申请上海交通大学工程硕士学位论文

基于规则引擎的信用卡申请反欺诈系统设计与实现

学校代码：　10248
作者姓名：　张峰
学　　号：　1100379104
第一导师：　步丰林
第二导师：　张艳红
学科专业：　软件工程
答辩日期：　2013 年 10 月 31 日

上海交通大学软件学院

2013 年 10 月

A Dissertation Submitted to Shanghai Jiao Tong University

for Master Degree of Engineering

DESIGN AND IMPLEMENTATION OF AN ANTI-FRAUD FOR CREDIT CARD APPLICATION BASE ON RULE ENGINE

University Code：	10248
Author：	ZhangFeng
Student ID:	1100379104
Mentor 1：	BuFengLin
Mentor 2:	ZhangYanHong
Field：	Software Engineering
Date of Oral Defense：	Oct 31, 2013

School of Software

Shanghai Jiaotong University

Oct, 2013

上海交通大学

学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。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作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：张烽

日期：2013 年 10 月 20 日

上海交通大学

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。本人授权上海交通大学可以将本学位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、缩印或扫描等复制手段保存和汇编本学位论文。

保密□，在___年解密后适用本授权书。

本学位论文属于

不保密☑。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：

日期：2013年10月20日

日期：2013年10月20日

基于规则引擎的信用卡申请反欺诈系统设计与实现

摘 要

目前，随着银行信用卡业务的日益发展，信用卡申请欺诈是银行信用卡面临的一个严重的问题。申请欺诈是欺诈分子通过盗取或仿冒他人的身份信息申请信用卡。在信用卡申请过程中申请人是通过电话、信件或因特网等渠道进行的，由于在信用卡申请审核过程中审核人员由于无法面对面地核对申请人身份和证件，所以只要欺诈分子盗取了他人身份信息，欺诈分子就可能成功地得到信用卡，所以近年来银行信用卡申请欺诈经常会发生。

本文首先分析了信用卡申请欺诈的国外和国内研究现状，对比分析国外和国内对于信用卡申请欺诈防范措施的差别，在结合银行的信用卡申请欺诈侦测需求，归纳出信用卡申请欺诈的侦测方案：使用信用卡申请欺诈风险评分模型来预测信用卡申请欺诈的概率，对信用卡申请欺诈侦测提供量化参考指标；使用信用卡申请欺诈规则为信用卡申请欺诈侦测提供政策性判断。以此方案进行分析设计：使用规则推理引擎技术来支持信用卡申请欺诈风险评分模型计算和信用卡申请欺诈规则的处理；使用基于规则的中文地址分词模糊匹配技术来进行申请信息的相关比对，获得信息比对的相似率，从而提高欺诈侦测的效率；建立规则管理平台来进行评分模型和规则开发管理；建立信用卡申请欺诈侦测相关的数据集市，对银行内部和外部数据源进行抽取、加载、转换、检查、存储等处理，来保证数据的完整和一致性，为信用卡申请欺诈侦测提供良好的数据基础。

本文依据银行信用卡申请欺诈侦测需求分析，对信用卡申请反欺诈系统的系统架构和系统功能进行设计、实现和验证。整个应用系统将由规则引擎平台子系统和欺诈侦测管理子系统组成。(1) 规则引擎平台子系统主要包括服务接口、规则引擎和规则管理等功能。服务接口功能是提供给欺诈侦测管理子系统调用规则引擎服务的实时联机接口；规则引擎功能是提供规则运行处理；规则管理功能是规则的编辑、验证、审批、部署、监控和参数维护等。(2) 欺诈侦测管理子系统主要包括数据接口服务、欺诈侦测服务、欺诈管理和统计报表功能，其中数据接口服务功能是批量的采集信用卡申请欺诈侦测相关数据；欺诈侦测服务功能是批量的使用信用卡申请欺诈侦测相关数据调用规则引擎平台子系统并返回侦测结果；欺诈查询服务功能是申请欺诈侦测结果的查询；统计报表服务功能是统计欺诈侦测的效果。

此系统设计在国内银行的项目中获得了应用，客户反应系统运行状况良好，欺诈侦测效率得到提高。为将来的业务发展预留的提升空间。

关键词 规则引擎，评分模型，申请欺诈，模糊匹配，信用卡申请

DESIGN AND IMPLEMENTATION OF AN ANTI-FRAUD FOR CREDIT CARD APPLICATION BASE ON RULE ENGINE

ABSTRACT

With the booming of credit card business, credit card fraud became one of the serious problems that banks are facing nowadays. Credit card fraud is performed by cheaters applying for credit cards using other people's identity information. During the process of credit card application, the appliers submit their applications through telephone, mail or internet, which causes the verification of the appliers' identity impossible. Once the cheaters stolen other people's identity, they can successfully get credit cards using the stolen information, thus why credit card fraud happens more often these years.

This article analysis the current research situation of credit card fraud both at home and abroad first and compares their different anti-fraud measures adopted. While taking the domestic banks' anti-fraud requirements into account, it gives a suggested anti-fraud detection scheme, which is using credit card fraud risk scoring model to predict the possibility of such fraud, making reference indexes to fraud detection measurable and provide policy decisions for fraud detection via decision rules. Analysis and design based on this scheme lead to the use of rule engine technology for scoring calculation based on credit card fraud risk scoring model and process of fraud detection via decision rules; the use of rule based Chinese words segamentation fuzzy matching technology to compare the similarity of applications' addresses in order to improve the efficiency of fraud detection; establishing a rule management platform to manage the development of scoring model and rules; establishment of credit card fraud detection related data mart, on the bank's internal and external data sources extract, load, transform, inspection, storage and other handling,to ensure data integrity and consistency, provide a good data base for credit card fraud detection.

