

密级： 保密期限：

北京邮电大学

硕士研究生学位论文



题目：计算机网络虚拟实验智能指导的设计与实现

学 号：076906

姓 名：艾建锋

专 业：计算机应用

导 师：文福安

学 院：网络教育学院

2010 年 1 月 10 日

独创性（或创新性）声明



本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：艾建锋 日期：2010.3.15

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在__年解密后适用本授权书。非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名：艾建锋 日期：2010.3.15

导师签名：文江安 日期：2010.3.15

计算机网络虚拟实验智能指导的设计与实现

摘 要

虚拟实验是远程教学的一种重要手段，是网上教学内容的一项重要补充。然而由于远程教学的特殊性，学生在实验过程中遇到困难无法得到及时的解答，降低了虚拟实验的实用性，成为远程教学的一个瓶颈。

论文针对计算机网络虚拟实验智能指导系统在设计开发过程中遇到的难点进行研究，在原有的计算机网络虚拟实验系统及智能批改系统基础上，主要解决了知识规则转换器和基于 J2EE 的 web 开发集成专家系统两方面的问题。

在原有的计算机网络虚拟实验 J2EE 框架基础上，本论文主要描述了针对计算机网络智能指导的需求、设计、实现工作，依托于专家系统、规则引擎的相关理论知识，在虚拟实验原有系统基础上对虚拟实验系统进行二次开发。

论文完成了计算机网络虚拟实验智能指导系统的设计和开发工作，经过系统的测试和短期使用，在教学实践中展开应用，收到良好效果。实践证明，利用专家系统对虚拟实验进行智能指导的研究思路是正确可行的，本文所谈到的相关技术和设计开发方案是正确的。

关键词： 智能指导，J2EE，专家系统，规则引擎

DESIGN AND REALIZATION OF INTELLIGENT-GUIDING IN NETWORK VIRTUAL EXPERIMENT

ABSTRACT

As an important method used by network of teaching, virtual experiment is a great complementarity in network teaching. Meanwhile, because of the particularity of network teaching, students couldn't get any help from their teachers when they get stuck anytime, which is now the bottle-neck of network teaching promotion.

The paper made research in difficulty in computer network virtual experiment development, it figured out two problems: one is how to make intellectual guiding, and the other is web development based on J2EE, integrating expert system.

The paper described the effort on the demand analysis, design and development of intellectual guiding system. The system is based on speculative knowledge on expert system and rule-engine, based on J2EE development skills (Struts, Spring, Hibernate), the system is divided into several modules. On the basis of original virtual experiment system, new intellectual guiding system is developed. By the diversion of original system, the development cut down the modification of original system.

The paper completed the design and development on computer network intellectual guiding system. After testing, the system has been used in teaching practice. The practice has proved that the mentality of intellectual guiding by expert system.

KEY WORDS: Intelligent-guiding, J2EE, Expert System, Rules Engine

目 录

摘 要.....	II
ABSTRACT.....	III
目 录.....	IV
第一章 绪论.....	- 1 -
1.1 论文的研究背景.....	- 1 -
1.1.1 远程教育的发展现状.....	- 1 -
1.1.2 网络虚拟实验的发展.....	- 2 -
1.1.3 项目背景.....	- 2 -
1.2 论文的研究内容、意义及创新点.....	- 3 -
1.2.1 研究内容.....	- 3 -
1.2.2 研究意义.....	- 3 -
1.2.3 创新点.....	- 3 -
第二章 对关键技术的研究.....	- 4 -
2.1 专家系统.....	- 4 -
2.2 规则引擎.....	- 5 -
2.2.1 规则引擎的构成及工作流程.....	- 5 -
2.2.2 规则引擎的知识表示.....	- 6 -
2.2.3 推理的控制策略.....	- 7 -
2.2.4 Rete 算法.....	- 10 -
2.2.5 规则引擎对比.....	- 11 -
2.2.6 Drools 规则引擎.....	- 12 -
2.3 Web 开发技术.....	- 14 -
2.3.1 表示层技术.....	- 14 -
2.3.2 业务层技术.....	- 16 -
2.3.3 持久层技术.....	- 18 -
2.3.4 Ajax 技术.....	- 19 -
2.3.5 JSON 格式.....	- 20 -
2.4 Java Applet 技术.....	- 21 -
第三章 对原有虚拟实验系统的描述.....	- 22 -
3.1 表示层.....	- 22 -
3.2 业务逻辑层.....	- 23 -
3.3 持久层.....	- 23 -
第四章 智能指导及运算符扩展需求分析.....	- 25 -
4.1 虚拟实验智能指导总体需求分析.....	- 25 -
4.1.1 基于 web 远程教学平台特点.....	- 25 -
4.1.2 智能指导系统总体需求概述.....	- 26 -
4.2 智能指导模块分析.....	- 27 -
4.2.1 知识库维护模块.....	- 28 -
4.2.2 学生实验智能指导推理模块.....	- 29 -
4.3 转换器分析.....	- 30 -
4.3.1 类实体转换器.....	- 30 -

4.3.2 规则转换器	- 31 -
第五章 网络虚拟实验智能指导设计	- 32 -
5.1 智能指导流程设计	- 32 -
5.2 系统模块设计	- 33 -
5.3 知识库维护模块	- 34 -
5.3.1 知识库维护模块类设计	- 34 -
5.3.2 知识库维护模块数据库设计	- 36 -
5.3 智能指导推理模块	- 37 -
5.3.1 智能指导推理模块类设计	- 38 -
5.3.2 智能指导推理模块数据库设计	- 39 -
5.3.3 规则实体的设计	- 39 -
5.3.4 运算符扩展功能设计	- 41 -
5.4 学生实验模块	- 42 -
第六章 网络虚拟实验智能指导系统实现及扩展	- 43 -
6.1 知识库维护模块的实现	- 43 -
6.1.1 知识库维护界面实现	- 43 -
6.1.2 知识库维护实现难点要点	- 45 -
6.2 智能指导推理模块实现	- 46 -
6.2.1 推理模块界面实现	- 46 -
6.2.2 推理模块实现的重点难点	- 47 -
6.3 运算符扩展	- 51 -
第七章 网络虚拟实验智能指导的不足与发展	- 53 -
第八章 论文工作总结	- 54 -
参考文献	- 55 -
致 谢	- 56 -
作者攻读学位期间发表的学术论文目录	- 57 -

第一章 绪论

1.1 论文的研究背景

1.1.1 远程教育的发展现状

远程教育现在多指现代远程教育也称为网络大学,它是综合运用现代通信技术、多媒体计算机技术和现代网络技术,特别是因特网技术实现交互式学习的新型教育模式,它是伴随着现代信息技术的发展而产生的一种新型教育方式。

远程教育在中国的发展经历了三代:第一代是函授教育,这一方式为我国培养了许多人才,但是函授教育具有较大的局限性;第二代是20世纪80年代兴起的广播电视教育,我国拥有全世界最大的广播电视大学;20世纪90年代,随着信息和网络技术的发展,产生了以计算机技术、网络技术和通信技术为基础的第三代远程教育,即现代远程教育。

我国现代远程教育的试点高校已由最初的4所发展为目前的67所,在读学生由开始试点时的3000多人扩展到今天的上百万人^[1]。已开设10个大类60多个专业,组织了近400多门课程和一批素材库、案例库,网络教育的覆盖面已达31个省、自治区和直辖市。迄今为止的230万网络学院的学生中90%为在职人员,不需要承担分配的压力。网络教育具有大众普及性特点,其开放性为不同基础、不同经历的受教育者提供了学习的机会;同时它的灵活性,也为不同背景、不同职业的受教育者接受继续教育提供了可能。

毫无疑问,现代远程教育对教育领域的发展将产生深厚的影响。随着互联网技术的发展,网上教学作为远程教育的一种重要手段,将会有广阔的发展空间。但就目前尚未成熟的中国远程教育来说,目前还存在以下缺陷:

- 现代远程教育的质量仍需建设。远程教育的师资力量薄弱,教学内容单一,课件贫乏。
- 培养学生动手实践能力受到制约。由于远程教育手段的特殊性,导致对学生动手能力的培养存在先天的缺陷。
- 教育是一个双向互动的教学过程,需要加强远程教学过程中的互动性,单一的填鸭式教学已经不能满足现代教育的质量要求。

1.1.2 网络虚拟实验的发展

现代远程教育,其突出特征是在网络环境下,以现代教育思想和学习理论为指导,充分发挥网络的各种教育功能和丰富的网络教育资源优势,向教育者和学习者提供一种网络教和学的环境,传递数字化内容,开展以学习者为中心的非面授教育活动。网络教学活动在互联网上的虚拟世界进行。它利用虚拟技术模拟出实际并不存在但又和现实相近的网上校园、网上教学、网上图书馆、网上阅览室、网上实验室、网上实习基地等虚拟现实社会环境。虚拟的环境,不仅可以逼真、仿真,还可以超真,将一些现实中没有的或极难仿真的情景模拟再现,在广度和深度上极大地丰富了教学活动的内涵。

同传统的实验室实验相比,网络虚拟实验降低了学生实验对物理条件的物质要求。学生在实验室做实验,要考虑到实验成本,人身安全等一系列要素,甚至对于一些极端的实验,学生无法亲手去尝试,极大的限制了学生的动手学习能力。相比较而言,虚拟实验通过使用计算机软件对传统的实验进行模拟,在个人计算机上就可以随意的尝试自己的想象,提高了实验效率。虚拟实验已经逐步发展成为远程教育的一种必要手段。

然而,虚拟实验虽然可以摆脱传统实验对客观条件的一些限制,远程教学中存在的一些弊端在虚拟试验中也逐步显现出来。在对已有的虚拟实验软件的一些调查中我们发现,学生一些常规的试验过程中缺乏指导,由于远程教育的分布性以及学生实验的随时性,教师不可能随时在线对学生的实验做出指导,甚至学生由于时间等一些条件的限制,在做完实验后仍不知道自己的实验操作正确与否,丧失了教学价值。因此,如何针对学生的实验做出指导成为虚拟实验研究的一项重要课题。

1.1.3 项目背景

本论文是在国家十一五科技攻关项目“虚拟实验智能指导与管理系统的研发”(编号 2008BAH29B04)的研究成果的基础上,北京邮电大学网络教育技术研究所虚拟实验实验室针对虚拟实验教学中发现的实验指导困难的问题开展的深入研究。项目是在原有智能批改系统的基础上进行的深入开发,借助智能批改系统的专家库,对学生的实验进行智能指导开发。

1.2 论文的研究内容、意义及创新点

1.2.1 研究内容

虚拟实验系统在教育领域的广泛应用对网络教育的发展起着巨大的推动作用，然而在学生的实验过程中，缺乏教师的实时指导，甚至学生在做完实验后，无法知道实验究竟错在哪。智能指导功能的开发研究，实现对学生的指导，提高虚拟实验的教学质量。

本论文的研究内容包括：

- 专家系统，包括知识库、规则引擎，其中知识库包括概念库和规则库，而规则引擎则采用成熟的开源引擎 Drools；
- 虚拟实验智能批改的模式探索与系统集成；
- WEB 框架技术，包括 struts、spring、hibernate 等，保证系统的扩展性；
- JAPPLET 插件技术，虚拟实验依托于 JAPPLET 作为实验平台，通过 JAPPLET 技术，提供给实验者良好的人机交互界面；

1.2.2 研究意义

从学生虚拟实验的开发完成以来，对于学生在实验中存在的问题无法做出实时的指导，已经成为提高教学质量的一个瓶颈，对于智能指导的开发，存在以下意义：

1、实时指导：学生在实验过程中遇到无法理解的操作或者不知道如何进行实验的时候，智能指导系统都可以给学生做出相应的指导。

2、即时反馈：对于学生没把握的步骤，学生可以先进行尝试然后得到系统的反馈指导，进而加深对该操作的学习理解。

1.2.3 创新点

专家系统作为新兴的技术，在我国的推广才刚开始不久，现在的各种专家系统的应用，还都处于摸索阶段，而将专家系统引入到虚拟实验教学平台智能指导上的应用，更是少之又少，因此本系统是专家系统在虚拟实验智能指导的一次成功尝试和应用。

第二章 对关键技术的研究

2.1 专家系统

专家系统是一个智能计算机程序系统，其内部含有大量的某个领域专家水平的知识与经验，能够利用人类专家的知识和解决问题的方法来处理该领域问题。也就是说，专家系统是一个具有大量的专门知识与经验的程序系统，它应用人工智能技术和计算机技术，根据某领域一个或多个专家提供的知识和经验，进行推理和判断，模拟人类专家的决策过程，以便解决那些需要人类专家处理的复杂问题，简而言之，专家系统是一种模拟人类专家解决领域问题的计算机程序系统。对于计算机网络实验智能指导功能的开发，就是通过各种途径吸收专家的经验并进行组织整理，供教师学生实验使用。

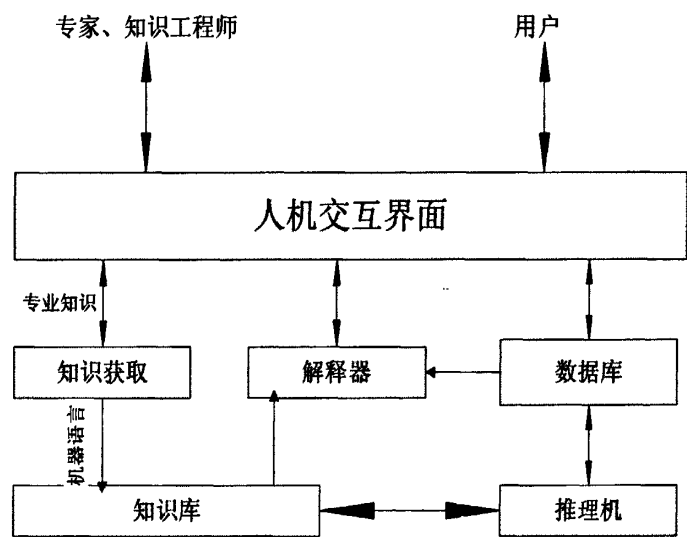


图 2-1: 专家系统结构图

专家系统的基本结构如图所示，其中箭头表示数据流动的方向。专家系统通常由以下 6 部分构成^[2]：

知识库：知识库用来存放专家知识，包括产生式规则、语义网、框架、脚本、逻辑、概念组等。专家系统的工作就是根据知识库的知识模拟专家的思维模式进行推理，所以知识库的质量直接决定了专家系统的优劣。在本系统中，专家库的

知识一般都是由教师直接制定标准答案。

推理机：推理机是根据当前条件或已知信息，反复根据知识库中的规则进行推理，获得最准确的结论。推理方式主要有前向和后向推理两种。

数据库：数据库又称事实库，即被推理机用来匹配规则时所使用的事实的全局数据库。在虚拟实验的智能批改中的具体体现是学生的实验脚本，这是推理机进行推理的基本素材。

解释器：解释器负责将系统的推理介绍给客户，在基于规则的系统中这些推理触发的规则都会存在一个栈或者列表中。

知识获取：知识获取只要负责将专家的系统知识经编译向系统提供能被机器识别的语言。

人机交互界面：人机交互界面是用户、专家和系统之间的交互机制。

2.2 规则引擎

Java 规则引擎是推理引擎的一种,它起源于基于规则的专家系统。专家系统是人工智能的一个分支,它模仿人类的推理方式,使用试探性的方法进行推理,并使用人类能理解的术语解释和证明它的推理结论。下面笔者将从规则引擎的构成阐述规则引擎的工作原理^[3]。

