 需要对主板上的各类接口进行建模, 包括: 串口连接器, 鼠标/键盘接口, 内存插槽, 硬盘接口, 基于 PCI 局部总线的扩展插槽, 通讯网络插卡, 中央处理器插

槽等, 上述部件可以通过扭曲、组合 3DS MAX 2009 提供的“创建”多边形、多面体等工具实现。以对串行口进行建模为例, 首先需要创建一个 L 形挤出的扩展基本体, 两个长方体, 两个圆柱体, 将其中一个长方体与 L 形挤出体组合在一起, 另一个则转换成可编辑多边形, 对其进行倒角等处理, 然后需要对以上各部分根据需要进行材质渲染处理。为了增强真实性, 模型各部分的材质使用不同的处理方式, 如串行口、鼠标键盘插孔、内存插槽等则使用不同的光泽度、环境光、高光、漫反射图片等方式进行材质设置, 从而在将不同材质指定给不同的面、元素时, 营造良好的视觉效果。其余各个部分的处理与串行口的建模方式类似, 此处不再赘述。需要指出的是, 对于主板上的电路板, 由于细节内容比较繁杂, 逐一呈现存在较大困难, 因此, 在主板模型的设计中并没有对上述此节分别建模, 而是通过主板高清图片进行材质渲染的方式实现。模型构造完成后, 由于 Autodesk 3DS MAX 2009 支持 VRML 格式的文件输出, 本文直接将模型导出为 mb.wrl。mb.wrl 在安装有支持 VRML 格式文件解析的浏览器中就能查看, 本文采用 ParallelGraphics Cortona VRML Client 5.1 插件进行浏览, 图 5-4 为主板的模型效果图。

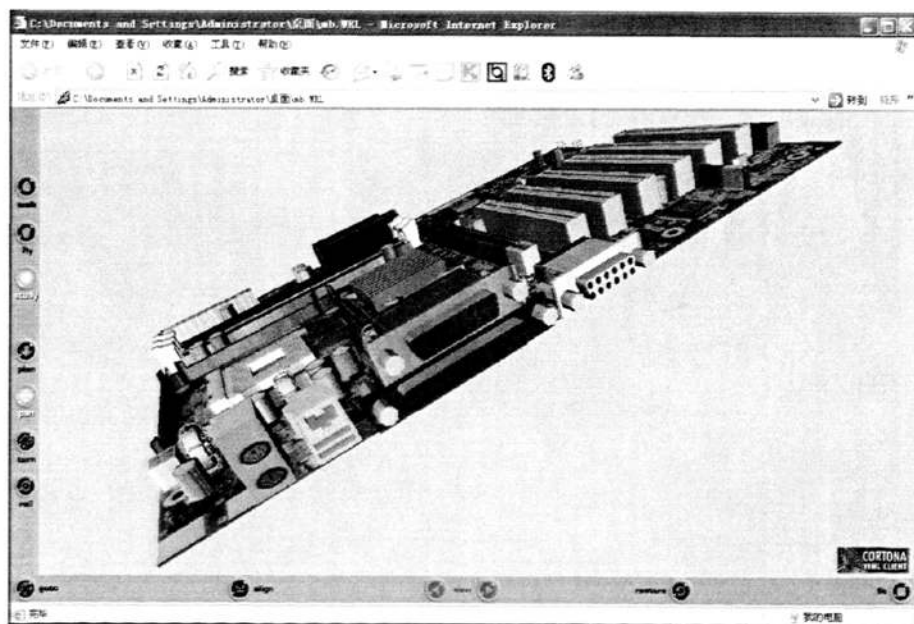


图 5-4 主板模型效果图

#### 5.2.3.2 基于 VRML 的计算机主板模型优化

三维建模和渲染完成之后, 就可以将场景导出为 wrl 文件。由于 wrl 文件中含有大量冗余数据, 有必要对其进行优化。优化包括对模型的优化和对 VRML 文件本身的优化。在 3DSMax 中, 对于复杂材质的目标采用二次建模, 先精细建模, 赋以纹理与材

质, 然后对所需的纹理进行渲染输出。首先, 对精细建模的目标结构进行分解, 采用拉伸或放样等功能生成简单几何体的组合, 最后, 将上述输出的包含细节信息的纹理对几何体进行贴图, 完成模型的构建可以减少三角面片的数量。