.Based on the credit card fraud requirement analysis, design and implementation and test of systems architecture and system functions is required. The system will be composed of two sub systems, i.e. a rule engine platform and anti-fraud application management system. Among which: the rule engine platform's functions mainly include service interface, rule engine and rule management, service interface function is real-time online interface for anti-fraud application management system to invoke rule engine service; rule engine function is to provide a rule processing; rule management function is the rule editor, verification, approval, deploying and monitoring etc. the anti-fraud application management system's functions mainly include data interface service, fraud detection service, fraud management and statistics report service, data interface service function is batch acquisition application for credit cardfraud detection data; fraud detection service are batch use credit card fraud detection data to invoke rule engine platform and returns the results of detecting; fraud management service function is to query for fraud detection results; statistics report service function is statistical fraud detection effect.

The system's design has been utilized in application projects at domestic banks and acquired well feedbacks from customers due to its imporved fraud detection efficiency and good system running condition, which leaves room for future business development.

Keywords rule engine, scoring model, application fraud, fuzzy matching, credit card application

目 录

1. 绪 论 1
1.1. 论文主要工作 1
1.2. 论文结构安排 2
2. 信用卡申请反欺诈系统概述 3
2.1. 研究现状 3
2.1.1. 国外研究现状 3
2.1.2. 国内研究现状 3
2.1.3. 国内外研究对比 4
2.2. 信用卡欺诈侦测手段 5
2.2.1. 业务手段 5
2.2.2. 技术手段 6
2.3. 信用评分模型开发流程 6
2.4. 相关技术知识 9
2.4.1. 规则引擎 9
2.4.2. 基于规则的中文地址分词模糊匹配 11
3. 信用卡申请反欺诈系统需求分析 13
3.1. 信用卡申请审核流程 13
3.2. 欺诈侦测需求 14
3.2.1. 黑名单检查 15
3.2.2. 评分模型计算 16
3.2.3. 规则处理 18
3.3. 系统管理功能需求 22
3.3.1. 规则管理 22
3.3.2. 欺诈查询 22
3.3.3. 统计报表 23
4. 信用卡申请反欺诈系统设计 26
4.1. 系统设计原则 26
4.2. 系统架构设计 27
4.3. 应用结构设计 28
4.4. 规则引擎平台子系统后台功能设计 30
4.4.1. 服务接口 31

4.4.2. 规则引擎 37
4.4.3. 规则管理服务 38
4.5. 欺诈侦测管理子系统后台功能设计 41
4.5.1. 数据接口服务 42
4.5.2. 欺诈侦测服务 48
4.5.3. 欺诈查询服务 49
4.5.4. 统计报表服务 50
4.6. 系统非功能设计 51
4.6.1. 系统安全设计 51
4.6.2. 系统监控设计 52
4.6.3. 可扩充性设计 53
4.6.4. 可靠性设计 54
5. 信用卡申请反欺诈系统实现与验证 55
5.1. 关键功能实现 55
5.1.1. 规则管理 55
5.1.2. 欺诈查询 69
5.1.3. 统计报表 76
5.2. 关键功能验证 83
5.2.1. 验证用例 83
5.2.2. 功能验证 84
5.2.3. 验证结论 86
6. 结 论 88
6.1. 总结 88
6.2. 改进建议 88
参考文献 89
致 谢 91
攻读学位期间发表的学术论文目录 92

1. 绪 论

随着银行信用卡业务的发展，由于信用卡的方便、快捷、安全的结算方式，受到消费者的普遍使用。随着信用卡的发卡量的增加，信用卡欺诈风险是银行信用卡业务面临的一个严重的问题，具有隐蔽性强、集团性高、损失金额大等特点。信用卡欺诈主要包括信用卡申请欺诈、丢失、被盗、伪造、机密信息被盗、邮寄被盗等。归纳总结都是通过欺诈性的申请或使用信用卡来实现的。

信用卡申请欺诈是欺诈分子盗取或仿冒他人身份信息，如姓名、性别、出生日期、身份证号码、家庭地址、公司名称、公司地址等信息来申请银行信用卡。信用卡的申请可以通过客服电话、邮政信件、官方网站等渠道进行，信用卡申请审核人员不能面对面的核对信用卡申请人的相关证件，即使面对面的审核，如果证件是伪造的，也可以成功通过信用卡申请的审批。所以欺诈分子就可能成功地得到信用卡。信用卡申请欺诈让银行带来巨大的经济损失，给被冒名的他人带来信用纠纷。

反欺诈是银行风险管理的一个重要组成部分。反欺诈的挑战性是巨大的，因为许多欺诈是有组织的犯罪团伙行为，而且欺诈手段在不断地演变之中。所以信用卡的反欺诈往往利用先进的评分模型和计算机系统，并结合其他行之有效的手段。

1.1. 论文主要工作

本论文通过实际参与项目所从事的工作，分析了国内银行在建设信用卡申请反欺诈系统过程中所涉及的一些问题。首先从业务的角度概括的总结了信用卡申请欺诈的特点，并与国外研究现状进行对比分析；然后提出了信用卡申请欺诈侦测的业务和技术手段。给出一套针对信用卡申请欺诈侦测的解决方案：建立数据集市为欺诈侦测提供良好的数据基础；搭建规则引擎平台能够正确的快速的进行欺诈侦测；建立申请欺诈风险评分模型为欺诈侦测提供量化参考信息；使用基于规则的中文地址分词模糊匹配技术来完善黑名单的检查。