2.2.1 规则引擎的构成及工作流程

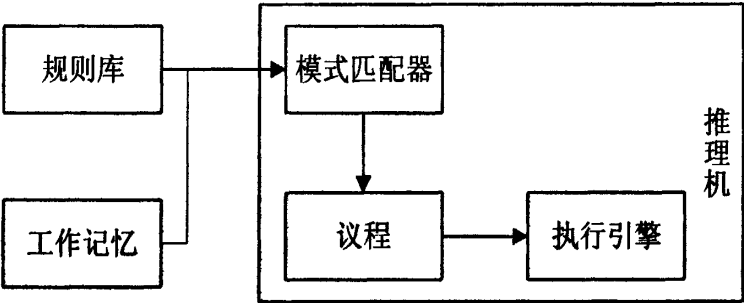


图 2-2: 规则引擎构成

如上图所示，一个基本的规则引擎主要包括三大块：知识库，工作事实和推理机，其中推理机又包含模式匹配器、议程和执行引擎三部分。

规则库是中心知识库，存储各类模拟人类问题求解的产生式规则。一个完整的规则包括前件部分和后件部分：

前件(Antecedent)又称条件部分、模式部分或左部(Left — Hand — Side, LHS), 是规则触发的条件。单独的条件称为条件元素或一个模式;

后件(Consequent)又称右部(Right — Hand — Side, RHS), 是规则触发时将要执行的一系列行为。

工作记忆又称事实, 是应用于规则的事实的全局数据库, 它保存系统的当前状态。

推理机(Inference engine) :通过决定哪些规则满足事实或目标, 并授予规则优先级, 满足事实或目标的规则被加入议程^[4]。其中, 推理机又由以下三部分构成: 模式匹配器: 事实库中可能包含数千个事实, 规则库中的每条规则又可能有两三个前提, 推理引擎必须在很短的时间内将事实与对应的规则匹配起来不是一件容易的事。好在这方面的研究已经非常成熟了, 很多模式匹配器都可以在极短的时间内完成匹配。尽管模式匹配仍然是基于规则的应用中最耗时的操作, 但它的效率实际上比大多数人想象的要高得多。

议程: 规则之间可能存在冲突, 此时就需要由调度器来决定采用哪条规则。通常规则可以有优先级的设置, 调度器将根据优先级来解决冲突。

执行引擎: 最后规则引擎需要执行规则的行为, 执行引擎就负责这项工作。在早期经典的基于规则的系统中规则引擎只能增、删、改事实库中的事实, 除此之外不能做任何事。但现代的规则引擎通常放宽了限制。

经过上述部分对规则引擎的构成分析, 接下来笔者通过一个简单的规则引擎工作流程更清晰的表述规则引擎的工作原理:

- a) 将初始数据输入至工作内存
- b) 使用模式匹配将规则库中的规则和数据进行比较
- c) 如果执行规则存在冲突, 就同时激活多个规则, 将冲突的规则放入冲突集合
- d) 解决冲突, 将激活的规则按顺序放入议程
- e) 执行规则
- f) 重复步骤 b 至 e, 直到执行完议程中的所有规则

2.2.2 规则引擎的知识表示

规则引擎采用产生式规则作为基本的知识表达式, 本系统采用 Drools 的语法结构, 规则引擎的知识表示如下^[6]:

1. 产生式规则

产生式规则是对商业知识的编码。产生式规则的一般形式如下所示:

rule “<name>”

<attribute><value>

when

<LHS>

then

<RHS>

End

每一条规则都是由一系列的属性、一个 Left Hand Side (LHS) 和一个 Right Hand Side (RHS) 组成的。

每条规则的 LHS 由条件元素和列组成。这样可以方便的用一阶逻辑或命名逻辑进行表述。列通常用来表示对一个事实的域约束。

规则 LHS 是由一个或多个条件组成。当所有的条件都满足时, RHS 将被执行。这很类似于程序中的 if 语句。

在规则的语义识别上, 当且仅当 LHS 被匹配时才能激发规则-RHS。规则可以通过 package 关键字同一个命名空间相关联。一个包中声明了类引用、全局变量、函数和规则等。

规则引擎实现了数据同逻辑的完全解耦。规则并不能被直接调用, 因为他们不是方法或函数。规则的激发是对工作记忆中数据变化的响应。同时, 规则语言又被声明成像 java 语言一样的强制性语言, 它描述了“什么”而不是“如何”。

2. 一阶逻辑

规则是用一阶逻辑写成的, 又称谓词逻辑, 它是命题逻辑的扩展。

一个命题就是能够用“真”和“假”来表示的一个陈述句。如果一个词语可以单独表达“真”, 就可以称这个语句是闭合的语句。从程序的角度看, 就是说这个表达式中不包含任何变量。

如果一个语句表达式包含变量, 这个语句就被称为开放的语句。在确定变量的值是什么之前, 我们无法判断这个语句是“真”还是“假”。

在 Java 中, 一个标准的命题是由变量、运算符和数值组成的。“数值”通常可以被理解成字面上说的数值, 因此, 一个命题也被称作一个域约束。命题还可以进一步结合连续性或转折性的连接词, 从逻辑角度讲便是“&&”和“||”。

2.2.3 推理的控制策略

和人类的思维相对应, 推理引擎存在三种推理方式, 分别是正向推理, 反向推理和混合推理^[6]。

正向推理是由已知事实出发向结论方向的推理, 也称为事实驱动推理, 正向

推理的基本思想是：系统根据用户提供的初始事实，在知识库中搜索能与之匹配的规则，构成可适用的规则集，然后按某种冲突解决策略从规则集中选择进行推理，并将推出的结论作为中间结果加入到数据库中作为下一步推理的事实，如此重复这一过程，直到得出最终结论或者知识库中没有可适用的知识为止。

正向推理简单，易实现，但目的性不强，效率低，需要用启发性知识解除冲突并控制中间结果的选取。

正向推理的过程如图所示：

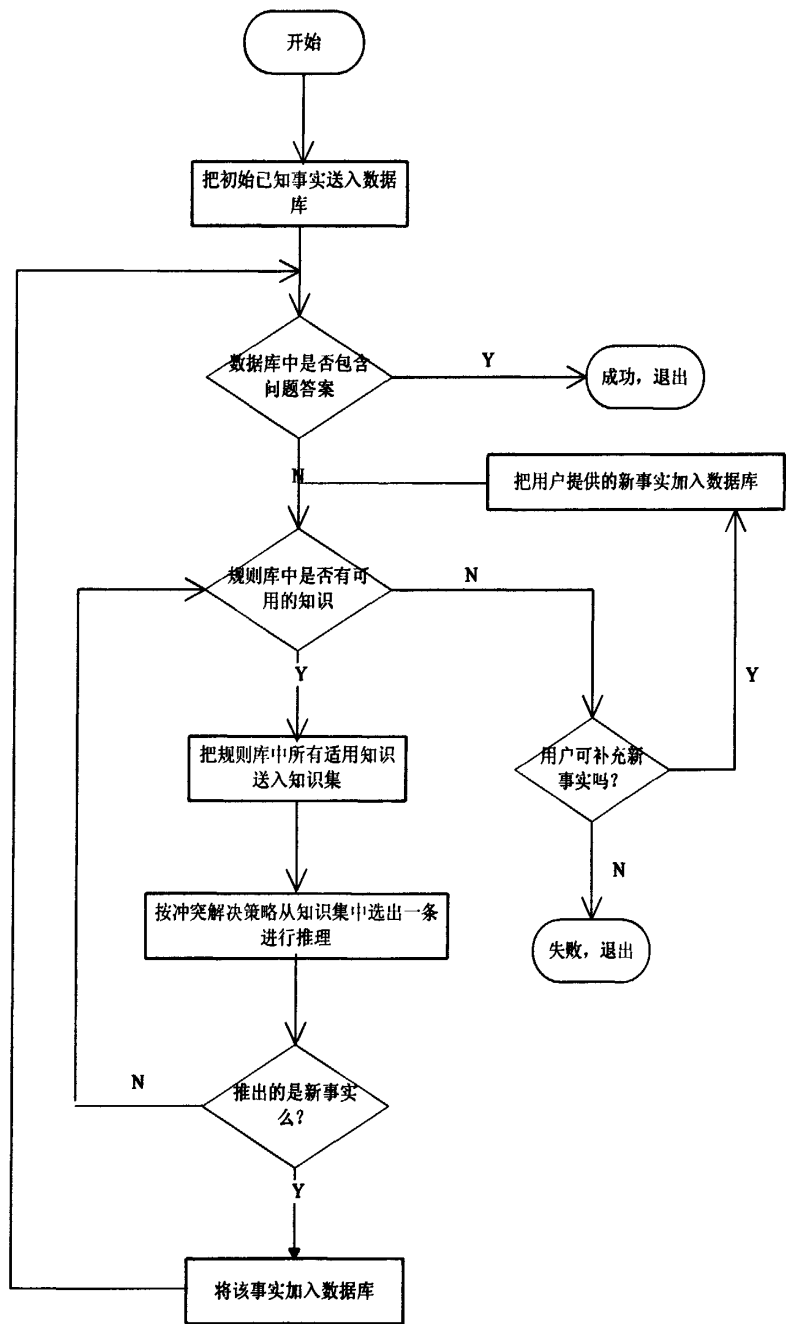


图 2-3: 正向推理流程图

反向推理是以某个假设目标作为出发点的一种推理，又称为目标驱动推理或逆向推理。反向推理的基本思想是：首先提出一个假设目标，然后由此出发，进一步寻找支持该假设的证据，若所需的证据都能找到，则该假设成立，推理成功；若无法找到支持该假设的所有证据，则说明该假设不成立，需要另作新假设。

与正向推理相比，反向推理的主要优点是不必使用与目标无关的知识，目的性强，同时他还有利于向用户提供解释。反向推理的缺点是，在选择初始目标时具有很大的盲目性，若假设不正确，就有可能需要多次假设，影响系统的效率。

反向推理的过程如图所示：

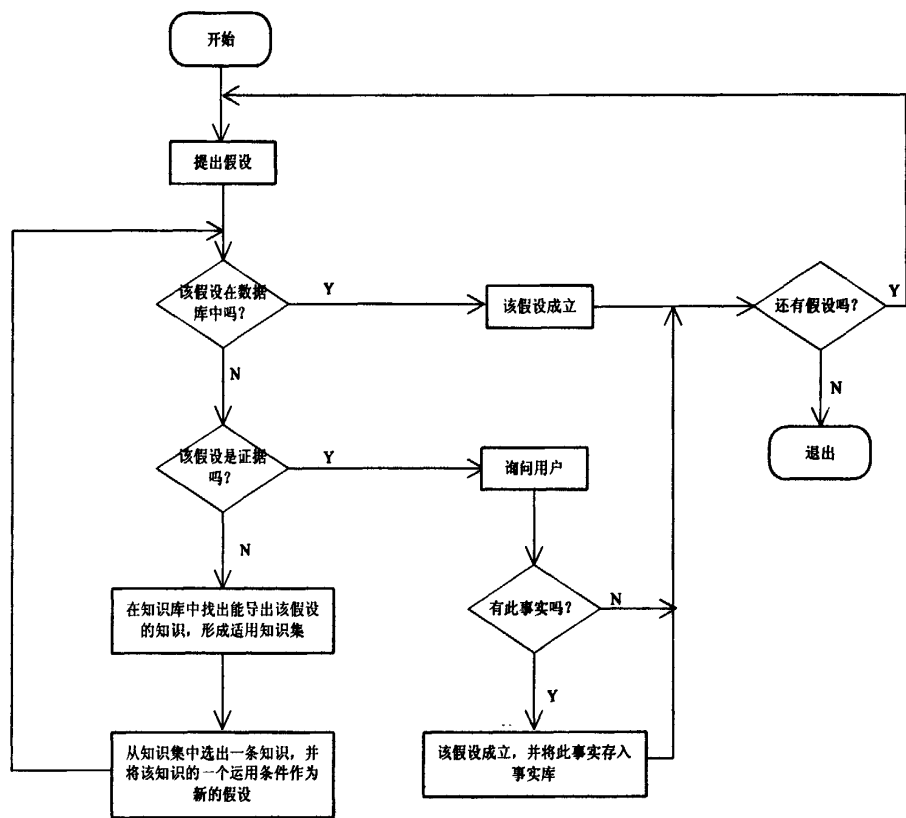


图 2-4: 反向推理流程图

混合推理的一般过程是：先根据初始事实进行正向推理以帮助提出假设，再用反向推理进一步寻找支持假设的证据，反复这个过程，直到得出结论为止。

混合推理集中了正向推理和反向推理的优点，但其控制策略相对复杂。

混合推理的过程如图所示：

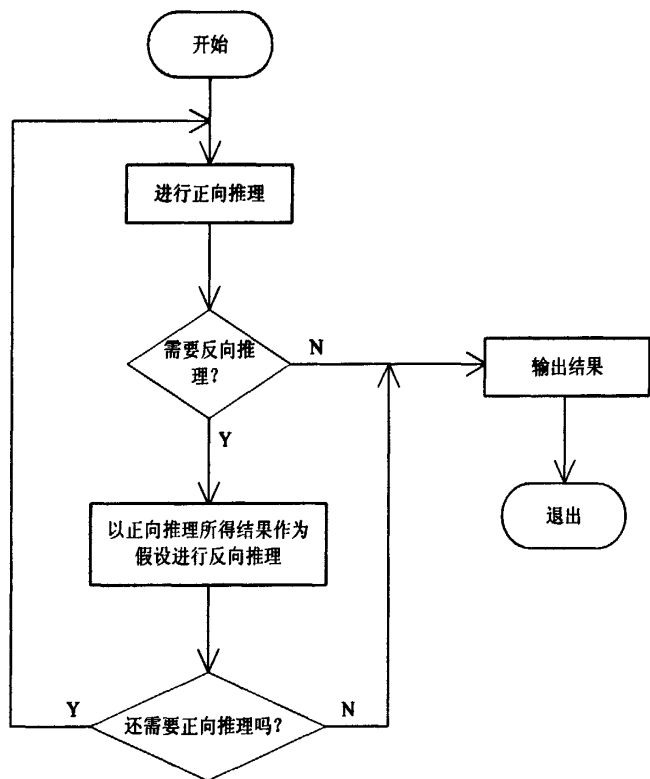


图 2-5: 混合推理流程图

2.2.4 Rete 算法

Rete 算法是一个用来实现产生式规则系统的高效模式匹配算法。该算法是由卡内基美隆大学的 Charles L. Forgy 在 1974 年发表的论文中所阐述的算法。Rete 算法提供了专家系统的一个高效实现。规则推理引擎做为产生式系统的一部分，当进行事实的断言时，包含三个阶段：匹配、选择和执行，称做 match- select- act cycle。Rete 算法可以对匹配阶段进行高效实现。

Rete 算法是目前效率最高的一个前项链推理算法^[7]，其核心思想是将分离的匹配项，根据内容动态构造成匹配树，以达到显著降低计算量的效果。

Rete 模式匹配算法是在模式匹配中利用推理机的时间冗余性和规则结构的相似性，通过保存中间过程及模式共享来提高推理效率的一种模式匹配算法。在模式匹配过程中，规则的前提(LHS)中可能会有很多相同的模块，因此在匹配规则的前提下，将进行大量的重复计算，这样就带来了时间冗余性的问题。例如：