对 VRML 文件自身的优化可以使用 VRML 中的 LOD (level of detail) 节点。LOD 节点根据人眼对同一个物体在远近不同的位置所能看到的细节详细程度不同的视觉特性, 对于复杂模型进行优化处理时, 需要为模型建立一组详细程度有别的几何模型, 模型离视点越远, 建立的模型包含的细节就越少, 从而加速模型绘制速度提高浏览虚拟场景的流畅性。此外, 对于相同造型的物体先用 DEF 语句对节点命名, 再通过 USE 引用就可获得物体的完整复制; 而对相近造型的物体可以使用原型节点 PROTO 来复制使用, 这样可大大简化代码。

本文使用 VrmIpad 2.1 工具对模型进行优化修改, 图 5-5 为优化后的模型构造结构:

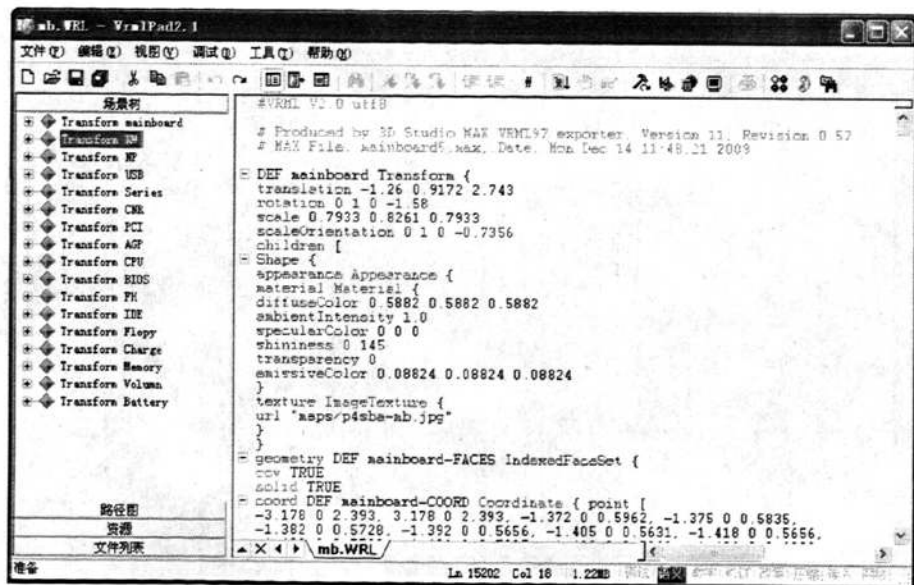


图 5-5 使用 VRMLPad 对主板模型进行优化设计

直接生成的 VRML 文件较大, 再加上在模型渲染时需要用到其他的图片文件, 庞大的模型文件占用空间对于网络传输的实时性是不利的, 但利用 VrmIpad 的压缩功能, 可在很大程度上减小文件容量。以本文构建的 mb.wrl 文件为例, 压缩之前文件大小为 1.21MB, 压缩后的文件大小为 247KB, 文件容量减少了 80%左右。

### 5.2.3.3 认识计算机主板部件的设计

为了使学生们了解主板上各个接口、插槽的名称及作用, 本实验做了如下设计: 首先向学生展示主板高清图片, 当学生将鼠标指向图片的某个位置时, 对应区域会高亮显示同时提示文字信息, 该部分适用 JavaScript 实现控制。