对银行信用卡申请欺诈侦测过程所在的信用卡申请审核流程需求、欺诈侦测需求和系统管理功能需求进行分析。在需求分析基础上进行了信用卡申请反欺诈系统的系统架构设计、应用结构设计，系统功能设计、系统非功能设计。然后进行了子系统的关键功能实现。

1.2. 论文结构安排

本文共分六章，内容安排如下:

第一章 绪论，介绍本论文的背景、论文主要内容以及论文的结构安排。

第二章 信用卡申请反欺诈系统概述。介绍目前国内外的研究现状；介绍反欺诈的侦测手段；介绍模型和规则的开发流程；介绍相关技术知识。

第三章 信用卡申请反欺诈系统需求分析。主要介绍了信用卡申请反欺诈系统在整个信用卡申请审批的流程中与其他系统的交互的流程；着重描述了欺诈侦测的需求以及系统管理功能需求。

第四章 信用卡申请反欺诈系统设计。首先进行系统总体架构设计、应用结构设计；然后分子系统进行设计说明。

第五章 信用卡申请反欺诈系统实现。对各个子系统的关键功能实现进行系统描述。

第六章 结束语。对本文工作进行全面总结，给出本文所取得的成果，指出存在的不足和改进方向。

2. 信用卡申请反欺诈系统概述

2.1.研究现状

2.1.1. 国外研究现状

从欧美的信用卡实践[1]来看，欺诈损失是总额是巨大的；比如2004年美国信用卡行业的欺诈总损失为7亿美元，英国信用卡行业的欺诈总损失额为5亿英镑。

欧美信用卡行业在长期的实践摸索制定的申请欺诈侦测策略——以欺诈风险评分模型为基础进行信用卡申请欺诈风险防范。信用卡申请欺诈风险评分模型是以信用卡申请信息与信用局记录的信用信息进行对比来预测欺诈概率。

在以信用卡申请欺诈风险评分模型计算的欺诈概率预测结果为基础，选择欺诈的防范措施：对于欺诈概率特别高的信用卡申请予以拒绝；对于欺诈率比较高但不算特别高的信用卡申请，可以进行额外的人工电话调查申请人身份真假和人工核对的申请人相关证件等，然后再人为的作决定是否批准申请；对于欺诈率较低的信用卡申请，则直接予以批准。

在使用信用卡申请欺诈风险评分模型进行欺诈概率预测之外，还可以利用规则来实现法规和政策方面的申请欺诈侦测策略：对于欺诈率高发的地区、行业、机构和年龄段等信用卡申请可以适当提高审批的门槛；对于申请信息的某些有相互关联的信息进行信息的一致性检查，如果不一致提醒进行人工调查；对于同一申请人在固定的时间段申请次数超过某一设定值的予以拒绝等。

2.1.2. 国内研究现状

国内的银行信用卡申请和审批过程中，其中的申请过程的欺诈形式多种多样。我国银行信用卡申请欺诈主要有以下特点[2]:

(1) 个人申请欺诈范围广

欺诈者通过编造虚假的个人身份信息来制作虚假的证明，向银行欺诈的申请信用卡。由于欺诈的信用卡申请者提供的是虚假身份证明，造成损失后银行将难以催收追索。

(2) 单位申请欺诈密度高

小企业由于融资十分困难，往往利用银行降低办信用卡标准的机会，在员工不知情的情形下，给员工申办信用卡由单位集中激活使用。当企业经营出现困难时，就无法还清信用卡的透支，这样就给银行造成重大经济损失。

(3) 中介公司申请欺诈危害多

中介公司为信用卡申请人提供办卡、套现协助。此类机构组织严密，分工明确，了解银行信用卡申请审批流程，可以规避信用卡申请的审批的策略，会产生批量信用卡申请的欺诈风险。

(4) 犯罪集团申请欺诈损失大。

信用卡申请集团化的欺诈是以集团性的骗取银行资金为最终目的，批量的冒用他人的身份信息欺诈的申请银行信用卡，会给银行带来巨大的经济损失。

(5) 银行预制卡被盗用影响坏

银行会为既有优质客户预制高额度信用卡。由于客户信息对内部人员不是秘密，银行内部人员有机可乘。会窃取卡片激活使用，给银行造成很坏的影响和损失。

2.1.3. 国内外研究对比

对国内和国外（以美国为例）的信用卡申请业务中的欺诈风险防范对比[3]，我将从以下三个方面对比：

(1) 征信系统

美国的健全及完善的征信系统，它们包含了巨量的信息。主要的三大征信局为：Experian, Equifax, Trans Union，数据库覆盖了个人和企业信用信息。以Experian为例，其数据库中包含2200万单位企业信息和2.2亿单位的个人信息。征信信息的内容包括企业和个人的注册名、历史地址信息、征税号、电话号码等等。

相对于美国的健全的征信系统，我国的个人和企业信用记录系统缺少信息的系统化和全面化。国内主要的征信系统是中国人民银行征信中心提供的信用报告，个人的信用报告内容个人基本信息、信贷交易信息和公共信息等。

(2) 欺诈风险模型和规则

美国有成熟的量化的欺诈风险模型对潜在的欺诈风险进行打分，欺诈风险

模型由征信提供商负责维护，定期的对欺诈风险模型进行优化分析以保证准确率。根据风险模型的评分和市场情况，信用卡公司可灵活调整规则策略。

国内的银行使用欺诈模型评分和欺诈规则对信用卡申请件进行欺诈侦测，给出欺诈风险点提示。欺诈模型和规则由银行欺诈调查团队负责维护，依据监控欺诈模型和规则的表现情况，对欺诈模型和规则进行优化。