Rule 1:if (A>B) and D or C then E = 100

Rule 2:if (A>B) and (B<C) then E=200

Rule 3:if(!(A>B) or (B<C)) then E=300

若要匹配这三条规则，对于表达式 A>B 要进行三次计算，对 B<C 需要进行

两次计算。Rete 采用的方法为：令 $M1 = A > B$, $M2 = B < C$ ；规则就可以修改为如下：

Rule 1: if (M1) and D or C then E=100

Rule 2: if (M1) and (M2) then E=200

Rule 3: if(!(M1) or (M2)) then E=300

这样只有当 A 或者 B 发生变化时，才重新计算 M1，同样当 B 或 C 发生变化时，重新计算 M2。这样就避免了在每次进行模式匹配都重复计算相同的表达式，而只要检测相关参数是否变化来决定是否更新表达式，在推理过程中可以节省大量的计算时间开销，提高推理效率。

简单来说，Rete 算法的实现是通过存储不断循环中匹配过程的状态，并且，只重新计算在事实列表中发生了变化，又反映到本次状态中的变化。也就是说，如果在一次执行周期中，一组模式找到三个所需事实中的两个，那么在下一周期中，就无需对已经找到的这两个事实进行检查，只有第三个事实才是需要关注的。仅当添加或删除事实的时候，匹配过程的状态才被更新。如果添加，删除事实的数量与事实和模式的总数相比很小，那么匹配过程会很快进行。最坏的情况是。如果所有的事实都改变了，那么所有的事实将与所有的模式进行匹配。如果仅是事实列表进行更新，那么每条规则必须记住已与之匹配的事实。也就是说，如果一个新的事实与规则的第三个模式相匹配，那么头两个模式匹配信息必须存在以完成匹配过程。这种状态信息指出了与某一规则中前面的模式相匹配的事实，它以规则的第一个模式为开始，以任一模式为结束。因此，一条有三个模式的规则对第一个模式，第一和第二个模式及第一，第二，第三模式都有部分匹配。一条规则的所有模式的部分匹配也是一个激活。另一种存储的状态信息称为模式匹配。当一个事实满足了任一规则中的单个模式而不需考虑在其他模式中可能会限制匹配过程的变量时，则出现的就是模式匹配。

从根本上来讲，RETE 算法的基本思想是保存匹配过程中的所有信息，以存储的代价换取算法的执行效率。对每一个过程，附加一个列表记录工作中所有能匹配的事实。每一个事实加入到工作记忆时，算法列出记录中所有能与之匹配的模式，并把这个事实加入到匹配事实表中；当一个事实从工作记忆删除时，同样找出与之匹配的模式，将该事实从匹配事实表删除^[8]。

2.2.5 规则引擎对比

目前主流的规则引擎组件多是基于 Java 和 C++ 程序语言环境，已经有多种 Java 规则引擎商业产品与开源项目的实现，其中有的已经支持 JSR94，有的正朝

这个方向做出努力，目前主流的规则引擎有以下几种：

Jess(Java Expert Shell System)：通用推理引擎，空间利用率低。**JESS** 可能是最成熟的 **Java** 规则引擎，有很多好的工具(包括 **Eclipse** 插件)和文档。然而它是一个商业产品，并且，规则是用类 **CLIPS** 的描述语言写的，比较晦涩难懂。

Drools - Drools 规则引擎应用 **Rete** 算法的改进形式 **Rete-II** 算法。从内部机制上讲，它使用了和 **Forgy** 的算法相同的概念和方法，但是增加了可与面向对象语言无缝连接的节点类型。**Drools** 使用像 **Java** 式的语法，并且有最开放的许可，同时，它是完全免费的。**Drools** 规则引擎应用 **RETE** 算法的改进形式的 **RETE-II** 算法，该算法的优点是系统不会随着规则数目的增加而导致性能方面的明显下降；缺点是牺牲系统的存储空间。

Mandarax 基于反向推理（归纳法）。能够较容易地实现多个数据源的集成。例如，数据库记录能方便地集成为事实集(facts sets)，**reflection** 用来集成对象模型中的功能。目前不支持 **JSR 94**。

2.2.6 Drools 规则引擎

Drools是用**Java**语言编写的开放源码规则引擎，实现了**JSR 94 Rule Engine** API并提供了单元测试代码。应用了**Rete**核心算法。**Drools**提供了声明式程序设计，并且使用域描述语言（DSL）来描述用户问题域^[9]。

Drools的工作流程可以分为两个部分：编制和运行。

编制的过程主要包括为规则建立**DRL**或**XML**规则文件，将文件传入解析器中对规则语法的正确性进行检查并为规则描述建立一个中间结构。**AST**然后将规则描述传入**Package Builder**中打包，**Package Builder**同时负责打包用到的所有代码产生器和编译器。

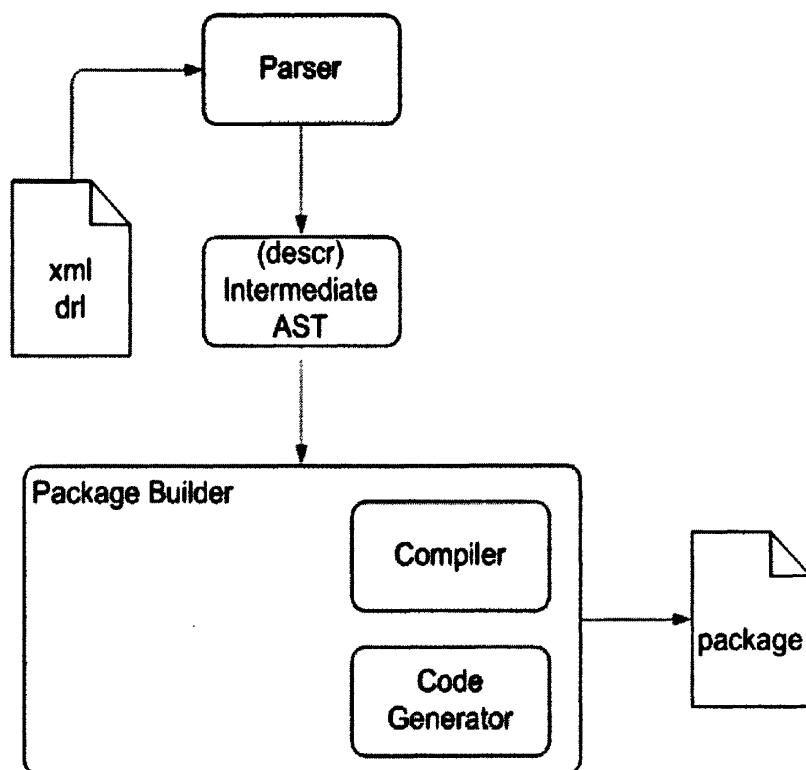


图2-6: 编制时流程图

RuleBase是运行时组件，包含一个或多个Package，任何时刻都可以向RuleBase中添加或删除Package。

一个 RuleBase 可以同时初始化多个 Working Memory，在其间维护一个弱引用，除非重新进行配置。Working Memory 包含许多子组件，如 Working Memory Event Support(事件支持)，Truth Maintenance System(真值维护系统)，Agenda 和 Agenda Event Support(事件支持)。向 Working Memory 中设置对象的工作可能要在建立了一个或多个激活的规则后才结束。Agenda 负有规划激活规则运行的责任。

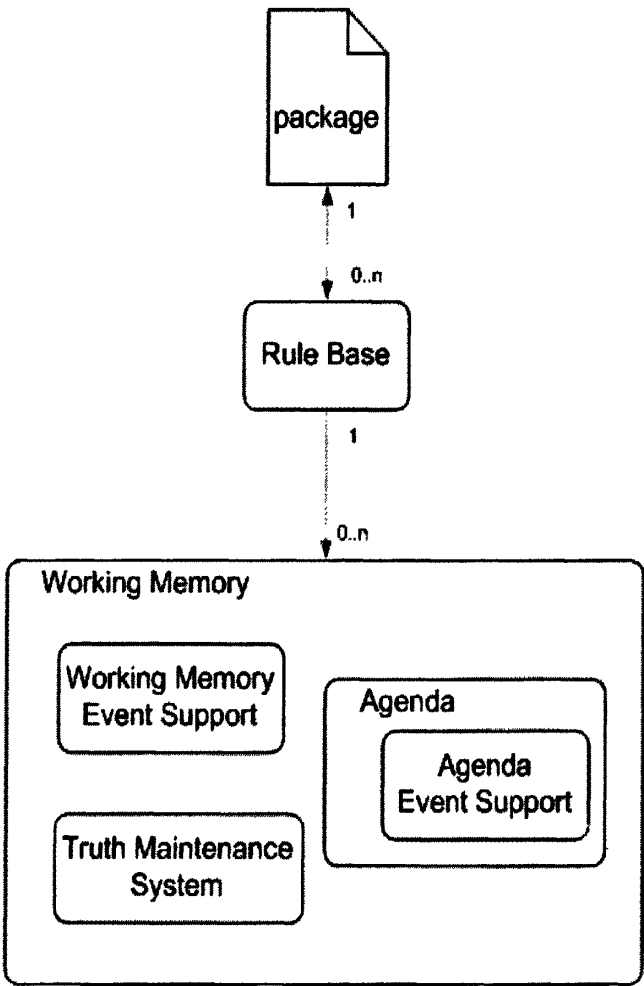


图 2-7：运行时

2.3 Web 开发技术

大多数 Web 开发应用都将分为三个层次，分别是表示层、业务层和持久层。

2.3.1 表示层技术

从实际应用开发的角度而言，Struts 应该是表示层开发技术的第一选择。

Struts 是 Apache 软件基金组织 Jakarta 项目的一个子项目，在英文中，Struts 是“支架、支撑”的意思，这表明了 Struts 在 Web 开发中的巨大作用，采用 Struts 可以更好的遵循 MVC 模式，同时，Struts 提供了一套完整的规范，可以充分利用 JSP/Servlet 的优点，减少程序员的工作量，具有很强的扩展性。

除此之外，Struts 的优点主要集中体现在两个方面：Taglib 和页面导航。Taglib

是 Struts 的标记库，灵活动用，能大大提高开发效率。另外，就目前国内 JSP 开发者而言，除了使用 JSP 自带的常用标记外，很少开发自己的标记，或许 Struts 是一个很好的起点。

关于页面导航，将是今后的一个发展方向，事实上，这样做，使系统的脉络更加清晰。通过一个配置文件，即可把握整个系统各部分之间的联系，这对于后期的维护有着莫大的好处。尤其是当另一批开发者接手这个项目时，这种优势体现得更加明显^[10]。

Struts 体系结构实现了 MVC 设计模式的概念，它将 Model、View 和 Controller 分别映射到 Web 应用中的组件。Model 由代表系统状态和商业逻辑 Action 来构建，View 是由 JSP 和 struts 提供的自定义标签来实现，Controller 负责控制流程，由 ActionServlet 和 ActionMapping 来完成。

就 MVC 的 Struts 实现来说：

1. 视图：视图主要由 JSP 来实现，同时 Struts 自身包含了可扩展的自定义标签库，用户界面的创建过程更加简便。
2. 模型：模型主要表示系统的业务逻辑和状态，在 Struts 中业务逻辑一般由 JavaBean 实现，而系统的状态则是由 ActionBean 来实现。
3. 控制器：控制器主要由 ActionServlet 和 Action 类实现，ActionServlet 是 Servlet 的一个具体实现，由 Struts 提供，是 Struts 框架的核心组件，负责接收 HTTP 的请求信息，把请求转发给 Action，Action 负责调用模型，实现逻辑控制。
4. 配置文件 Struts-config.xml：当 ActionServlet 接收到 HTTP 的请求信息时，就需要读取 Struts-config 的配置信息来决定将请求信息发送给哪个 Action 对象。Struts-config 文件就是描述用户请求路径和 Action 映射关系的配置信息，在该配置文件中，每一个 Action 的映射信息都通过一个 action 元素来配置，这些配置信息会在系统启动时就被读取，供 Struts 运行期间使用。

Struts 的具体实现如图所示：

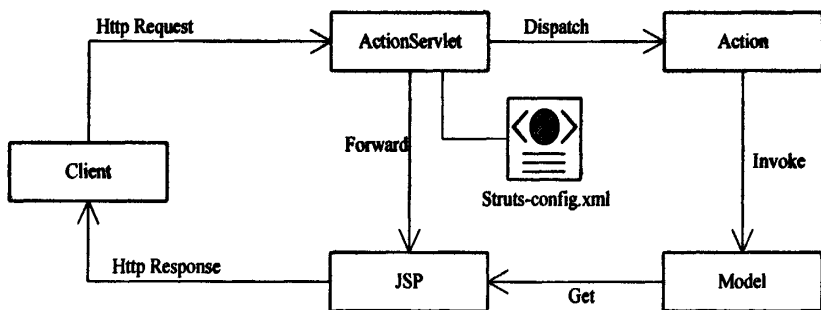


图 2-8: Struts 流程图

2.3.2 业务层技术

Spring 为企业应用的开发提供了一个轻量级的解决方案。该解决方案包括：基于依赖注入的核心机制，基于 AOP 的声明式事务管理与多种持久层技术的整合，以及优秀的 Web MVC 框架等。

Spring 框架是一个分层架构，由 7 个定义良好的模块组成。Spring 模块构建在核心容器之上，核心容器定义了创建、配置和管理 bean 的方式。

组成 Spring 框架的每个模块（或组件）都可以单独存在，或者与其他一个或多个模块联合实现。每个模块的功能如下^[11]：

核心容器：核心容器提供 Spring 框架的基本功能。核心容器的主要组件是 BeanFactory，它是工厂模式的实现。BeanFactory 使用控制反转（IOC）模式将应用程序的配置和依赖性规范与实际的应用程序代码分开。

Spring 上下文：Spring 上下文是一个配置文件，向 Spring 框架提供上下文信息。Spring 上下文包括企业服务，例如 JNDI、EJB、电子邮件、国际化、校验和调度功能。

Spring AOP：通过配置管理特性，Spring AOP 模块直接将面向方面的编程功能集成到了 Spring 框架中。所以，可以很容易地使 Spring 框架管理的任何对象支持 AOP。Spring AOP 模块为基于 Spring 的应用程序中的对象提供了事务管理服务。通过使用 Spring AOP，不用依赖 EJB 组件，就可以将声明性事务管理集成到应用程序中。

Spring DAO：JDBC DAO 抽象层提供了有意义的异常层次结构，可用该结构来管理异常处理和不同数据库供应商抛出的错误消息。异常层次结构简化了错误处理，并且极大地降低了需要编写的异常代码数量（例如打开和关闭连接）。Spring DAO 的面向 JDBC 的异常遵从通用的 DAO 异常层次结构。

Spring ORM：Spring 框架插入了若干个 ORM 框架，从而提供了 ORM 的对象关系工具，其中包括 JDO、Hibernate 和 iBatis SQL Map。所有这些都遵从 Spring 的通用事务和 DAO 异常层次结构。

Spring Web 模块：Web 上下文模块建立在应用程序上下文模块之上，为基于 Web 的应用程序提供了上下文。所以，Spring 框架支持与 Jakarta Struts 的集成。Web 模块还简化了处理多部分请求以及将请求参数绑定到域对象的工作。

Spring MVC 框架：MVC 框架是一个全功能的构建 Web 应用程序的 MVC 实现。通过策略接口，MVC 框架变成高度可配置的，MVC 容纳了大量视图技术，其中包括 JSP、Velocity、Tiles、iText 和 POI。