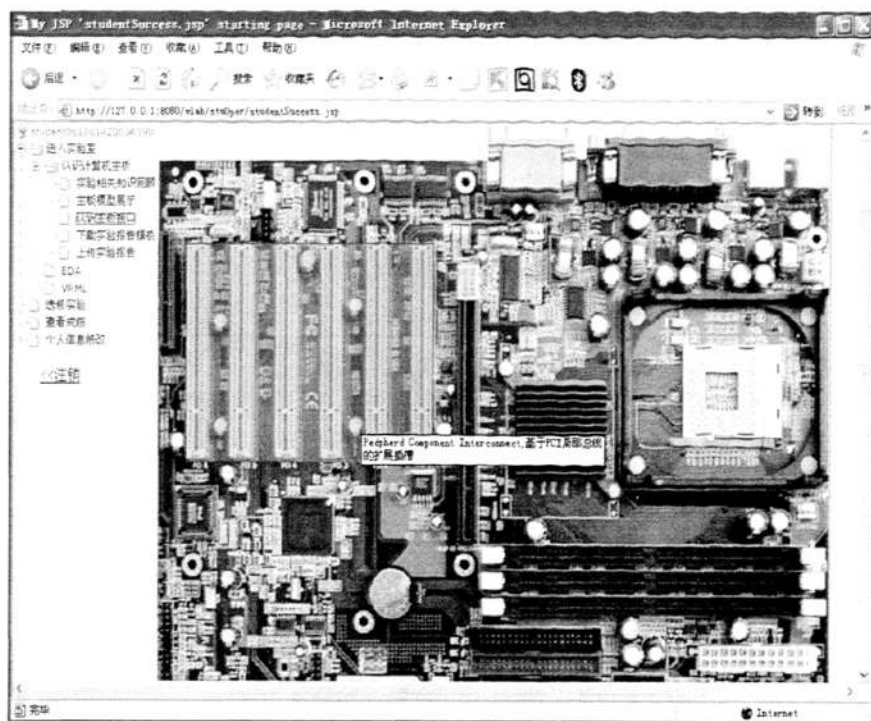


图 5-6 认识主板接口效果图

图 5-6 为该部分设计的效果图。

### 5.3 虚拟实验室平台功能验证

为了验证基于虚拟实验室平台开发的实验室之间的数据通信以及统一身份认证服务的有效性，本论文新建了名为 vlab2 的工程，该工程开设的实验名为 EDA，由于研究时间比较仓促，尚未为该实验室添加任何实验，只用于统一身份认证服务的验证。vlab2 仍然采用与 vlab 类似的设计路线，表现层使用 Struts，业务层则通过调用 vlabS 提供的服务实现。

下面以用户名为 861061420034390 的学生为例，简单介绍该学生虚拟实验室中穿梭时的情况。需要说明的是，学号为 861061420034390 的学生已经提前在虚拟实验室平台中申请注册了计算机硬件组成、EDA、VRML 三个实验，并且已通过实验指导教师的注册审核。

对于已选修实验，并被教师接受到该班级的学生，在进入任何一个实验室时，浏览器都会被重定向到虚拟实验室统一入口。因此，当学生 861061420034390 在浏览器地址栏中输入虚拟实验室平台地址后，浏览器会被重定向至虚拟实验平台入口（本文 5.3.1 已作介绍），CAS 服务器处理学生信息凭证，该学生通过身份验证后，CAS 服务器将 TGC 发送到用户浏览器，同时为 vlabhome 应用服务生成唯一的 ST，同时用户浏览器页面跳转至 vlabhome 应用页面。

图 5-7 为用户登陆成功后的 vlabhome 页面, Tomcat 服务器对上述过程的监测情况如图 5-8 所示:

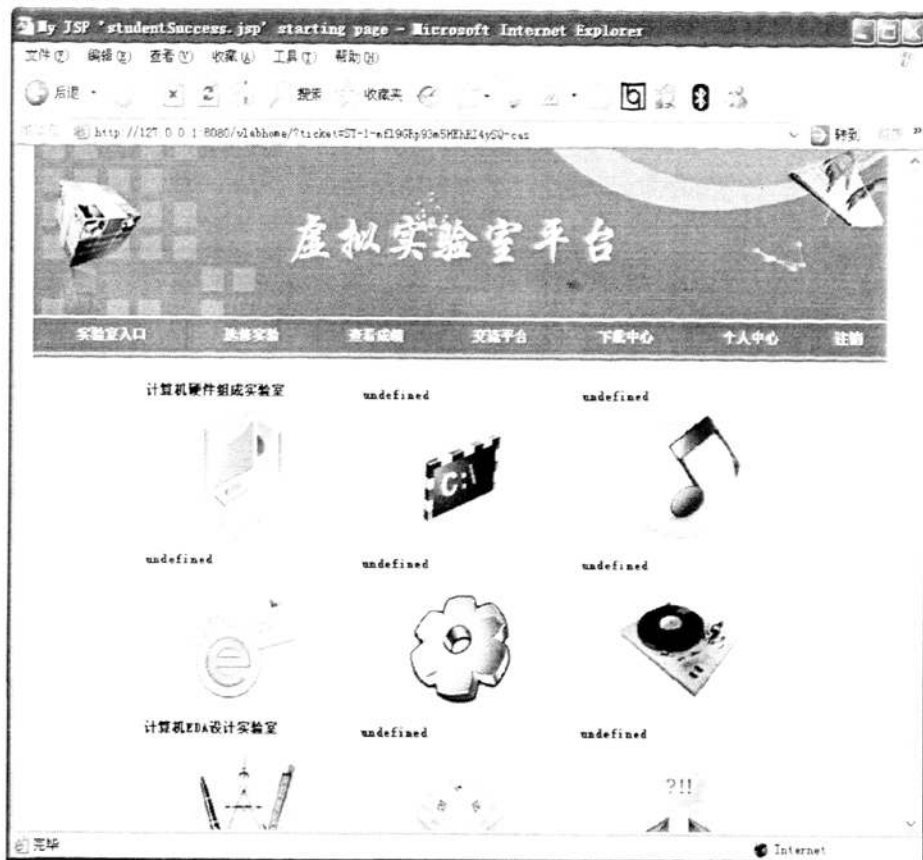


图 5-7 CAS 服务器完成用户合法性验证后跳转至应用页面

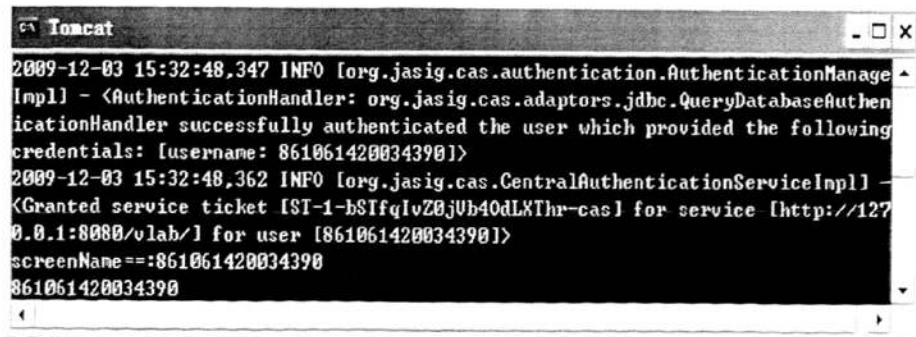


图 5-8 Tomcat 服务器对 CAS 身份验证过程的监测

此时, 由于该学生已经注册计算机应用实验并且已被实验指导教师接受, 因此, 可以选择进入认识计算机主板虚拟实验室, 并对已授权的功能进行操作。

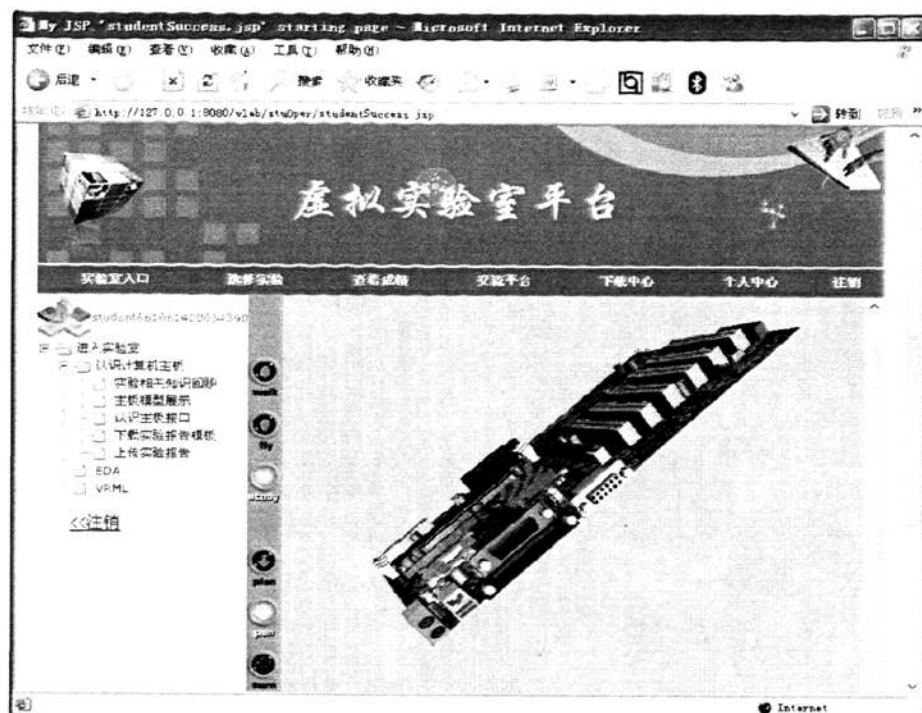


图 5-9 学生用户进入认识计算机硬件虚拟实验室界面

图 5-9 所示为 861061420034390 学生用户进入认识计算机主板虚拟实验后的效果图。

此时，如果该学生需要进入其他实验室，可点击页面中的链接进入，由于该浏览器中保存了 TGC，在页面重定向到 CAS 服务器的时候，CAS 服务器主动获取到这个 TGC，因此，该学生可以不用重新登录虚拟实验室平台，直接进入 vlab2，也就是 EDA 实验室。CAS 服务器端同时为 vlab2 应用生成 ST，Tomcat 服务器端的监测信息如图 5-10 所示：