(3) 人工电话调查

美国信用卡公司以一个欺诈风险模型评分为起点进行更详尽的人工调查和电话核查。对于低于这个起点的信用卡申请件完全可以实现自动审批。

国内银行的信用卡申请审批人员依据欺诈侦测对关键风险点的提示，使用人工电话征信调查和第三方信息核查等手段排查欺诈风险。最终由审批人员对信用卡申请进行终审。

2.2.信用卡欺诈侦测手段

信用卡申请欺诈侦测的目标是建立信用卡申请欺诈侦测的业务和技术手段，自动将可能为欺诈的申请筛选出来，供审批人员作为决策依据；提高欺诈调查的效率与欺诈风险识别率；更加及时有效的进行反欺诈工作，全面降低欺诈风险，最大限度的减少银行欺诈损失。

2.2.1. 业务手段

信用卡申请欺诈侦测的业务手段主要包括以下方面:

(1) 人行征信

查询人行征信系统的数据来判断申请人的信用状况；

(2) 第三方核查

查询第三方信息系统（公安部门的身份证信息系统、电话黄页信息系统等）的数据来判断申请人的所提供的信息真伪；

(3) 电话核实

银行申请审核人员按照申请表上填写的电话号码联系申请人，确认申请人是否本人申请，如果是本人申请，核对申请表信息与申请人表述的是否一致；

(4) 查询相关信息系统

在银行账务信息系统，查询申请人在本行个人账户信息。

2.2.2. 技术手段

信用卡申请欺诈侦测的技术手段主要包括以下方面:

(1) 建立数据集市

建立信用卡申请欺诈侦测相关的数据集市，对这些银行各种内部和外部数据源进行抽取、加载、转换、检查、存储等处理。这样能够保证数据的完整和一致性，为信用卡申请欺诈侦测提供良好的数据基础;

(2) 建立风险评分模型

建立信用卡申请欺诈风险评分模型来评估信用卡申请的欺诈风险值并持续不断的完善，对信用卡申请欺诈侦测提供量化参考信息;

(3) 搭建规则引擎平台

在规则引擎平台中部署欺诈侦测规则集，能够正确的快速的进行信用卡申请欺诈侦测;

(4) 建立欺诈侦测系统

信用卡申请反欺诈系统对信用卡申请欺诈行为进行侦测。

2.3.信用评分模型开发流程

信用评分模型是利用数据挖掘和统计分析手段服务于信贷管理的先进技术，它的开发有一个科学的标准的流程[4]。评分模型开发流程如图2-1所示：

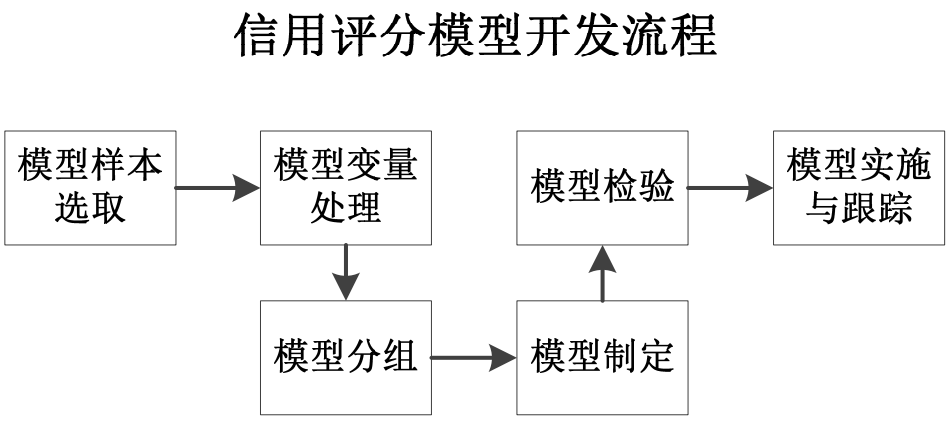

图 2-1 模型开发流程图

Fig.2-1 The flow chart of model development

信用评分模型的开发流程主要包括模型样本获取、模型变量处理、模型分组、模型制定、模型检验、模型实施与跟踪，具体工作内容如下：

(1) 模型样本选取

选取好的样本数据是开发模型的首要环节。建立预测模型的前提是客户未来的行为与过去相似，而且这种客户行为可以用数理统计技术提炼和总结出来，所以可以用过去的客户数据来预测未来客户行为。样本数据能否有效地代表总体，样本数据的质量高低将在很大程度上决定模型的预测能力和效果。

模型样本选取时须注意下列原则：样本必须能够充分的代表总体，不仅代表过去的总体，更要代表未来实施模型的总体；样本的观察期一般是越近越好，但观察期的选择还取决于数据的可获得程度和表现期的长短；样本必须有可靠的预测信息和表现信息。