Spring 框架的功能可以用在任何 J2EE 服务器中，大多数功能也适用于不受管理的环境。Spring 的核心要点是：支持不绑定到特定 J2EE 服务的可重用

业务和数据访问对象。毫无疑问，这样的对象可以在不同 J2EE 环境（Web 或 EJB）、独立应用程序、测试环境之间重用。

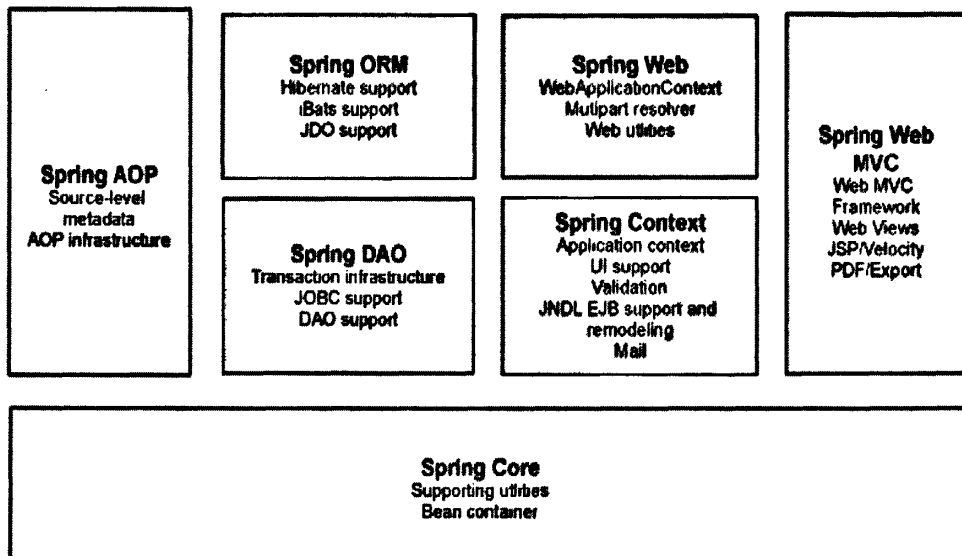


图 2-9 Spring 总体结构

Spring 的特征如下：

1、控制反转 (IOC=Inverse Of Control)，依赖注入 DI(Dependence injection)

不管是控制反转，还是依赖注入，其含义完全相同。当某个 Java 实例（调用者）需要另一个 Java 实例（被调用者）时，在传统的程序设计过程中，通常由调用者来创建被调用者的实例。而在依赖注入的模式下，创建被调用者的工作不再由调用者来完成，通常由 Spring 容器来完成，然后注入调用者。

2、对 bean 的管理

bean 是 Spring 管理的基本单位，在 Spring 的 J2EE 应用中，所有的组件都是 bean，bean 包括数据源、Hibernate 的 SessionFactory 及事务管理等。Spring 里的 bean 是非常广义的概念，任何的 Java 对象、Java 组件都可被当成 bean 处理。

总结起来，Spring 有如下优点：

- 低侵入式设计，代码污染极低
- 独立于各种应用服务器，可以真正实现 Write Once, Run Anywhere 的承诺
- Spring 的 DI 机制降低了业务对象替换的复杂性
- Spring 并不完全依赖于 Spring，开发者可以自由选用 Spring 框架的部分或全部

2.3.3 持久层技术

Hibernate 是一个开放源代码的对象关系映射框架,它对 JDBC 进行了非常轻量级的对象封装,使得 Java 程序员可以随心所欲的使用对象编程思维来操纵数据库。Hibernate 可以应用在任何使用 JDBC 的场合,既可以在 Java 的客户端程序使用,也可以在 Servlet/JSP 的 Web 应用中使用,最具革命意义的是, Hibernate 可以在应用 EJB 的 J2EE 架构中取代 CMP,完成数据持久化的重任^[12]。

Hibernate 可以将数据库资源映射为一个或者多个 POJO(普通的 JAVA 对象),将面向数据库资源的各种业务操作以 POJO 的属性和方法的形式出现,摆脱繁琐的 JDBC 代码,将精力更多地集中在业务方法的实现上。为了把数据访问细节和业务逻辑分开,可以把数据访问作为单独的持久化层。持久化层封装了数据访问细节,为业务逻辑层提供了面向对象的 API。完善的持久化层可以达到以下目标:

- (1)代码可重用性高,能够完成所有的数据库访问操作。
- (2)如果需要的话,能够支持多种数据库平台。
- (3)具有相对独立性,当持久化层的实现发生变化,不会影响上层的实现。

Hibernate 提供了强大、高性能的对象到关系数据库的持久化服务。利用 Hibernate,开发人员可以按照 Java 的基础语义(包括关联、继承、多态、组合及 Java 的集合架构)进行持久层开发。Hibernate 能消除那些针对特定数据库厂商的 SQL 代码,并且把结果集由表格式的形式转换成值对象的形式。Hibernate 不仅管理 Java 类到数据库表的映射(包括 Java 数据类型到 SQL 数据类型的映射),还提供数据查询和获取数据的方法,可以大幅度地减少在开发时人工使用 SQL 和 JDBC 处理数据的时间。Hibernate 提供的 HQL 是面向对象的查询语言,它在对象型数据和关系型数据库之间构建了一条快速、高效、便捷的沟通渠道。

Hibernate 的核心接口一共有 5 个,分别为: Session、SessionFactory、Transaction、Query 和 Configuration。这 5 个核心接口在任何开发中都会用到。通过这些接口,不仅可以对持久化对象进行存取,还能够进行事务控制。下面对这五个核心接口分别加以介绍^[13]。

- Session 接口: Session 接口负责执行被持久化对象的 CRUD 操作(CRUD 的任务是完成与数据库的交流,包含了很多常见的 SQL 语句)。但需要注意的是 Session 对象是非线程安全的。同时, Hibernate 的 session 不同于 JSP 应用中的 HttpSession。这里当使用 session 这个术语时,其实指的是 Hibernate 中的 session,而以后会将 HttpSession 对象称为用户 session。
- SessionFactory 接口: SessionFactory 接口负责初始化 Hibernate。它充当数据存储源的代理,并负责创建 Session 对象。这里用到了工厂模式。

需要注意的是 **SessionFactory** 并不是轻量级的, 因为一般情况下, 一个项目通常只需要一个 **SessionFactory** 就够, 当需要操作多个数据库时, 可以为每个数据库指定一个 **SessionFactory**。

- **Configuration 接口**: **Configuration** 接口负责配置并启动 **Hibernate**, 创建 **SessionFactory** 对象。在 **Hibernate** 的启动的过程中, **Configuration** 类的实例首先定位映射文档位置、读取配置, 然后创建 **SessionFactory** 对象。
- **Transaction 接口**: **Transaction** 接口负责事务相关的操作。它是可选的, 开发人员也可以设计编写自己的底层事务处理代码。
- **Query 和 Criteria 接口**: **Query** 和 **Criteria** 接口负责执行各种数据库查询。它可以使用 **HQL** 语言或 **SQL** 语句两种表达方式。

Hibernate 能在众多的 **ORM** 框架中脱颖而出, 因为 **Hibernate** 与其他 **ORM** 框架对比有如下优势:

- 开源和免费的 **License**, 方便需要时研究、改写代码并进行功能定制
- 轻量级封装, 避免引入过多复杂的问题, 调试容易, 减轻程序员负担
- 具有可扩展性, **API** 开发
- 开发者活跃, 产品有稳定的发展保障

2.3.4 Ajax 技术

Ajax是**Asynchronous JavaScript and XML**的缩写。实际上, **Ajax**并不是什么新兴技术, 而是由几种蓬勃发展的技术以新的方式组合而成的, 这几种技术分别如下^[14]:

- **XMLHttpRequest**

XMLHttpRequest是**Ajax**当中最重要、最核心的技术, 它最早是在**IE5**中通过一个名为**XMLHTTP**的**ActiveX**对象来实现的。**XMLHTTP**对象使得页面中的脚本具有了在不刷新页面的情况下直接与服务器端通信的能力。

- **XML**

XML具有一种开放的、可扩展的和可自描述的语言结构, 目前, 它已经成为因特网上数据和文档传输的标准。

- **DOM**

DOM是面向**HTML**和**XML**文档的**API**, 它为文档提供了结构化表示, 并定义了如何通过脚本来访问文档结构。

- **Javascript**

Javascript是一种在浏览器端大量使用的编程语言, 它可以使用浏览器中的很多对象, 包括**XMLHttpRequest**。

传统的 web 应用允许用户填写表单(form)，当提交表单时就向 web 服务器发送一个请求。服务器接收并处理传来的表单，然后返回一个新的网页。这个做法浪费了许多带宽，因为在前两个页面中的大部分 HTML 代码往往是相同的。由于每次应用的交互都需要向服务器发送请求，应用的响应时间就依赖于服务器的响应时间。这导致了用户界面的响应比本地应用慢得多。

与此不同，AJAX 应用可以仅向服务器发送并取回必需的数据，它使用 SOAP 或其它一些基于 XML 的 web service 接口，并在客户端采用 JavaScript 处理来自服务器的响应。因为在服务器和浏览器之间交换的数据大量减少，结果我们就能看到响应更快的应用。同时很多的处理工作可以在发出请求的客户端机器上完成，所以 Web 服务器的处理时间也减少了。

使用 Ajax 的最大优点，就是能在不更新整个页面的前提下维护数据。这使得 Web 应用程序更为迅捷地回应用户动作，并避免了在网络上发送那些没有改变过的信息。

2.3.5 JSON 格式

JSON (JavaScript Object Notation) 是一种轻量级的数据交换格式，它基于 Javascript 语言，是在 Javascript 的数组 (array) 和对象 (Object) 基础上发展而来。与 XML 相似，都是独立语言，在跨平台数据传输中有优势，但 XML 需要 DOM 来解析，而对于浏览器来说，不同厂商的支持的 XML 和 DOM 又有所不同，这给 WEB 开发者或多或少带来一些麻烦。JSON 采用完全独立于语言的文本格式，但是也使用了类似于 C 语言家族的习惯 (包括 C, C++, C#, Java, Javascript, Perl, Python 等)，这些特性使 JSON 称为理想的数据交换语言。

JSON 具体有以下几种形式^[15]:

1. 对象是一个无序的“‘名称/值’对”集合。一个对象以“{”开始，以“}”结束。每个“名称”后跟一个“:”，“‘名称/值’对”之间使用“,”分隔。
2. 数组是值 (value) 的有序集合。一个数组以 “[” 开始，“]” 结束。值之间使用 “,” 分隔。
3. 值 (value) 可以是双引号括起来的字符串 (String)、数值 (number)、true、false、null、对象 (object) 或者数组 (array)。这些结构可以嵌套。
4. 字符串 (string) 是由双引号包围的任意数量 Unicode 字符的集合，使用反斜体转义。一个字符 (character) 即一个单独的字符串 (character string)。

2.4 Java Applet 技术

Java Applet 就是用 Java 语言编写的这样的一些小应用程序,它们可以直接嵌入到网页中,并能够产生特殊的效果。包含 Applet 的网页被称为 Java-powered 页,可以称其为 Java 支持的网页。

当用户访问这样的网页时,Applet 被下载到用户的计算机上执行,但前提是用户使用的是支持 Java 的网络浏览器。由于 Applet 是在用户的计算机上执行的,因此它的执行速度不受网络带宽或者 Modem 存取速度的限制。用户可以更好地欣赏网页上 Applet 产生的多媒体效果。

在 Java Applet 中,可以实现图形绘制,字体和颜色控制,动画和声音的插入,人机交互及网络交流等功能。Applet 还提供了名为抽象窗口工具箱 (Abstract Window Toolkit, AWT) 的窗口环境开发工具。AWT 利用用户计算机的 GUI 元素,可以建立标准的图形用户界面,如窗口、按钮、滚动条等等。

第三章 对原有虚拟实验系统的描述

计算机网络虚拟实验智能指导系统是在原有计算机网络虚拟实验系统的基础上进行的二次开发，为方便读者更清楚的了解整个系统的开发过程，本章将简单介绍原有的虚拟实验系统框架结构。

在关键技术部分我们介绍过，大多数 Web 开发应用都将分为三个层次，分别是表示层、业务层和持久层。本系统采用的框架如图所示，表示层采用 Struts；业务层采用 Spring；而持久层则用 Hibernate。

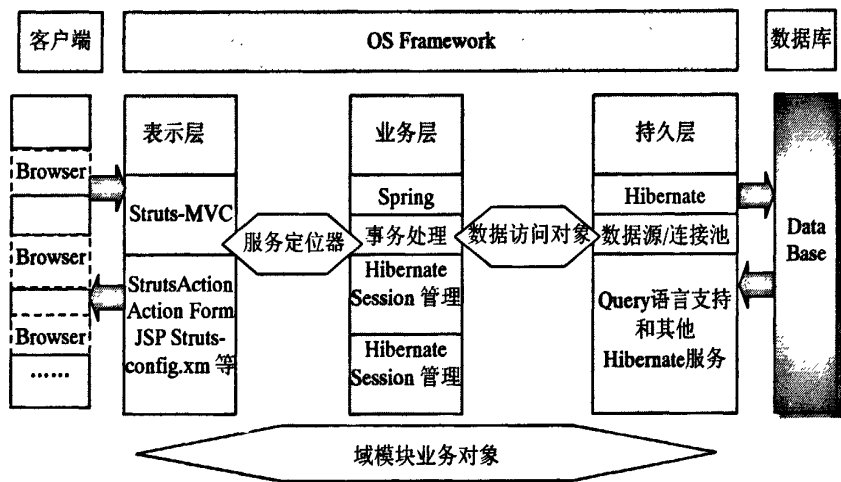


图 3-1: 虚拟实验系统框架图

3.1 表示层

表示层主要负责用户的交互，将系统信息展示给用户，并且从用户那里获得系统数据。主要包括用户界面的控制以及用户操作的权限控制等多个部分，表示层决定了系统界面的可用性及系统信息输入输出的可靠性。

表示层使用 Struts 框架实现，完成页面功能的跳转以及业务的处理等功能。表示层包含的类文件有 *Action 和 *Info 两种（*代表具体的 Action）。

在表示层中我们对 Struts 提供的 DispatchAction 文件扩展，创建 com.bupticet.base.web.action.BaseAction 作为我们的基类，然后针对每个模块的不同业务处理需求继承这个基类，创建相应的 *Action 类文件。Action 文件通过 Struts-config 文件从 ActionForm 中获取相应的信息，进行相关的处理操作后，调

用相应的业务逻辑组件。它们是用用户请求信息和相应的业务逻辑处理之间的桥梁。

*Info 类是指 VO (Value Object 值对象), 描述一个相应的实体, 可以与 PO (Persistent Object) 进行转换, 为各层的数据交换服务。

3.2 业务逻辑层

业务逻辑层负责应用程序的运行和处理流程, 完成业务逻辑, 决定客户的请求是否可以得到实现, 外在的表现是系统的具体功能, 它是系统的关键。失败的业务逻辑层设计可能导致用户端功能丢失, 功能实现不可靠等。业务逻辑层决定怎样对客户的输入数据进行处理, 利用相应的业务实现来完成用户所需的功能。

业务逻辑层包含了业务逻辑的接口 *Manager 及接口的实现类 *ManagerImpl。在 Manager 接口中, 我们定义了如 getObject(), saveObject() 等一系列各个模块业务服务对象所共有的方法, 在具体的实现类中, 调用持久层的相关类方法, 实现接口中定义的方法。

3.3 持久层

持久层负责系统数据的持久化。失败的持久层设计可能导致数据存储、读取的不可靠。持久层不负责业务处理, 只处理数据的处理。同时, 持久化并不单单只数据库的操作, 任何形式的持久化应该由持久层来完成。