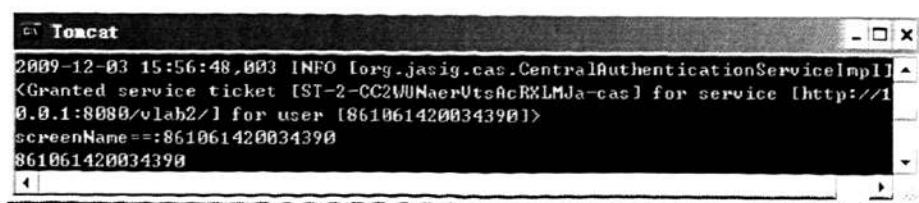


图 5-10 Tomcat 服务器对 CAS 服务器生成 ST 的过程监测

这样一来，该学生就能同时使用计算机硬件组成实验室和 EDA 实验室，学生可以方便地根据需要调用实验室间的数据。

当实验结束需要退出实验室时，只要在任意一个实验室点击注销，浏览器就会重定向到统一的登出页面 <https://localhost:8443/cas/logout>，即可实现所有实验室的登出。此时，如果需要再进入某个实验室，由于 CAS 服务器已经销毁 TGC 以及与其相关的 ST，因此，浏览器就会被重定向到 5.3.1 节提到的虚拟实验室平台入口，只有重新登录

才能进入实验室。

## 本章小结

本章从计算机硬件组成虚拟实验室构建的背景介绍出发，分别实现了该实验室对本文第四章通用功能服务的调用，并实现统一身份认证客户端的配置，在此基础上，5.2.3 节应用 3DSMAX 和 VRML 制作了计算机主板模型并进行了优化，充实了虚拟实验室的内容；本章的最后简单验证了在虚拟实验室平台中，统一身份认证在虚拟实验室之间进行跳转时的效果，基本实现了快捷的应用交互。

## 总结与展望

### 总结

在现代远程教育中,由于教学机构和学生在空间上的分离,学生难以到学校实验室做具体的实验,而实验教学在学生获取知识的过程中占有非常重要的地位,这已成为制约现代远程教育质量的一个重要因素。虽然目前虚拟实验室平台在教育领域中的实际应用已有很大进展,但在很多方面,还存在很多尚未解决的理论问题和尚未克服的技术障碍。

本文在研究国内外虚拟实验室平台研究现状、已有虚拟实验室平台构建技术以及虚拟实验开发技术的基础上,着眼于跨学科虚拟实验室通用平台的构建,提出了一个虚拟实验室平台设计方案,着重解决虚拟实验室间资源共享、虚拟实验交互能力、用户访问不同虚拟实验室时存在的障碍等问题。为了解决以上问题,本文从分析虚拟实验室平台功能分析入手,针对虚拟实验室之间的数据流通、保密数据的维护等因素,提出了使用 Web 服务以及统一身份认证技术的虚拟实验室平台构建方案。随后,本文分别对统一身份认证和虚拟实验室通用功能两个大模块进行详细的体系结构、业务流程、数据库以及接口设计。在虚拟实验室平台功能实现方面,本文基于耶鲁大学 CAS 认证模型对虚拟实验室平台的统一身份认证进行了部署实现,为虚拟实验室通用功能构建实现了部分基于 XFire 框架的功能接口;在虚拟实验构建方面,本文使用 3DMAX 和 VRML 设计了认识计算机主板的实验;为了验证虚拟实验室平台统一身份认证以及通用功能服务的调用,本文的最后构建了基于计算机硬件组成的虚拟实验室,基本实现了前文对虚拟实验室平台设计的构建蓝图。

### 存在问题及展望

由于时间和精力的不足,本文在对虚拟实验室平台进行设计和实现的过程中还存在以下问题:

1. 本文对于虚拟实验室平台数据流通安全性方面的研究不够深入细致,只设计了自定义签名证书来保障数据在用户和服务器间进行交换时的安全,而没有一个详尽的安全模型防止对平台功能服务的恶意访问。
2. 尽管本文对于虚拟实验室平台的设计初衷是可以实现跨平台、跨开发语言的效果,但是由于条件限制,在对虚拟实验室功能进行验证性调用时,没有在跨平台、跨开发语言的环境中进行开发。
3. 受个人经验所限,虚拟实验室平台在设计和实现的效率上难免有缺陷,希望能在以后的工作中做进一步的完善和改进。

## 致谢

值本毕业论文完成之际，首先我要感谢我的导师。由衷感谢老师三年来在我学业和生活上的倾力培养和深切关心，从研究领域的兴趣培养，研究技能的提高，直至本论文的开题，设计期间的关心和指导，都饱含着导师辛勤的汗水和心血。导师博学的知识、认真负责的精神、乐观的态度以及不知疲倦的工作热情将是我今后学习和工作的榜样。

感谢西南交通大学的每一位老师，在西南交通大学七年的学习生涯里，我每一分的收获都有你们的功劳；感谢软件学院学生工作组的老师们，是你们的信任提高了我的组织能力；同时所有在学习和生活中关心帮助包容过我的同学们，愉快的学习和生活氛围永远是我人生美好的回忆。

感谢我的父母和家人，感谢你们对我永不停歇的关爱和教诲，你们的关心和支持是我每次扬帆迎接挑战的动力之源。

最后，衷心感谢各位专家和学者在百忙之中对本文的审阅，感谢专家们提出的宝贵意见和建议！

## 参考文献

- [1] Richard W. Riley. E-Learning: Putting a World-Class Education at the Fingertips of All Children (The National Educational Technology Plan). U. S. Department of Education, Office of Educational Technology. Dec. 2000
- [2] Marco Anisetti, Valerio Bellandi, Alberto Colombo. Learning Computer Networking on Open Paravirtual Laboratories. IEEE TRANSACTIONS ON EDUCATION, VOL. 50, NO.4. NOVEMBER 2007
- [3] J. W. Rickel. Intelligent computer-aided instruction: A survey organized around system components. IEEE Trans. on System Man Cybernet. 1989: 40~57
- [4] 上海市教科院智力开发研究所. 美国教育部教育技术白皮书. 2001
- [5] A.Ferrero, V.Piuri. A Simulation Tool for Virtual Laboratory Experiments in a WWW Environment. IEEE Transaction on Instrumentation and Measurement. 1999: 741~ 746
- [6] Reinhard Klein, L.Miguel Encarnacao. A Web-based Framework for the Complete Integration of Teaching Concepts and Media in Computer Graphics Education. In: T.Ottmann, I.Tomek eds. Proceeding of ED-MEDIA' 97,Vancouver , Canada, 1997: 231~240
- [7] James P. Vary. Report of the Expert Meeting on Virtual Laboratories. United Nations Educational Scientific and Cultural Organization, Paris, 2000: 3~5
- [8] Polly Huang, John Heidemann. Minimizing Routing State for Light-weight Network Simulation. In: Fujimote eds. Proceedings of the International Symposium on Modeling. Analysis and Simulation of Computer and Telecommunication Systems (MASCO TS, 01), Cincinnati, Ohio, USA. 2001: 108~116
- [9] 周琪锋.虚拟实验室的建设与应用探讨. 农业网络信息. 2006, (8): 88~89
- [10] Eliane Guimaraes, Antonio Maffei, Jamas Pereirat, et al. REAL: A Virtual Laboratory for Mobile Robot Experiments. IEEE Transaction On education, 2003, 46(1):37-42
- [11] Khoarl. <http://www.khoral.com/contrib/bontrib/dip2001/index.html>. 2001
- [12] 喻红, 何岭松, 王峻峰. WWW模式的工程测试远程教学实验基地建设. 实验技术与管理. 1999, 16:1
- [13] John T, Bell, Fogler H Scott, Vicher. A Prototype Virtual Reality Based Educational Module for Chemical Reaction Engineering. Computer Applications in Engineering Education. 1996.4(4).
- [14] Saliah H H, Burnett P D, Loizeau C, RV LabX. A Web-Based Interactive Laboratory Environment for Education and Research. Proceedings of the 31<sup>st</sup> Annual Frontiers in Education Conference, Albuquerque ,USA, 2001: T4C/5-T4C/9.
- [15] Saliah H H, Villardier L, Assogba B, Kedowide C, Wong T. Resource Management Str-