模型样本主要选取方法有随机抽样和分类抽样两种：

- 随机抽样

在确定样本规模后从总体中完全随机的抽取，每种类型的个体在样本中的比例与在总体中的比例是一样的；

- 分类抽样

根据模型的需要确定样本的类别，对每一类别的样本分别确定其抽样个数，然后在每一类别内部随机抽取所需的样本。

(2) 模型变量处理

模型变量处理主要包括表现变量的界定和预测变量的提炼：

- 表现变量的界定

表现变量是模型所要预测的目标，在有些情况下其界定是直观的，但在更多情况下表现变量的界定是不直观的，需要在仔细衡量个方面因素后才能确定。

- 预测变量的提炼

预测变量的提炼是开发评分模型非常重要的一环，从观察期的大量原始信息里，往往可以提炼出成百上千个具有一定预测能力的变量。

(3) 模型的分组

模型分组是模型开发流程中的常用且重要的一环，它是把总体数据分组，每一组内的数据具备同质性，而不同组之间的数据具备不同性质。分组的依据

是可观察到的行为信息和预测变量，而不是目标变量。分组的目标是按照不同行为模型和数理关系归类，以提高模型预测力。分组的方法是直觉、经验和数据分析的综合结果。

(4) 模型的制定

模型的制定主要包括以下几个方面：

- 分析单个标量的预测能力

  发现具有预测能力变量作为模型的候选变量，剔除不具备预测能力变量，以缩小候选变量的范围。

- 减少候选变量的数量

  把候选变量进一步分组，每一组之内变量间的相关性和高，组与组之间变量相关性很低。然后从每一组变量中选择预测力最强的一个变量作为模型候选变量。

- 选择适当的模型方法

  对于二元性结果的预测，最流行的模型方法是逻辑回归模型[5], [6]和神经网络模型[7], [8]。逻辑回归模型在信用卡申请欺诈风险评分模型中广泛应用。

- 确定模型的变量组合和权重

  选择一定的变量组合，进入最终的模型，并根据统计原理分配相对应评分权重。

(5) 模型的检验

模型的检验有样本内检验和样本外检验两种方式：

- 样本内检验

  是利用模型开发所使用的样本来对比预测请况与实际情况的差；

- 样本外检验

  是利用没有用于模型开发的样本来对比预测情况与实际情况的差别。

(6) 模型实施与跟踪

模型实施主要是将批准的模型部署到信用卡申请欺诈侦测中。然后通过分析模型监控跟踪报表来判断模型的表现情况，及时发现问题并进行优化。

2.4.相关技术知识

2.4.1. 规则引擎

规则引擎[9]由推理引擎发展而来，使用规则引擎可以将业务决策部分从传统的源代码中分离，并使用预定义语言来编写业务决策部分。根据业务数据和预先定义的业务规则进行业务决策。

规则引擎都支持规则冲突检验和规则的次序，使用简单脚本语言的进行业务规则定义，支持通用开发语言的嵌入开发。

(1) 规则引擎算法

目前大部分商用和开源的规则引擎产品都使用Rete算法。它是Dr. Charles Forgy在1979年提出的针对基于规则知识表现的模式匹配算法，其核心思想是将分离的匹配项根据内容动态构造匹配树，以达到显著降低计算量的效果。是目前效率最高的推理算法，

Rete算法[11], [12], [13]是一种前向规则快速匹配算法，其匹配速度与规则数目无关。Rete算法是通过形成rete网络进行模式匹配，利用基于规则系统的两个特征：时间冗余性和结构相似性，提高模式匹配效率。

相关概念如下：

- 事实(fact)

  对象之间及对象属性之间的多元关系。

- 规则(rule)

  由条件和结论构成的推理语句，当存在事实满足条件时，相应结论被激活。

- 模式(patten)

  规则的 IF 部分，已知事实的泛化形式，未实例化的多元关系。

- 模式匹配算法

  规则主要由两部分组成：

  1) 条件，也称为左端(记为 LHS, left-hand side)，

  2) 结论，也称为右端(记为 RHS, right-hand side)。

  规则匹配，就是对规则 r，判断当前的事实 o 是否使 LHS(r)=True，如

果是，就把规则 r 的实例 r(o)加到冲突集当中。

- Rete 网络

Rete 算法的编译结果是规则集对应的 Rete 网络。Rete 网络是一个事实可以在其中流动的图。Rete 网络的节点可以分为四类：根节点、类型节点、alpha 节点、beta 节点。其中，根结点是一个虚拟节点，是构建 Rete 网络的入口。类型节点中存储事实的各种类型，各个事实从对应的类型节点进入 Rete 网络。

(2) 规则引擎推理

规则引擎的架构如图2-3所示：

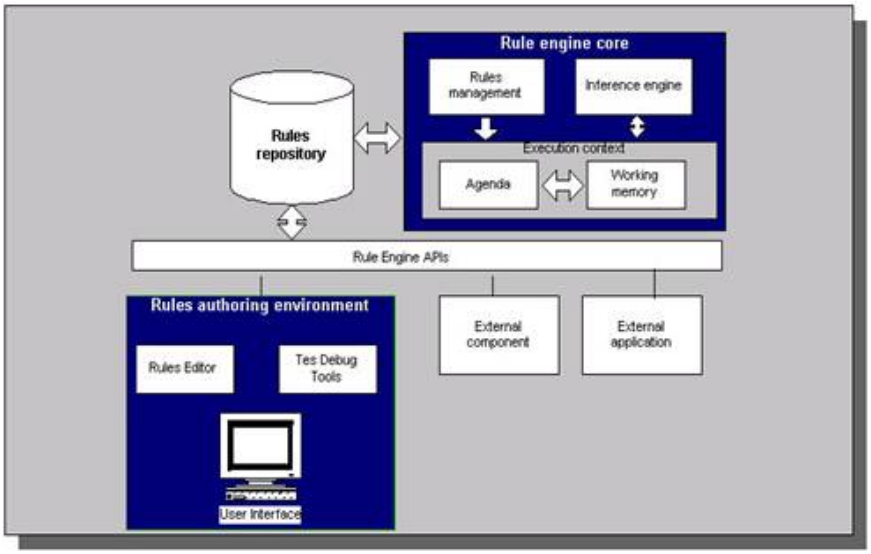

图 2-2 规则引擎的架构

Fig.2-3 The framework of rule engine

规则引擎推理过程：

- 将数据加载到工作内存中；
- 使用模式匹配程序将规则库中的规则和数据比较；
- 如果存在规则冲突，即激活多个规则，将冲突的规则放入冲突规则集合；
- 将冲突规则按照既定策略解决，将激活规则按顺序全部放入规则执行队列；
- 将规则执行队列中规则执行；
- 重复上述步骤，直到规则执行队列中所有规则执行完成。