在持久层设计中, 我们采用 DAO (Data Access Object) 模式来实现代码的分离设计。持久层包含接口 *Dao 及其实现类 *DaoImpl。Dao 中定义了业务逻辑层与数据库直接交互的各种方法, DaoImpl 文件实现了 Dao 接口中定义的方法。

系统的整体框架采用 Spring 与 Struts、Hibernate 的整合, 表示层 Action 通过依赖注入得到与 Manager 的依赖关系, 业务逻辑层的 ManagerImpl 通过依赖注入得到与 Dao 的依赖关系。

为了提高代码的重用率, 优化程序结构, 系统采用模板模式来设计业务层和持久层中的类, 采用树形结构来定义业务逻辑层和持久层的接口。在定义的接口中, 包含每个模块都会用到的方法, 然后在每个模块中, 定义子接口继承基本接口, 利用面向对象思想的重写来定义每个模块的特有方法, 使各个模块的业务逻辑层和持久层在公共方法上统一。

在表示层中,扩展 `DispatchAction` 得到的 `BaseAction` 作为各个模块的基类,然后在各个模块中继承 `BaseAction` 得到各个模块的 `Action` 类方法,在 `Action` 中重写 `BaseAction` 的各个方法来处理各个模块具体的功能需求。

在 `Manager` 接口中定义了获取、保存、删除对象的方法,在 `HibernateManager` 中增加了获得记录总数、分页等方法。`HibernateManager` 是各个模块 `Manager` 的父接口。

第四章 智能指导及运算符扩展需求分析

4.1 虚拟实验智能指导总体需求分析

虚拟实验系统是借助计算机仿真技术在计算机上营造可辅助、部分替代甚至全部替代传统实验各操作环节的相关硬件操作环境,实验者可以在 Inter 网上通过接近真实的人机交互页面完成实验,极大的减轻了教学硬件条件对学生实验带来的压力。虚拟实验的实现可以有效缓解很多高校在经费、场地、器材等方面普遍面临的困难和压力,而且开展网上虚拟实验能够突破传统实验对“时、空”的限制,无论是学生还是教师,都可以自由、无顾虑地随时随地上网进入虚拟实验室,操作仪器,进行各种实验,有助于提高教学质量。由于虚拟实验本身的可操作性和交互性,大大提高了远程教学的水平,逐渐成为远程教学的一种有效手段,受到各远程教学院校的青睐。

4.1.1 基于 web 远程教学平台特点

从根本上来讲,虚拟实验系统属于远程教学 web 平台的一种具体实现,通过对远程教学的相关理论分析,笔者认为远程教学应该具备以下能力[8]:

(1)远近结合 内外结合

通过该系统可以进行师生分离情况下的远程实验教学,也可以进行本地教学。通过该系统可以进行实验课堂内的实验教学,也可以进行课堂之外的学生自主实验学习。克服实验教学长期受到课堂、课时限制的困扰,实验教学在时间和空间上得到延伸。

(2)虚实结合

虚拟实验对于远程实验教学有着重要的作用,是系统中必不可少的一部分,可以通过系统对虚拟实验进行有效的管理。但模拟中的鼠标和键盘的操作不可能替代学生的实验技能训练,许多实验过程只有学生亲自动手才能逐步掌握,而且有的实验是不能模拟的,所以必须在实验教学中把虚拟和实际实验有效的结合,管理系统中也应该对实际实验进行管理,并通过系统实现两种实验有效的整合。

(3)开放性

系统用户量比较大,地理位置上比较分散,为了便于用户访问系统,系统基于 Web 进行开发设计,采用 B/S 的系统构架,具有很好的开放性。用户只要能上网,通过浏览器就能访问系统。

(4)集成性

网上实验是实验教学的一部分,但不是实验教学的全部,实验教学还包括实验设计、实验布置、实验报告、实验批改等等多个教学环节,系统应该集成多种实验教学功能环节。

(5)交互性

远程实验师生分离、学生分散的特点要求教学管理系统必须提供有效的信息交流平台,便于师生间的交互,也便于同学间的交流。

(6)系统与课程低耦合

系统作为实验教学的平台、教学管理的平台应该是通用的,系统和具体的实验课程插件的结合应该是松散的,这样才能在系统中进行多种实验课程的教学,便于系统中实验课程的扩充。

通过以上对 web 教学平台的分析,对比虚拟实验教学系统,虚拟实验教学系统在交互性上依然存在部分的不足,由于远程教学的特殊性,系统不可能保证学生在做实验的过程中遇到问题随时都可以请教老师,智能指导系统的开发,可以有效的解决远程教学系统的这一缺陷。利用专家系统丰富的知识库,随时对学生的疑问进行解答,减轻了教师的教学负担,同时也提高了学生的学习能力。

4.1.2 智能指导系统总体需求概述

虚拟实验智能指导系统的开发是基于已有的虚拟实验系统的二次开发,在原有系统中,通过课程管理模块对课程的管理,系统可以根据需要随时对系统课程进行更新,这样可以保证现有系统可以满足不同课程实验的需求。同时,虚拟实验系统的根本是实验,实验模块是整套系统的重点,教师可以在实验模块中发布实验、安排实验任务,学生也可以利用这个模块在线实验、复习实验、查看实验成绩。

在学生实验模块,由于学生实验“时、空”的随意性,学生可以随时随地在网络进行实验,然而系统却不可能要求教师随时随地都可以解答学生的疑问。学生在遇到问题时不能得到及时的解答很有可能就此形成知识的死角,同时,随着虚拟实验平台的推广,越来越多的学生开始使用,教师的答疑任务越来越重。虚拟实验智能指导系统的开发,可以有效的解决这两方面的矛盾:

- 虚拟实验智能指导系统利用丰富的专家知识库，可以针对学生不同的疑问进行解答，针对不同学生不同知识点做出不同指导，为学生个人量身定做，可以有效的提高学生的成绩。
- 虚拟实验系统依托于虚拟实验系统，可以实时解答学生的疑问，在学生的实验过程中可以随时针对学生的知识盲点进行解答，避免了学生由于找不到教师请教问题造成的知识死角。

4.2 智能指导模块分析

智能指导系统主要由两个具体模块组成，分别是负责专家库的知识库维护模块和智能指导学生实验推理模块。

智能指导系统主要有三种角色，分别是：知识库管理员、教师、学生。

- 知识库管理员主要负责知识库的维护。知识库的丰富程度直接决定专家系统的优劣，知识库管理员可以在知识库维护模块添加、删除、修改知识点。
- 教师主要负责实验的发布以及正确答案的维护。在学生实验推理模块，教师可以增加制定实验的标准答案、维护实验规则。
- 学生主要使用系统进行实验，包括典型实验和自主实验。典型实验主要是教师发布的实验任务，智能指导功能主要针对这种实验，自主实验是学生根据自己的学习自行设计的实验，随着知识库的逐步完善，智能指导功能可以向这种实验扩展。学生可以在线实验、复习实验、智能指导。

三种角色的用例图如下所示：

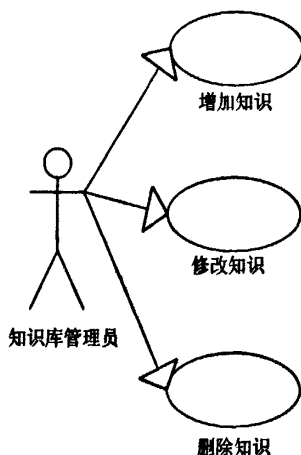


图 3-1: 智能指导知识库管理员用例图

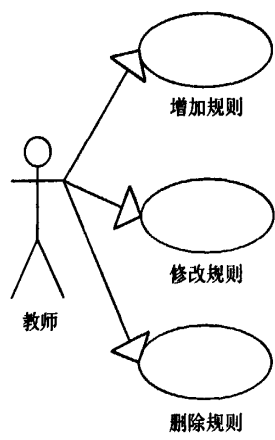


图 3-2: 智能指导教师用例图

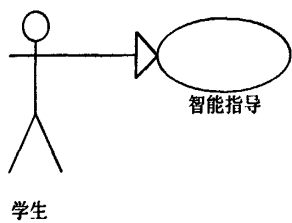


图 3-3: 智能指导学生用例图

4.2.1 知识库维护模块

以下是知识库维护模块的具体用例：

用例目标：知识库管理员对知识库进行正常维护，包括增、删、改等操作。

前件：

成功后件：系统显示知识点列表

失效后件：进入“异常处理”用例

首要角色：知识库管理员

触发：用户登录进入知识库维护模块，触发对知识点的增、删、改操作。

主要步骤：

1. 系统显示虚拟实验登录页面
2. 用户输入虚拟的用户名和密码，点击“知识库维护”
3. 系统显示所有的知识点列表。
4. 用户选择相应的操作：
 - 增：完成必要的填写，完成知识点的增加。
 - 删：选择相应的知识点的删除操作，确认删除
 - 改：找到相应的知识点进行编辑操作，完成内容编辑，确认提交。
5. 系统成功返回知识点列表页面。

扩展：无

4.2.2 学生实验智能指导推理模块

教师制定正确答案的具体用例描述：

用例目标：教师制定实验的正确答案并发布实验任务。

前件：

成功后件：系统显示实验列表

失效后件：进入“异常处理”用例

首要角色：教师

触发：用户登录进入学生实验模块，触发评分标准

主要步骤：

1. 系统显示虚拟实验登录页面
2. 用户输入虚拟的用户名和密码，成功登陆
3. 系统显示所有的实验列表。
4. 用户选择新建实验。
5. 添加实验完成，返回实验列表。
6. 用户触发实验的评分标准，系统转至实验评分标准列表。
7. 用户选择相应的操作：
 - 增：完成必要的添加操作，完成评分标准的添加
 - 删：选择相应的评分标准的删除操作，确认删除
 - 改：找到相应的评分标准进行编辑操作，完成内容编辑，确认提交。
8. 用户完成对评分标准的操作，进入实验任务列表
9. 发布实验任务

扩展：无

学生实验触发智能指导的具体用例描述：

用例目标：学生实验触发智能指导功能

前件：

成功后件：系统指导信息

失效后件：进入“异常处理”用例

首要角色：学生

触发： 用户登录进入学生实验模块，触发智能指导

主要步骤：

1. 系统显示虚拟实验登录页面
2. 用户输入虚拟的用户名和密码，成功登陆
3. 系统显示所有的实验任务列表
4. 用户开始实验
5. 用户实验过程中遇到困难，选择智能指导操作
6. 页面返回智能指导信息

扩展：无

4.3 转换器分析

由于智能指导系统是在原有虚拟实验系统的基础上进行的二次开发，原有的系统器材表示方法不能被推理引擎接受，实验的正确答案也需要转换成能被推理引擎接受的标准。

智能指导系统的开发需要在原有系统的基础上开发一个转换器，转换器主要有以下两方面的任务：

1. 对实验器材实体类的转换。
2. 对实验规则的转换。

4.3.1 类实体转换器

类实体转换器采用了面向对象的思想，将实验用到的器材抽象成可以被推理机接受的类文件，主要有两个功能：

- 抽象实验器材，将实验用到的器材抽象为推理机可以认可的知识类。
- 剔除实验器材不需要的属性和方法，只保留推理引擎需要的部分，方便推理引擎的推理工作。

我们以 windows 计算机和交换机为例，类实体转换器的转换图如下所示：

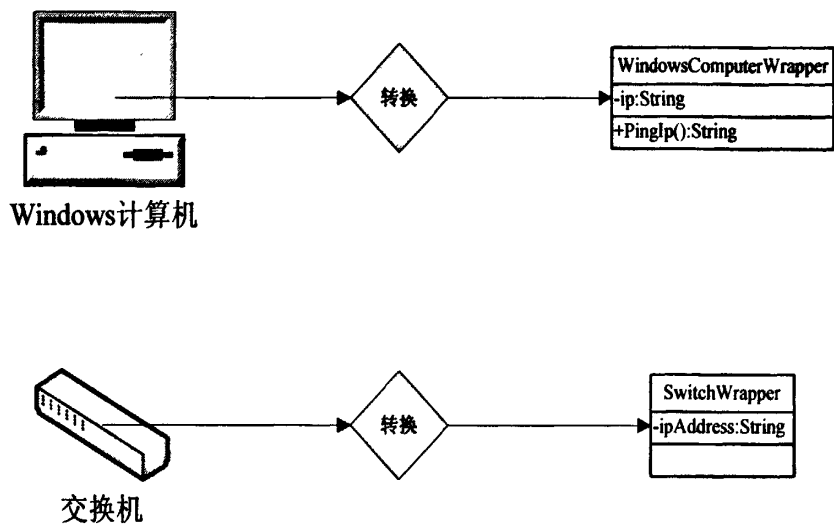


图 3-4：类实体转换图

4.3.2 规则转换器

规则转换器的主要任务是将教师制定的正确答案转换成规则引擎的规则文件。由于教师不属于系统维护人员，在发布实验的时候，教师只能通过实验平台做一遍实验做标准答案，而规则转换器的主要任务就是对教师做的正确答案进行分析，根据教师提供的指导规则提取出可供推理机使用的规则文件。

第五章 网络虚拟实验智能指导设计

5.1 智能指导流程设计

智能指导系统中涉及到三种角色：知识库管理员、教师和学生。首先是底层的知识库管理员对知识库进行维护，添加编辑实验需要用的器材，将器材抽象为类添加到知识库中，并添加相应的属性和方法。教师登陆平台发布实验，同时调用知识库中的知识点，针对实验的要求，制定实验的正确答案及指导规则，形成实验指导规则库。学生登录平台后可以看到教师发布的实验任务，根据实验任务要求开始实验，在学生实验过程中，如需要智能指导系统提供的帮助，可以触发智能指导功能，系统会读取学生当前实验的脚本内容，作为事实库放入推理机，推理得出实验指导信息反馈回学生实验页面。

下图是对智能指导流程的描述：

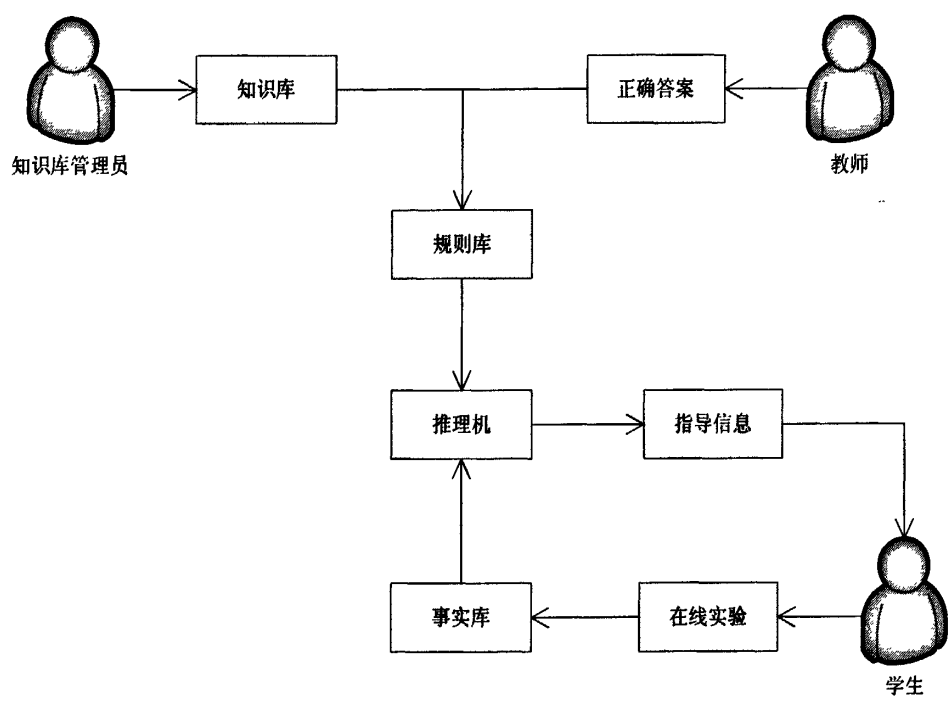


图 5-1：智能指导流程图

5.2 系统模块设计

智能指导系统主要由两个核心模块来实现具体功能，分别是知识库维护模块和智能指导推理模块。

知识库负责知识库的维护，模块的主要功能是与学生实验的知识领域相联系，利用面向对象思想的抽象化特征，将学生的知识点抽象为系统能够识别的类，包括有关教学内容的知识和技能。知识库模块是对学生实验知识的底层抽象，可以供教师制定答案使用，该模块主要包括知识库管理员维护的知识库。