- ategies for Remote Virtual Laboratory Experimentation. Proceedings of the 30<sup>th</sup> Annual Frontiers in Education Conference, Kansas City ,2000:T1D/8~T1D/12
- [16] Saliah H H, Nurse E, Abecassis A. Design of a Generic, Interactive, Virtual and Remote Electrical Engineering Laboratory. Proceedings of the 29<sup>th</sup> Annual Frontiers in Education Conference, San Juan, Puerto Rico ,1999:12C6/18~12C6/23
- [17] 张耀鸿, 罗雪山, 余滨. 基于组件的分布离散事件仿真环境. 系统仿真学报. 2002: 1019~1021
- [18] A VIRTUAL ENGINEERING/SCIENCE LABORATORY COURSE. 2000.2.29.  
[http:// www.jhu.edu/virtlab/virtlab.html](http://www.jhu.edu/virtlab/virtlab.html)
- [19] R.Subramanian, I.Marsic, ViBE. Vitural Biology Experiments. In: Lynda Hardman eds. Proceedings of the Ten International World Wide Web Conference (WWW10), Hong Kong, China, 2001: 316~325
- [20] L.Fabrega, J.Massaguer, T.Love, et al. A Virtual network laboratory for learning IP network. In: Greg Sarris eds. The 7<sup>th</sup> Annual Conference on Innovation and Technology in Computer Science Education, Aarhus, Denmark , 2002: 89~95
- [21] 杨楠, 王云莉, 杨涛, 肖田元. 虚拟企业信息基础设施框架研究田. 计算机集成制造系统CIMS. 2002, 8 (3): 182~186
- [22] Dana M.John H. 对等网. 苏忠, 战晓雷译. 清华大学出版社, 2003: 118. 12
- [23] Albu, M.M, Holbert, K.E, Heydt, Grigoreseu, S.D, Trusea, V. Embedding Remote Experimentation in Power Engineering Education. IEEE Transactions on Power Systems. 2004, 19(1): 139~143.
- [24] Nedie, Z, Maehotka, J, Nafalski. A Remote laboratory versus virtual and real laboratories. 33<sup>rd</sup> Annual Frontiers in Education, Boulder, Colorado, 2003:20~25.
- [25] Rohrig, C, Bisehoff. A Web-based environment for collaborative remote experimentation. Proceedings of 42<sup>nd</sup> IEEE Conference on Decision and Control, Maui, HI, USA , 2003: 18~23.
- [26] Ciubotariu C, Hancock G. Work in progress-virtual laboratory with a remote control instrumentation component. 34<sup>th</sup> Annual Frontiers in Education, Savannah, GA , 2004:67~71
- [27] W3C. Web Services Architecture[EB/OL]. <http://www.w3.org/>
- [28] 李安渝. Web Service技术与实现. 国防工业出版社, 2003
- [29] Mark Endrei. Patterns: Services oriented Architecture and Web Services. <http://ibm.com/red books>. 2004
- [30] H. Afsarmanesh, E.C. Kaletas. A. Benabdelkader. A reference architecture for scientific virtual laboratories. Future Generation Computer Systems. 2001:999~1008
- [31] Dongilshin, Ensup Yoon, etc. Web based interactive virtual laboratory system for unit operations and process systems engineering education. Computers and Chemical Engineering. 2000:1381~1385