当规则引擎执行时，根据规则执行队列中规则的优先顺序执行规则实例，由于在规则执行过程中可能会对数据进行改变，会使规则执行队列中的某些规则实例失效，从而必须在规则执行队列中将这些失效的规则进行撤销，也有可能会激活不在规则执行队列中的其他规则，规则执行队列将会增加新的规则实例。这种动态的规则执行队列，形成规则的推理机制。这种规则执行队列的动态反应完全是由数据进行驱动的。

规则引擎的性能取决于规则条件匹配的效率，规则引擎需要对工作区中的数据进行迅速比较，发现符合条件的在规则引擎加载的规则集中的规则，生成规则实例放入规则执行队列中。

(3) 规则引擎的使用规范(JSR-94)

JSR-94[14]是Java规则引擎API的Java规范，由Java Community Process (JCP)制定。Java规则引擎API是访问规则引擎的标准企业级API。Java规则引擎API使用统一的规则引擎调用不同厂商或者开源的规则引擎产品。Java规则引擎API分为两个主要部分：

- 规则管理 API (the rules administration API)

  包括装载规则以及与规则对应的动作以及实例化规则引擎。

- 运行时 API (the Runtime client API)

  为运行规则及获得结果提供了类和方法。

目前业内有多个支持JSR-94规范规则引擎可供使用，其中包括商业和开放源码选择。开源的代表是Drools(JBoss Rules) [15]，商业的代表是ILOG公司的JRules[16]，BlazeSoft公司的Blaze[17]。

2.4.2. 基于规则的中文地址分词模糊匹配

基于规则的中文地址分词模糊匹配[18]是在建立标准地址库的基础上使用自定义的地址匹配规则进行中文地址的模糊匹配。该方法在依据标准地址库分词的同时,使用自定义的地址匹配规则进行推理,从而缩小了下次分词所用到的目标数据集,提高了系统执行效率。另外,通过借助构建的规则树与分词的权重和阀值,提高了中文地址模糊匹配的成功率。

汉字模糊匹配技术在反身份欺诈系统中可以处理系统中多数据源间的地址比对，将地址模糊比对的结果应用在反身份欺诈的规则或者评分卡中；并且通过扩展其使用功能，达到全面反身份欺诈的效果。例如将汉字模糊匹配技术应用于黑名单检查。

针对现有的黑名单进行分析，并建立广义黑名单，使其不仅包括姓名、证件类型和证件号码，还要包括地址、单位名称、电话等重要内容，并建立相应的数据存储机制，以便将来使用。利用中文模糊匹配技术将申请件内容与黑名单内容相匹配，以期获得比精确匹配更多的匹配结果，从而提高捕获欺诈的效率。

基于规则的中文地址分词模糊匹配流程如下：

(1) 地址预处理

进行处理主要包括：中文大小写数字和阿拉伯数字转换处理、简繁体转换、全角半角转换、特殊字符处理、关键字处理。

(2) 地址分词解析

依据分级结构的地址库的分词解析的地址如表2-1所示（示例）：

表 2-1 地址分词解析

Table2-1 Segmentation analysis of address

行政区划	道路	门牌号	地标名称	楼号	室号
上海市浦东新区	源深路	200 号	信用卡中心	10 号楼	101 室

(3) 对分词地址进行匹配

- 行政区划段：精确匹配。
- 道路段在道路库中找到，则精确匹配，否则，模糊匹配
- 地标名称与道路门牌的含义是重复的，建立地标库可做到精确匹配。
- 楼号+层号+室号：模糊匹配

(4) 分词地址匹配度计算

利用匹配结果和权重的组合算法进行计算，最终得到两地址的匹配度（相似度）。计算公式如表2-2所示：

表 2-2 地址匹配度计算公式

Table2-2 The calculation formula of address matching degree

	行政区划	道路	牌号	地标名称	楼号	室号
地址 1	A1	B1	C1	D1	E1	H1
地址 2	A2	B2	C2	D2	E2	H2
权重	35%	25%		20%	20%	
匹配度						