智能指导推理模块主要负责具体规则的制定以及逻辑功能的实现。智能指导推理模块主要是利用知识库中提供的知识创建规则，分析学生的实验脚本，采用前向推理方式利用规则引擎对学生的实验情况进行推理，对学生的实验状况进行分析，得到学生遇到的难点，同时针对难点给出相应的指导信息，达到智能指导的效果。

由于智能指导是根据学生的实验情况进行推理，所以智能指导系统只能依托于学生实验模块进行的二次开发。学生实验模块的主要功能是提供学生在线实验的平台，同时监控学生的实验活动，在学生需要指导时提供学生实验的信息。

详细的系统框架图如下：

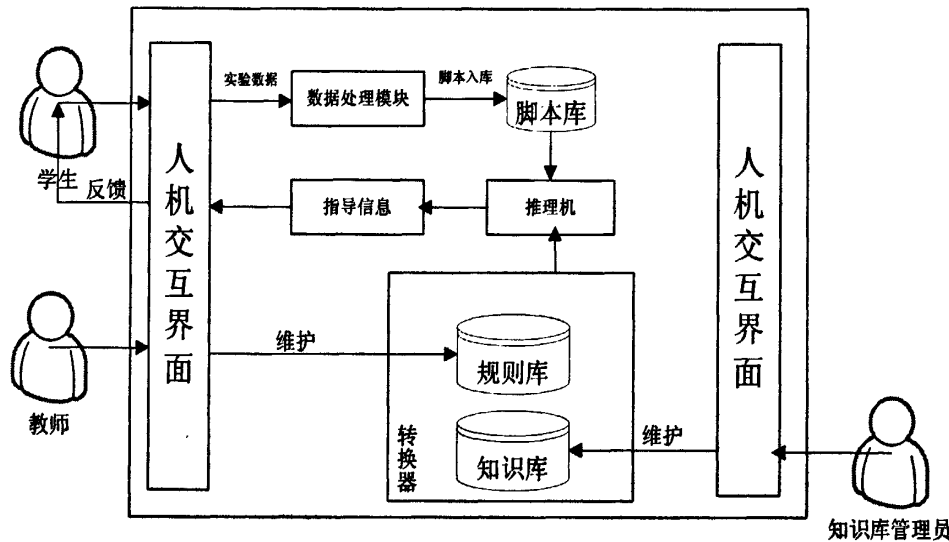


图 5-2：系统详细框架图

通过系统详细框架图可以看到，本系统一共涉及三种角色，知识库管理员负责底层知识库的维护，教师负责规则库的维护，同时有发布实验、安排实验任务、制定正确答案的权限。学生在登录系统后可以看到自己的实验任务，同时有触发智能指导的权限。

5.3 知识库维护模块

从具体功能的实现上来讲，知识库中的元素是知识的底层抽象实现，是指导规则的元规则库，是专家系统的组成部分之一。

在知识库中，具体的知识点对象对抽象为类，在类中实现了对具体对象属性和方法的约束。例如现实中的 **Windows PC** 在知识库中被抽象为 **Windows PC** 类，而针对现实中 **PC** 的属性和方法则分别被抽象为这个类的属性和方法。

5.3.1 知识库维护模块类设计

表示层：

（1）实体操作类图：

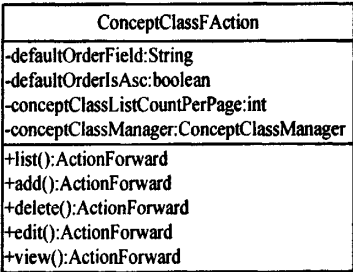


图 5-3：ConceptClassFAction 类图

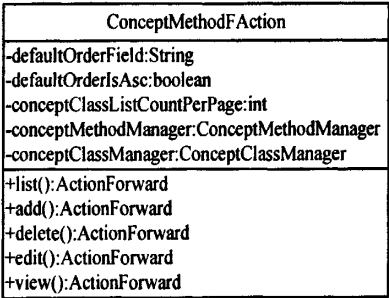


图 5-4：ConceptMethodFAction 类图

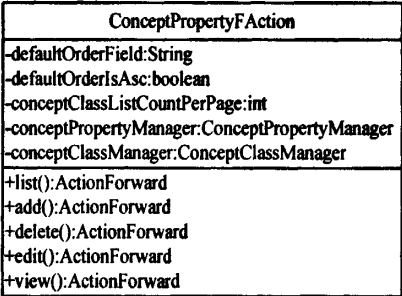


图 5-5: ConceptPropertyFAction 类图

实体操作类主要负责对知识库相关实体进行操作，包括 ConceptClass、ConceptMethod、ConceptProperty 三个实体，其中每个实体操作类都包含对相关实体的读取、添加、删除、编辑、查看操作方法。

(2) ToolUtilAction 类:

ToolUtilAction 类主要负责对知识库类、属性、方法、参数的正确性进行检验，确保各个知识点可以正确的应用于创建的规则。

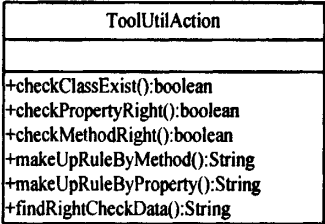


图 5-6: ToolUtilAction 类图

方法描述:

- checkClassExist(): 检查提交的知识类是否已经存在，如已经存在，则返回异常。
- checkProperttRight(): 将概念属性放到规则引擎中去，生成测试规则，验证概念属性是否满足规则引擎要求的格式。
- checkMethodRight(): 将概念方法放到规则引擎中去，生成测试规则，验证概念方法是否满足规则引擎要求的格式。
- makeUpRuleByMethod(): 根据类名和方法表达式构造规则，主要用来测试用户填写的概念方法是否满足规则引擎的要求。
- makeUpRuleByProperty(): 根据类名和属性表达式构造规则，主要用来测试用户填写的概念属性是否满足规则引擎的要求。

- findRightCheckData(): 根据用户填写的参数类型，生成测试数据用来构造测试规则。

业务层:

在业务逻辑层，系统扩展了 Manager 接口，针对知识类、类属性、类方法分别写了相应的 Manager 接口和 ManagerImpl 实现方法，下面我们以 ConceptClass 为例进行简单的描述。

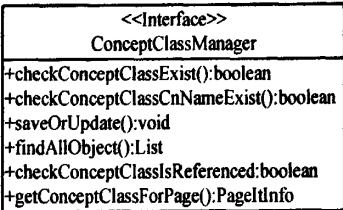


图 5-7： ConceptClassManager 接口图

方法描述:

- checkConceptClassExit(): 每一个类在知识库中只能有一个对应，这个方法用来判断表达式所代表的类是否已经存在。
- checkConceptClassCnNameExist(): 判断知识类的中文名是否已经存在。
- saveOrUpdate(): 保存或修改知识类。
- findAllObject(): 找到所有的知识类。
- checkConceptClassIsReferenced(): 检测知识类是否已经被规则引用，如果已经被引用，则无法删除该类。
- getConceptClassForPage(): 分页查看所有的知识类。

5.3.2 知识库维护模块数据库设计

表 5-1: 知识类表 (CONCEPTCLASS)

字段名	类型	非空?	缺省值	说明	备注
CONCEPTCLASS_ID	VARCHAR(32)	Y		知识类 ID	主键
CONCEPTCLASSNAME	VARCHAR(32)	Y		知识类名称	
CONCEPTCLASSCNNAME	VARCHAR(32)	Y		知识类中文名	
PLUG_ID	VARCHAR(32)	N		使用插件 id	

表 5-2: 知识方法表 (CONCEPTMETHOD)

字段名	类型	可否为空	初始 值	说明	备注
CONCEPTMETHOD_ID	VARCHAR(32)	Y		知识方法 ID	主 键
CONCEPTMETHODNAME	VARCHAR(32)	Y		知识方法名称	
CONCEPTMETHODCNNAME	VARCHAR(32)	Y		知识方法中文 名	

表 5-3: 知识属性表 (CONCEPTPROPERTY)

字段名	类型	非 空?	缺 省 值	说明	备注
CONCEPTPROPERTY_ID	VARCHAR(32)	Y		知识属性 id	主 键
CONCEPTPROPERTYNAME	VARCHAR(32)	Y		知识属性名称	
CONCEPTPROPERTYCNNAME	VARCHAR(32)	Y		知识属性中文 名	
CONCEPTCLASS_ID	VARCHAR(32)	Y		知识类 id	
PLUG_ID	VARCHAR(32)	N		使用插件 id	

表 5-4: 参数表 (PARAMTERENTITY)

字段名	类型	非空?	缺省值	说明	备注
PARAMTERENTITY_ID	VARCHAR(32)	Y		参数 id	主键
PARAMTERENTITYNAME	VARCHAR(32)	Y		参数名称	
PARAMTERENTITYTYPE	VARCHAR(32)	Y		参数类型	

表 5-5: 参数值表 (PARAMTERENTITY)

字段名	类型	非空?	缺省值	说明	备注
PARAMTERVALUEENTITY_ID	VARCHAR(32)	Y		参数值 id	主键
PARAMTERENTITY_ID	VARCHAR(32)	Y		参数 id	外键
PARAMTERVALUE	VARCHAR(32)	Y		参数值	
RULEENTITY_ID	VARCHAR(32)	Y		规则实体 ID	

5.3 智能指导推理模块

智能指导推理模块是智能指导的具体实现模块，工作流程如下：

- 教师制定实验的正确答案。
- 学生在线实验，触发智能指导功能，提交实验数据。
- 在原有系统基础上，将教师正确答案、学生实验数据转换成推理引擎可

接受的标准。

- 生成规则文件。
- 推理机工作，返回推理结果

5.3.1 智能指导推理模块类设计

(1) RuleFAction

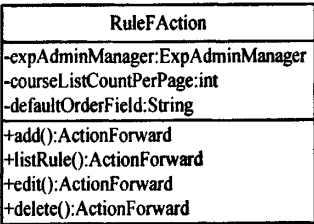


图 5-8： RuleFAction 类图

RuleFAction 主要负责实验规则文件的操作，包括规则的增删查改，每个实验可以有多个实验规则，每个规则都是由知识类、知识属性或方法、智能指导意见构成的。

(2) ToolUtilAction

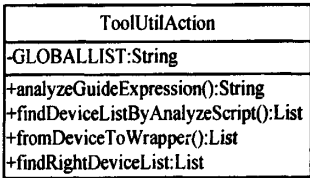


图 5-9： ToolUtilAction 类图

ToolUtilAction 类主要负责将实验器材、实验规则转换成可以被推理机接受的标准。

主要类属性、方法描述：

- **GLOBALLIST:** GLOBALLIST 是指导规则的全局变量，用于存储推理机推理的结果，推理结果以 String 的形式表示，主要包括每个实验规则的 id，然后根据实验规则 id 读取智能指导信息
- **findDeviceListByAnalyzeScript():** 分析实验脚本信息，获取实验中使用的实验器材。无论是教师制定正确答案，还是学生提交的实验脚本，都需要先读取实验中用到的仪器，这个方法的主要作用就是提取实验的器材，提交给系统由实验器材转换成推理机可以接受的知识类。

- **fromDeviceToWrapper()**: 将提取到的实验器材转换成可以被推理机接受的知识类。
- **analyzeGuideExpression()**: 转换实验规则，将教师制定的正确答案转换成可以被推理机认可的规则文件。

(3) ExpTaskFAction

ExpTaskFAction
-expAdminManager:ExpAdminManager
+expGuide():ActionForward

图 5-10: ExpTaskFAction 类图

ExpTaskFAction 主要负责推理机的实现，获取推理结果返回给用户。

方法描述：

- **expGuide**: 负责推理机的实现，获取推理结果，根据推理结果读取实验规则的指导信息，将指导信息返回给学生。

5.3.2 智能指导推理模块数据库设计

表 5-6: 规则实体属性表

字段名	类型	非空?	缺省值	说明	备注
RULEENTITY_ID	VARCHAR(32)	Y		规则实体 id	主键
CONCEPTCLASS_ID	VARCHAR(32)	Y		知识类 id	
DEVICENAME	VARCHAR(32)	Y		设备名	
EXPERIMENT_ID	VARCHAR(32)	Y		实验 id	
OPERATIONGUIDE	VARCHAR(32)	Y		操作指导	

表 5-7: 规则实体方法表

字段名	类型	非空?	缺省值	说明	备注
RULE_ID	VARCHAR(32)	Y		规则实体 ID	主键
CONCEPTMETHOD_ID	VARCHAR(32)	Y		知识方法 ID	主键
CONCEPTPROPERTY_ID	VARCHAR(32)	Y		知识属性 ID	主键

5.3.3 规则实体的设计

智能指导系统是基于学生实验信息进行匹配，要求系统对学生的实验信息进行快速反应，综合考虑现在成熟的推理引擎技术，我们选择 Jboss Drools 作为系统的推理引擎。

传统的 Drools 规则引擎工作流程，需要事先在持久层写好相关的 Drools 规则文件，然而，对于虚拟实验系统来说，系统的开发人员并不负责具体实验规则

的制定，而教师也不能像系统维护人员那样轻松的书写 Drools 规则文件，所以具体的规则实体是由系统将教师制定的实验标准编译而来的。

一条完整的规则文件应该由包名、全局变量、规则名、规则体构成。

包的设计类似于 Java 中的包，主要作用是将相似或者同一个实验的规则文件封装在一起，这样方便区别于其他的规则文件或者不同实验的相同规则。

全局变量负责系统与规则引擎之前的通信工作，推理机工作的结果将会存在全局变量中供 Java 读取。

规则名和包共同构成了规则的唯一识别，对于同一个实验有不同的规则，系统采用顺序命名的方法，每读取一个规则变量 i 自动加 1，与字符串“rule”共同构成规则名。

规则体是规则引擎运行的根本，规则体是由正确答案经转换器转换而成。

以下是以计算机网络实验为例设计的规则文件：

```
package com.bupticet.rule
import com.bupticet.exp.know_admin.web.action.ToolUtilAction;
global java.util.List scoreLists;
rule "rule0"
when
not($entity:bupticet.virexp.linux.wrapper.WindowsComputerWrapper(m_name=="Windows_pc 0",NET_address=="192.168.199.1"))
then
scoreLists.add("402847f124eb428d0124ebd8efe80015");
end
rule "rule1"
when
not($entity:bupticet.virexp.linux.wrapper.WindowsComputerWrapper(m_name=="Windows_pc 1",NET_address=="192.168.199.3"))
then
scoreLists.add("402847f124eb428d0124ebdb896001f");
end
rule "rule2"
when
$entity:bupticet.virexp.linux.wrapper.WindowsComputerWrapper(m_name=="Windows_pc 0")
not(eval($entity.pingIP("192.168.199.3")==true))
```

```
then
scoreLists.add("402847f124eb428d0124ebed5865001c");
end
```