- 
- [32] Earnshaw, R.Jones, Gigante. Virtual Reality systems. Academic Press. 1996:43~45.
- [33] Codehaus Xfire. <http://xfire.codehaus.org/>
- [34] 高焕芝. 单点登录技术的研究. 北京邮电大学, 硕士学位论文. 2006
- [35] Single Sign On. IBM[EB/OL]. <http://www.ibm.com/developerworks/cn/security/se-ssso/index.html>
- [36] Gang Zhao, Dong Zheng, Kefei Chen. Design of Single Sign-On of the IEEE International Conference on E-Coerce Technology. Proceedings for Dynamic E-Business. 2005
- [37] 沈浩, 薛贵荣. Cookie的安全及其解决方案. 计算机工程与应用. 2002, 149~151
- [38] 李笑, 董云耀. 基于WEB的虚拟实验室的研究与设计. 计算机与数字工程. 2006, 34(2): 132~134
- [39] 李圣良. 电类专业虚拟实验室的构建. 实验室研究与探索. 2005, 24(4): 41~42
- [40] 张枝军. 教学实践中虚拟现实的实现途径. 无锡商业职业技术学院学报. 2005, 5(1): 71~73
- [41] 宁超, 张世英, 高巍然, 韦素媛. 网络虚拟实验的研究与开发. 实验技术与管理. 2005, 22(4): 76~80
- [42] 程永军, 刘蔚. 基于VRML的实时交互式虚拟教学系统. 电化教育研究. 2002(4): 45~47
- [43] 卢宇, 陈宏敏, 赖恒, 李丽滨. 虚拟现实技术在光学实验教学中的应用. 福建师范大学学报(自然科学版). 2004, 20(3): 106~108
- [44] 王鸣, 刘喜昂, 宋蔚. 基于VRML的虚拟实验系统研究. 现代电子技术. 2006, 230(15): 105~112
- [45] Jaime Camelio, S JackHu. Modeling Variation Propagation of Multi-station Assembly Systems with Compliant Parts. Journal Mechanical Design. 2003, 125(4): 673~681.
- [46] 赵小林. 网页制作技术教程——虚拟现实篇. 国防工业出版社, 2002
- [47] 龚曙光等. CAE仿真分析中计算精度与网格划分关系的探讨. 现代机械. 2003, (6): 35~38.
- [48] 夏平, 黄菊生, 吴安如. 铆接汽车车架的接触法分析与计算. 机械设计. 2005, 22(3): 35~37.
- [49] 何利等. 基于虚拟技术的实验教学. 实用测试技术. 2001, (4)
- [50] <http://www.jhu.edu/~virtlab/virtlab.html>
- [51] <http://neon.chem.ox.ac.uk/vrchemistry>
- [52] Craig Walls, Ryan Bredenbach. Spring in Action中文版.第二版. 人民邮电出版社, 2006
- [53] 杨敬伟, 薛玉倩, 刘振鹏. 基于C/S与B/S混合模式的软件体系结构. 河北大学学报(自然科学版). 2006, 26(3):315~318
- [54] Sun Microsystems. Sun Java System Access Manager Administration Guide. <http://java.sun.com>
-

- 
- [55] 张颖江, 郑秋华, 李腊元. 单次登录技术分析及其集中身份平台设计研究. 武汉理工大学学报. 2004, 28(2): 240~243
- [56] 任栋, 刘连忠. 一种Web应用环境下安全单点登录模型的设计. 计算机工程与应用. 2005, 2(7): 100~103
- [57] Gaiyo Ellison, Jeff Hodges, Susan Landau. Risks Presented by Single Sign-On Architectures. <http://research.sun.com/liberty/RPSSOA/index.html>
- [58] 陈重威. 一个Web应用单点登录系统的设计和实现. 哈尔滨师范大学自然科学学报. 2004, 20(1): 47~52
- [59] D.Kristol, L.Mont. HTTP State Management Mechanism. <ftp://ftp.isi.edu/in-notes/rfc2965.txt>, 1997
- [60] Chi Chung K, Chen B M, Shaoyan H, Ramakrishnan V, Chang Dong C, Yuan Z, Jianping C. A web-based virtual laboratory on a frequency modulation experiment. IEEE Transactions on Systems, Man and Cybernetics, Part C, 2001, 31(3): 295~303.
- [61] Cheng K W E, Chan C L, Cheung N C, Sutanto D. Virtual laboratory development for teaching power electronics. Cairns, Australia: Institute of Electrical and Electronics Engineers Inc., 2002 vol.2: 461~466.
- [62] 清华大学工程力学虚拟实验室. <http://www.xauat.edu.cn/ex/tsinghua/software/08/01/07/01/00001/index.htm>
- [63] 浙江大学物理虚拟实验. <http://zjuphyllab.zju.edu.cn/xnsy/run/Frame-1.htm>
- [64] Sandhu R, Coyne E, et al. Role Based Access Control Models. IEEE Computer, 1996, 29(2): 38~47.
- [65] Ravi Sandhu, Venkata Bhamidipati, Qamar Munawer. The ARBAC97 Model for Role Based Administration of Roles. ACM Transactions on Information and System Security, 1999, 2(1): 105~135.
-

## 攻读学位期间发表的论文

- [1] 陈怡.Wonderland在虚拟实验室中应用初探.中国科教创新导刊, 2009年第25期