3. 信用卡申请反欺诈系统需求分析

3.1. 信用卡申请审核流程

信用卡申请欺诈侦测作为银行信用卡申请审核流程中的一个环节。具体流程如图 3-1 所示：

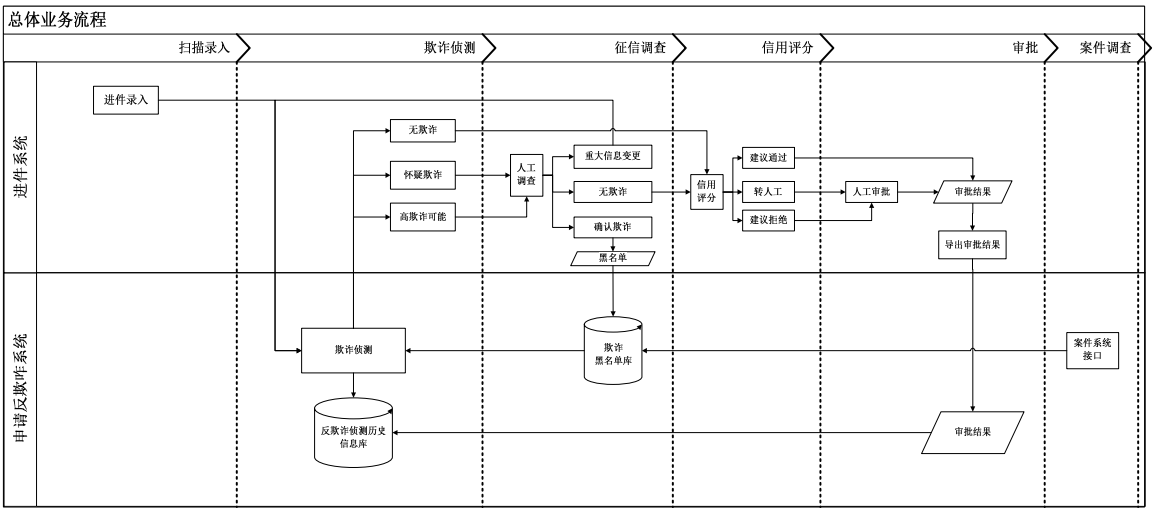

图 3-1 信用卡申请审核业务流程

Fig.3-1 The business process of credit card application process

业务流程说明：

(1) 申请欺诈系统服务接口接收信用卡审批系统的申请件信息；

(2) 将接收到的申请数据进行数据转换，然后将申请数据送至申请反欺诈检查；

(3) 反欺诈检查处理：

- 先对申请件进行黑名单检查，将结果提供给评分模型以及规则集处理；
- 然后对该申请件进行模型评分，并将评分结果提供给规则处理；
- 最终规则处理将结果返回至信用卡审批系统；
- 由信用卡审批系统根据欺诈侦测结果进行相关业务决策。

3.2.欺诈侦测需求

欺诈侦测主要是对信用卡申请数据进行黑名单检查、评分模型计算、规则处理。主要处理流程如图 3-2 所示：

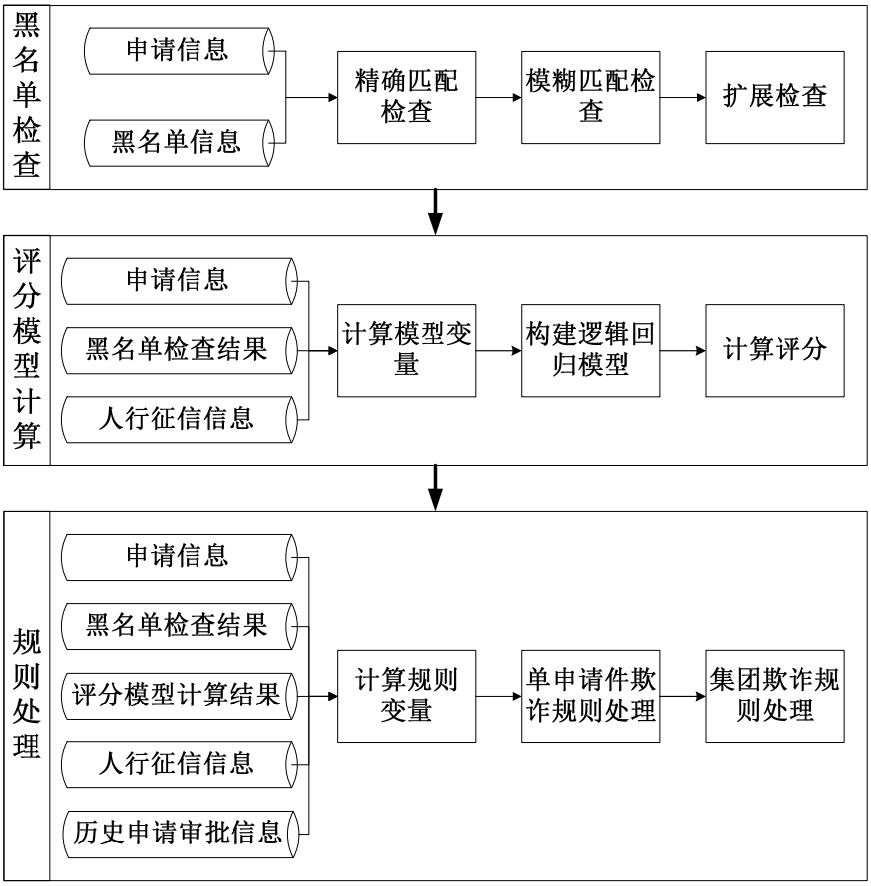

图 3-2 信用卡申请欺诈侦测业务流程

Fig.3-2 The business process of credit card application fraud detection

处理流程说明：

(1) 黑名单检查

提供广义的黑名单检查功能。黑名单检查是以规则的形式在规则引擎中运行。将黑名单检查结果传递给下一步评分模型计算进行后续处理；

(2) 评分模型计算

评分模型的计算是在规则引擎中完成计算。主要是将输入变量传输到规则引擎，结合评分模型计算模型评分结果，将模型评分结果传递给下一步规则处理进行后续处理；

(3) 规则处理

规则处理的执行是调用规则引擎完成，主要是结合不同的规则处理，适时将规则变量传输到决策引擎，并结合规则进行判断，给出最终的建议结果。

3.2.1. 黑名单检查

申请反欺诈系统提供的是广义黑名单检查功能。在对申请人信息与黑名单精确比对基础上使用基于规则的中文地址分词模糊匹配技术扩展了黑名单检查的范围，并且这些黑名单信息在实现黑名单过滤的过程中可以指定数据项的相似度，从而调节黑名单过滤的数据。