规则说明：

Rule0

如果不存在 ip 地址为“192.168.199.1”的 Windows_pc0,那么全局变量将添加该实验正确答案的 id，然后根据 id 读取指导信息。

Rule1

如果不存在 ip 地址为“192.168.199.3”的 Windows_pc1，那么全局变量将添加该实验正确答案的 id，然后根据 id 读取指导信息。

Rule2

如果 Windows_pc1 无法 ping 通“192.168.199.3”的 ip 地址，那么全局变量将添加该实验正确答案的 id，然后根据 id 读取指导信息。

5.3.4 运算符扩展功能设计

运算符扩展是对智能指导系统功能的一个补充。在计算机网络实验中，由于 ip 地址等参数都是以字符串存储在数据库中的，然后由于在实际实验操作中，实验要求对 ip 区间进行判定，智能指导系统需要完成对运算符的扩展。

运算符扩展主要存在两个方向，一个是对数值区间的扩展，一个是对字符串的扩展判定。

数值区间的判定：Drools 规则引擎支持多种运算符，包括介于、大于、小于等，针对数值区间的判定，只需要直接编写规则文件即可。

字符串的扩展判定：对字符串的判定有多种，包括包含等一系列的转换操作，所以对字符串的扩展我们需要在规则文件中载入对字符串判定的 Java 方法。

数据库设计如下：

表 5-8: 运算符扩展数据库表

字段名	类型	非空?	缺省值	说明	备注
PARAMTERVALUEENTITY_ID	VARCHAR(32)	Y		参数值 id	主键
PARAMTERENTITY_ID	VARCHAR(32)	Y		参数 id	主键
PARAMTERVALUE	VARCHAR(32)	Y		参数值	
RULEENTITY_ID	VARCHAR(32)			规则实体 id	
OPERATOR	VARCHAR(32)			运算符	

一条简单的运算符扩展后的规则如下：

```
package com.bupticet.rule
import com.bupticet.exp.know_admin.web.action.ToolUtilAction;
global java.util.List scoreLists;
rule "rule0"
when
$entity:bupticet.virexp.linux.wrapper.WindowsComputerWrapper(m_name=="Windows_pc 0")
not(eval(ToolUtilAction.ipIsValid("192.168.199.1-192.168.199.10",$entity.getNET_address())))
then
scoreLists.add("402847f124eb428d0124ebd8efe80015");
end
```

规则说明:

如果不存在 Windows_pc0 的 ip 地址介于"192.168.199.1-192.168.199.10"之间,那么将在全局变量添加该规则的 id, 然后根据 id 读取相应知识点的指导信息。

从上述规则我们可以看到, 与我们之前的规则设计相比, 除了包名、全局变量、规则名、规则体外, 运算符扩展后的规则增加了加载文件的信息。

5.4 学生实验模块

智能指导系统是在原有虚拟实验系统的基础上进行的二次开发, 由于智能指导功能是依附于系统的实验模块工作的, 在这部分我们只简单的介绍下学生实验模块在智能指导中的作用。

系统采用 Java Applet 技术向学生提供虚拟的实验环境, 学生登陆系统后, 根据教师发布的实验任务进行在线实验, 在实验的过程中, 如果遇到困难可以触发智能指导, 向智能指导系统提交学生实验脚本存入事实库进行推理。

学生实验模块在智能指导中主要负责向推理机提交学生实验脚本存入事实库, 是推理机工作的重要条件。

第六章 网络虚拟实验智能指导系统实现及扩展

虚拟实验智能指导系统的实现主要分成三部分实现，分别是知识库维护模块、智能指导推理模块、实验模块。从系统开发来讲，知识库维护模块和智能指导模推理块是开发的重点和难点，专家知识模块负责智能指导系统的底层实现，智能指导模块是系统的直接实现。

智能指导系统是基于已有的智能批改专家库实现的，所以本章只简单介绍下知识库维护模块的页面实现，重点介绍智能指导推理模块的实现。

6.1 知识库维护模块的实现

6.1.1 知识库维护界面实现

知识库的实现如下图所示：

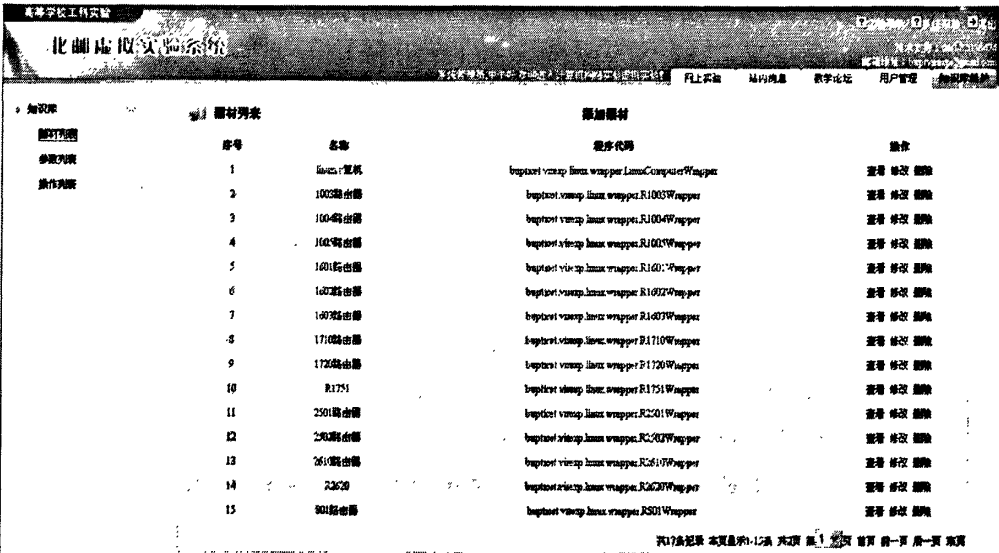


图 6-1：知识类维护界面

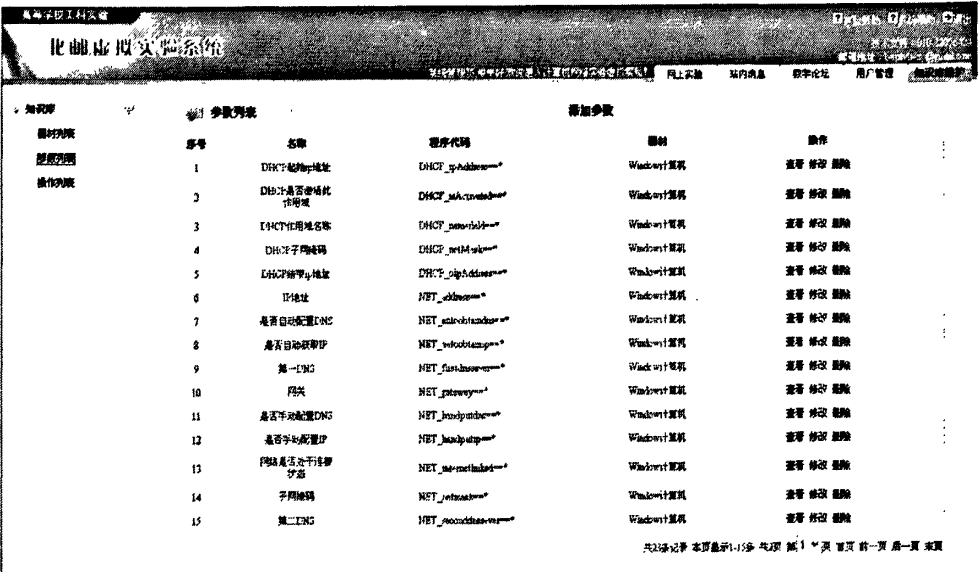


图 6-2：知识类属性维护界面

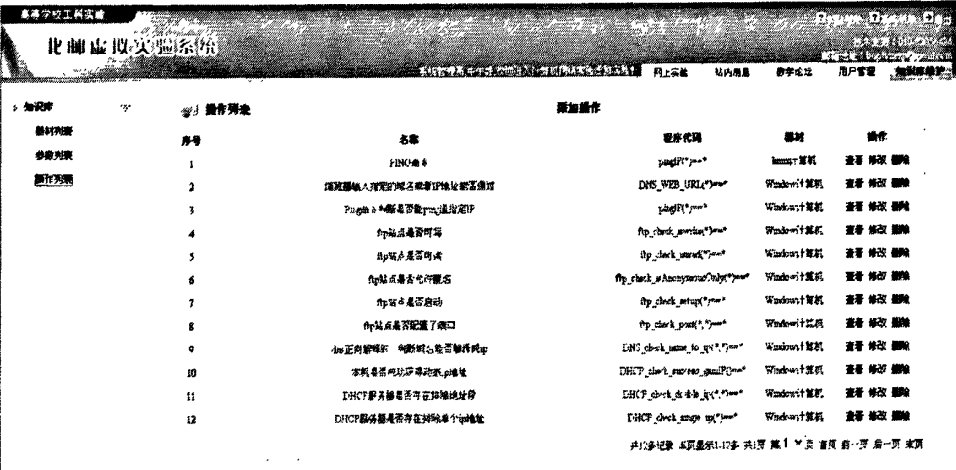


图 6-3：知识类方法维护界面

知识库维护主要是针对知识类、知识类属性、知识类方法的增删查改功能，以下是具体的实现界面：

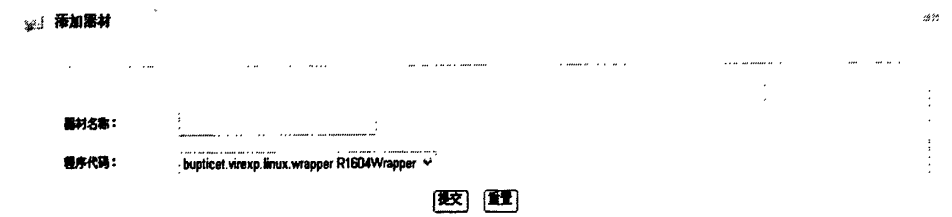


图 6-4：知识类操作界面

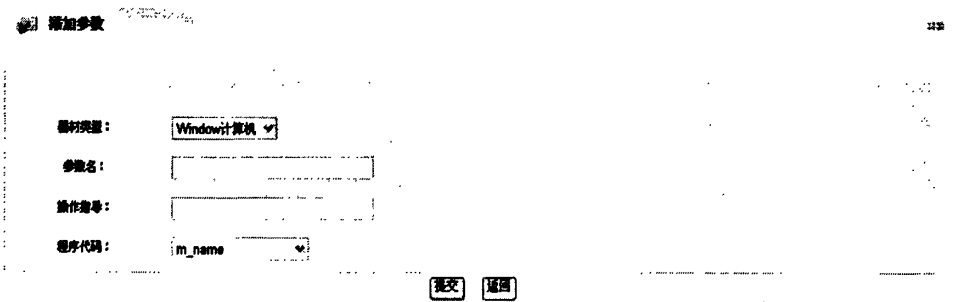


图 6-5：知识类属性操作界面

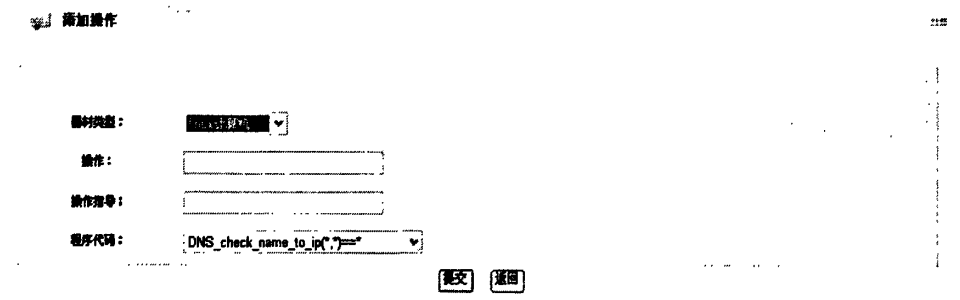


图 6-6：知识类方法操作界面

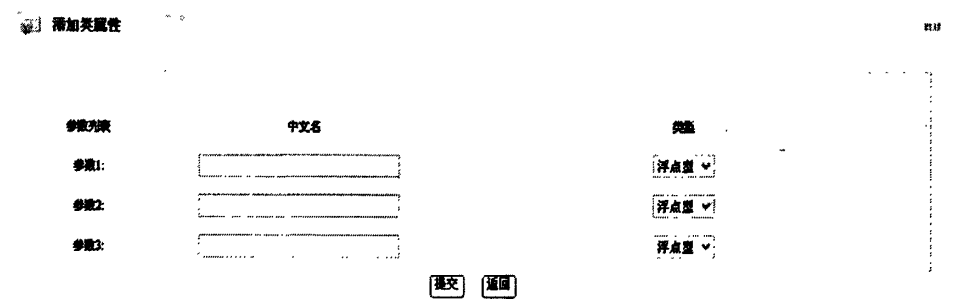


图 6-7：参数操作界面

6.1.2 知识库维护实现难点要点

知识库维护模块实现的主要难点是通过 Ajax 异步实现类属性和类方法的级联菜单。在选择类参数或类方法的时候，每次对类下拉框进行操作，会通过 Ajax 的异步调用，将选择的类传送到后台，通过 Java 的反射机制将这个类的属性和方法传输到页面，实现无刷新更新页面内容。

6.2 智能指导推理模块实现

智能指导推理模块的实现主要包括指导规则的制定、转换器的实现、推理机的实现三部分。

6.2.1 推理模块界面实现

教师制定实验规则的界面如下图所示：

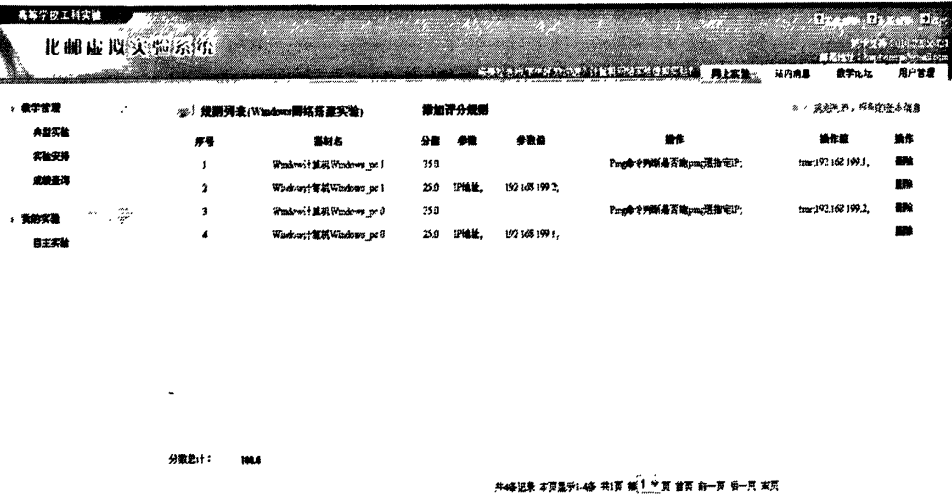


图 6-8：实验规则界面



图 6-9：添加规则-选择设备界面

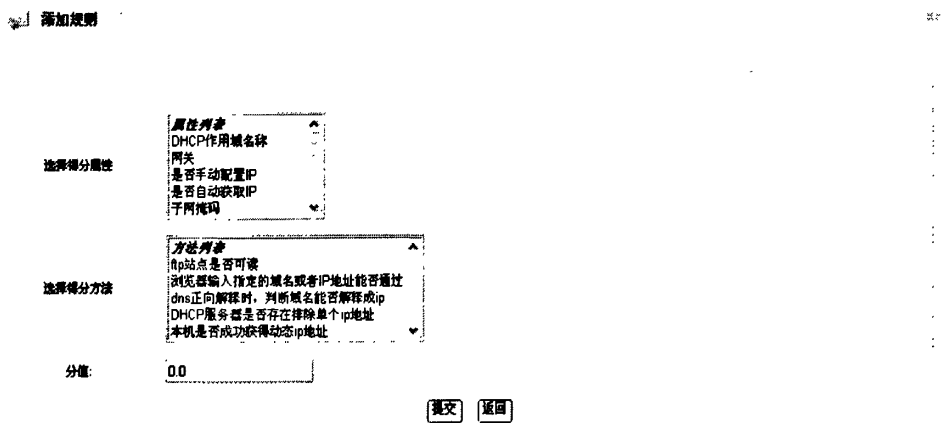


图 6-10：添加规则-选择属性、方法界面

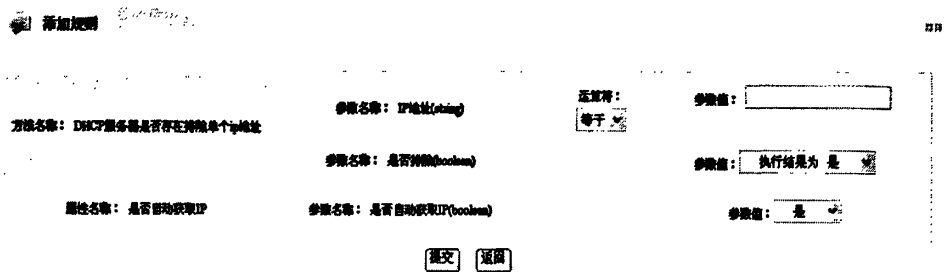


图 6-11: 添加规则-填写参数界面

智能指导推理机的实现如图所示：

智能指导窗口		
序号	错误地方	操作指导
1	Windows_pc 0IP地址配置错误	ip地址应配置为192.168.199.1-192.168.199.10
2	Windows_pc 0Ping命令判断是否能ping通指定IP失败	

图 6-12: 智能指导推理机实现页面

6.2.2 推理模块实现的重点难点

推理模块实现的难点在于教师制定实验规则的实现、转换器的实现、推理机的实现。

● 教师制定规则的实现

教师制定实验规则是以对每个实验发布正确的实验操作为基础的，要将教师的实验标准转换成可以被推理机认可的规则文件。在转换规则之前，首先要识别脚本中的仪器以及仪器的属性和方法。具体的实现如下：

```
/**
 * @param script
 * @return
 * 根据标准答案的脚本，找到实验台上的主要设备，在此，我们只找metaDevice，原因是，只有这些设备上存在得分点；
 */
public static List findDeviceListByAnalyzeScript(String script)
{
    if(Experiment.readScriptForAnalyze(script) != null){
        return(Experiment.readScriptForAnalyze(script).getMetaDeviceBuf());
    }
}
```

```

    }
    else{
        return null;
    }
}

```

然后根据识别的仪器读取相应的类文件，列出该类的属性和方法，教师选择相应的属性和方法作为实验的指导规则。

● 转换器的实现

转换器的实现主要分为两种，一种是对器材实体的转换，一种是对教师制定规则的实现。

类实体转换器：

类实体转换器主要负责将实验中使用到的仪器转换成可以被推理引擎识别的类文件，类实体转换器主要采用工厂模式对实验模块仪器的 `Javabean` 进行分析转换。

具体的实现如下：

```

/**
 * @param deviceList
 * @return 对设备进行转化，实现包转；
 */
public static List fromDeviceToWrapper(List deviceList)
{
    if(deviceList == null){
        return null;
    }
    List result=new ArrayList();
    if(Experiment.instance().getPhysicsLayerFacade() != null){

        Experiment.instance().getPhysicsLayerFacade().doArithmetic();
    }
    for(int i=0;i<deviceList.size();i++)
    {
        MetaDevice dev=(MetaDevice)deviceList.get(i);
        if(dev.m_component_type==Device.LINUX_COMPUTER)
        {
            LinuxComputerWrapper wrapper=new

```

```

        LinuxComputerWrapper((LinuxComputer)dev);
    result.add(wrapper);
}

```

```

.....
}
}
}

```

规则转换器：规则转换器的主要任务是将教师制定的具体规则转换成可以被规则引擎识别的规则语句。具体的实现如下：

```

/**
 * @param ruleEntityList
 * @return 根据给定的规则实体列表，解释出符合规则引擎的表达式；
 */

```

```

public static String analyzeRuleExpression(Set ruleEntityList)

```

● 学生实验参数的获取

推理机的工作需要规则库和事实库的支持，规则库来自教师制定的具体规则，针对每个学生实验的具体情况，推理机进行推理需要获取学生的实验数据。考虑到指导的信息是基于学生的实验，学生在触发智能指导帮助的时候能够及时的反馈指导的信息可以提高指导的效率，在实验数据获取这部分采用 Javascript 来传输数据并及时反馈。具体的实现代码如下：

```

function pop(action){
    var s;
    try{
        s = getScriptFromPlugin();
        document.expTaskForm.script.value=s;
    }catch(e){
    }
    Var
    experimentTaskId=document.getElementById("experimentTaskId")
    .value;
    var actionUrl=action;
    while(true){
        index = s.indexOf(">");
        if(index < 0){
            break;

```

```

    }
    s = s.replace("&gt;", "@@@");
}

Var
    pars="method=expGuide&expScript="+s+"&experimentTaskId="+
        experimentTaskId+"";
    show_popup(actionUrl,pars);
}

```

● 对实验数据的推理

推理机的运行需要知识库和事实库的数据,在智能指导系统中,对这两方面数据的处理方法分别是:

获取实验指导规则数据,转换为可以被推理机接受的数据:

```
String rule=ToolUtilAction.analyzeGuideExpression(ruleEntityList);
```

获取学生实验脚本信息,对学生实验中的实体类进行转换:

```
List deviceL=ToolUtil.findDeviceListByAnalyzeScript(expScript);
```

程序说明: ToolUtilAction 是自定义的类方法,主要负责对实体类和规则实体类进行转换。

推理机的实现:

```

PackageBuilder builder=new PackageBuilder();
builder.addPackageFromDrl(changeFromStringToReader(rule));
RuleBase ruleBase=RuleBaseFactory.newRuleBase();
ruleBase.addPackage(builder.getPackage());
StatefulSession session=ruleBase.newStatefulSession();
session.fireAllRules();
session.dispose();

```

● 对实验指导数据的封装

学生的实验指导信息,采用 JSON 格式来封装。

```

JSONObject jsonobj = new JSONObject();
jsonobj.put("mistake",exp+"</table>");
String json=jsonobj.toString();
response.setHeader("X-JSON", json.toString());
response.setCharacterEncoding("UTF-8");
response.setHeader("Cache-Control", "no-cache");

```

```
log.info(json.toString());
response.getWriter().print(json.toString());
```

6.3 运算符扩展

Drools 规则引擎本身支持多种运算符的使用,然而在虚拟实验中仍然存在诸如 ip 地址等属性是以字符串的形式存储的,要判断学生设定的 ip 是否在实验要求的 ip 区间,需要在 Drools 规则体中加入 Java 逻辑判断语句。

具体的实现代码如下:

```
public static boolean ipIsValid(String ipSection, String ip) {
    if (ipSection == null)
        throw new NullPointerException("IP段不能为空!");
    if (ip == null)
        throw new NullPointerException("IP不能为空!");
    ipSection = ipSection.trim();
    ip = ip.trim();
    final String REGX_IP =
        "((25[0-5]|2[0-4]\\d|1\\d{2}|[1-9]\\d|\\d\\.){3}(25[0-5]|2[0-4]\\d|1\\d{2}|[1-9]\\d|\\d)";

    final String REGX_IPB = REGX_IP + "\\-" + REGX_IP;
    if (!ipSection.matches(REGX_IPB) || !ip.matches(REGX_IP))
        return false;
    int idx = ipSection.indexOf('-');
    String[] sips = ipSection.substring(0, idx).split("\\.");
    String[] sipe = ipSection.substring(idx + 1).split("\\.");
    String[] sipt = ip.split("\\.");
    long ips = 0L, ipe = 0L, ipt = 0L;
    for (int i = 0; i < 4; ++i) {
        ips = ips << 8 | Integer.parseInt(sips[i]);
        ipe = ipe << 8 | Integer.parseInt(sipe[i]);
        ipt = ipt << 8 | Integer.parseInt(sipt[i]);
    }
    if (ips > ipe) {
```

```
        long t = ips;
        ips = ipe;
        ipe = t;
    }
    return ips <= ipt && ipt <= ipe;
}
```

第七章 网络虚拟实验智能指导的不足与发展

经过近两年的开发,网络虚拟实验智能系统的开发已经逐渐成熟,实现了需求与设计的所有功能,并与 2008 年 3 月正式投入教学使用,经过 1 年半以来的使用,发现系统仍存在部分实际应用中的不足。

首先是知识库的知识量不足。由于系统的使用期太短,没有大量的数据支持,学生实验数量少,都是一些相对比较简单实验,知识库能够获取的知识点有限,造成指导系统知识库的相对匮乏。随着虚拟实验系统的发展以及用户量的增加,逐步扩展指导系统的数据库,提取学生实验知识点进行总结归纳,知识库的容量和质量都会有逐步的提升。

其次,智能指导的帮助方式有待提升。智能指导系统目前的开发主要是面向主动帮助,即学生在做教师发布的实验时遇到困难主动触发帮助才可以针对特定的知识点进行帮助。随着知识库的扩展,主动帮助要逐渐向被动帮助扩展,常规实验帮助要逐渐向学生自主实验扩展,学生无论在做教师布置的常规实验还是自己设计的自主实验的时候,知识库都可以根据掌握的专家知识对学生的实验进行一定的主动帮助,实时向学生提供每步实验的操作专家知识帮助。

第八章 论文工作总结

自 2008 年 3 月, 作者参加了国家十一五科技攻关项目“虚拟实验智能指导与管理系统的研发”(编号 2008BAH29B04), 该项目基于国产基础软硬件产品, 结合虚拟实验课程的需求, 面向高校开发实验管理系统。目前本系统已经开发完成, 并且做了产品化工作, 产品网站为 <http://virexp.bupticet.com>。

论文主要以这个项目的开发工作作为背景, 以提出问题、分析问题、解决问题为框架针对开发的一些工作进行阐述。调查了虚拟实验系统的发展现状, 同时针对系统开发需要用到的理论进行分析, 在理论的基础上对系统的开发工作进行了详细的分析和设计。其中分析了基于 Web 的远程教学的流程框架, 在这基础上对系统的功能模块进行了划分, 并对各个模块进行了详细的分析、设计, 最后论文描述了系统的实现状况。

计算机网络智能指导系统经过半年的运行以来, 已经应用于计算机网络、计算机通信网等 4 门课程, 使用学生超过 2000 人, 学生的实验成绩较智能指导系统发布前平均提高 10 分。

经过对远程教育技术的研究, 作者认为将虚拟实验技术引入远程教学的方案, 是对远程教学手段的一个很重要补充, 而智能指导的开发就是对虚拟实验教学模式的完善, 同时也是将专家系统引入远程教学的一个大胆尝试。

论文源于一个软件开发项目, 在论文中也针对该项目进行了一些描述, 论文难免较多的针对具体的一个项目进行描述, 然而这个项目毕竟知识专家系统引入教学系统的一个具体应用, 作者更多的是希望能通过针对这个项目总结一些远程教学技术手段的思路。由于作者水平有限, 论文很多方面难免会存在偏颇和疏忽。相信随着远程教学技术的不断发展, 虚拟实验系统会越来越多的用于远程教学系统。

参考文献

- [1] 吴维云. 我国远程教育发展中的云思考. 中小学电教 2009.9 P1
- [2] 母晓科, 蹇滔, 李新科. 浅析人工智能与专家系统. Computer Knowledge and Technology 电脑知识与技术. ISSN 1009-3044 Vol.5, No.7, March 2009. P1
- [3] 缴明洋, 谭庆平. Java 规则引擎技术研究. 计算机与信息技术. P41
- [4] 刘 伟. Java 规则引擎—Drools 的介绍及应用. 微计算机应用. 第 26 卷第 6 期. 2005 年 11 月. P2
- [5] JBossRulesUserGuide.
<http://labs.jboss.com/file-access/default/members/jbossrules/freezone/docs/3.0.5/html/index.html>
- [6] 金聪, 戴上平, 郭京蕾, 张维. 人工智能教程. 清华大学出版社. 2007. P62
- [7] 刘际 赵广利. Drools 规则引擎在实现业务逻辑中的应用. 中国科技论文在线. 论文编号:200612-273
- [8] 丁启杰. Linux 网络虚拟实验智能批改的设计与实现. [学位论文]. 北京邮电大学 2008
- [9] 吴勇军, 李 勇, 甘路明, 李玲鞠. 基于规则引擎的智能决策支持系统开发. 电脑学习. 第一期. 2009 年 2 月. P2
- [10] 李刚. 轻量级 J2EE 企业应用实践. 电子工业出版社. 2007. P114
- [11] Spring 技术文档. <http://www.ibm.com/developerworks/cn/java/wa-spring1/>
- [12] 孙卫琴. 精通 Hibernate: Java 对象持久化技术详解. 电子工业出版社. 2007. P16
- [13] 百度百科 Hibernate 词条
http://baike.baidu.com/view/7291.htm?fr=ala0_1
- [14] 祝红涛, 刘海松, 郝军启. Ajax 从入门到精通. 电子工业出版社. 2008. P98
- [15] 廖雪峰. JSON 入门指南. IBM DeveloperWorks
<http://www.ibm.com/developerworks/cn/web/wa-lo-json/?ca=drs-tp3308>

致 谢

衷心感谢我的导师文福安教授在研究生学习期间对我的谆谆教诲和悉心指导。在攻读硕士学位期间，恩师在学业和生活上给了我多方面的帮助。文老师严谨的治学态度，朴实的工作作风，深刻的理解力和耐心的指导为我在攻克课题中的困难阶段拨开迷雾，令我茅塞顿开。恩师在学术上的高深造诣，渊博的科学知识和宽以待人的高尚品德给我树立了榜样，是我人生的指路灯。再次衷心感谢。

衷心感谢上官右黎教授，上官教授在研究生学习和实验室工作中给了我很多帮助。特别是上官老师求实的治学态度和实事求是的工作作风给我今后的生活和学习指明了方向。

衷心感谢实验室陈美松老师，在实验室工作期间，陈老师给了我悉心指导，对我学习中的不足和学习的误区提出宝贵的建议，使我在学习和工作中少走弯路，并在论文开题提出宝贵的意见。

衷心感谢已经毕业的实验室各位学长，他们在我刚进实验室，什么不懂的时候给予讲解，使我很快熟悉实验室的项目。

感谢研究所“虚拟实验”项目组所有成员，这几年大家朝夕相处，工作中群策群力、互相帮助、精诚合作。感谢项目组成员在生活、学习、工作中对我的支持。

感谢网络教育技术研究所全体老师和同学的热情帮助和支持！

作者攻读学位期间发表的学术论文目录

- [1] Drools 规则引擎的开发应用, 中国科技论文在线, 2009 年, 第一作者 属于规定的核心学术期刊, 论文署名单位为北京邮电大学网络教育学院。