(1) 原始数据

黑名单检查所需的原始数据为申请信息和黑名单信息。主要包含信息为：

- 申请信息

主要包括：姓名、证件类型、证件号码、家庭电话、家庭地址、单位名称、单位电话、单位地址等信息；

- 黑名单信息

黑名单信息是扩展了的黑名单，主要包括身份（姓名，证件号码，证件类型组合）黑名单、电话黑名单、地址黑名单、单位名称黑名单等。

(2) 黑名单检查规则

黑名单检查的规则如下：

- 精确匹配检查

对申请人的身份（姓名，证件号码，证件类型组合）信息与黑名单比对，给出结果是明确是否的在黑名单中；

对于申请人的电话类（例如家庭电话、单位电话）信息与黑名单比对，给出结果是明确是否的在黑名单中。

- 模糊匹配检查

对于申请人的地址类（例如家庭地址、单位地址）的信息与黑名单进行模糊比对，给出的的结果是与黑名单中数据的最高的相似度。

对于申请人的单位名称类的信息与黑名单进行模糊比对，给出的的结果是与黑名单中数据的最高的相似度。

- 扩展检查

黑名单的扩展检查主要是对于申请人的电话类（例如住宅电话、单位电话）信息，首先查询固定电话运行商提供的电话相关的信息获得电话安装的地址，然后将查询的得到的固定电话安装地址在与黑名单进行模糊比对，给出的的结果是与黑名单中数据的最高的相似度。

(3) 输出结果

黑名单检查输出结果如下：

- 身份（姓名，证件号码，证件类型组合）信息、家庭电话、单位电话是否在黑名单中；
- 家庭地址、单位地址、单位名称在黑名单中匹配的相似度。

3.2.2. 评分模型计算

评分模型采用类型为“模型规则”的规则进行实现。部署到规则引擎中后，运行规则进行评分模型计算。

评分模型计算主要过程：首先依据原始数据计算模型预测变量，然后依据评分模型、模型预测变量及权重计算出模型评分。

(1) 原始数据

评分模型计算所需原始数据主要包括申请信息、人行征信信息和黑名单检查结果：

- 申请信息

主要包括：姓名、证件类型、证件号码、手机号码、家庭电话、家庭地址、出生日期，学历，单位名称、单位电话、单位地址等信息；

- 人行征信信息

主要包括：姓名、证件类型、证件号码、手机号码、家庭电话、家庭地址、出生日期，学历，单位名称、单位地址等信息；

- 黑名单检查结果

主要包括：身份（姓名+证件号码+证件类型组合）信息、家庭电话、单位电话是否在黑名单中；家庭地址、单位地址、单位名称在黑名单中匹配的相似度。

(2) 模型预测变量

- 是否存在人行征信信息

使用申请信息中的姓名+证件号码+证件类型组合是否查询到人行征信信息；

- 手机号码是否相符

申请信息中的手机号码与人行征信信息中的手机号码是否精确匹配；

- 家庭电话是否相符

申请信息中的家庭电话与人行征信信息中的家庭电话是否精确匹配；

- 单位电话是否相符

申请信息中的单位电话与人行征信信息中的单位电话是否精确匹配；

- 学历是否相符

申请信息中的学历与人行征信信息中的学历是否精确匹配；

- 家庭地址是否相符

申请信息中的家庭地址与人行征信信息中的最近更新的家庭地址进行模糊匹配，当相似度达到指定值时，视为相符；

- 单位名称是否相符

申请信息中的单位名称与人行征信信息中的最近更新的单位名称进行模糊匹配，当相似度达到指定值时，视为相符；

- 单位地址是否相符

申请信息中的单位地址与人行征信信息中的最近更新的单位地址进行模糊匹配，当相似度达到指定值时，视为相符；

- 黑名单精确匹配

姓名，证件号码，证件类型组合）信息、家庭电话、单位电话是否在黑名单中；

- 黑名单模糊匹配

1） 家庭地址在黑名单中匹配的相似度是否达到指定值；

2） 单位地址在黑名单中匹配的相似度是否达到指定值；

3） 单位名称在黑名单中匹配的相似度是否达到指定值。

(3) 评分模型计算

使用逻辑回归模型、预测变量及其权重进行模型评分计算，得出评分模型计算结果。计算过程如下：

1) 选择一个模型预测变量；

2) 将模型预测变量使用逻辑回归模型进行欺诈预测；

3) 得到预测的欺诈概率；

4) 重复 1）- 3）步骤，使所有的模型预测变量获得预测的欺诈概率；

5) 计算模型评分：各个预测变量的欺诈概率与权重乘积的求和。

(4) 输出结果

评分模型计算结果：模型评分。

3.2.3. 规则处理

规则处理主要是进行政策性的判断，采用类型为“校验规则”的规则来实现。部署到规则引擎中后，运行规则进行规则处理，并给出规则判断建议结果。

(1) 原始数据

规则处理所需原始数据主要包括申请信息、人行征信信息、评分模型计算结果、黑名单检查结果：

- 申请信息

主要包括：姓名、证件类型、证件号码、手机号码、家庭电话、家庭地址、出生日期，学历，工作年限，单位名称、单位电话、单位地址等信息；

- 人行征信信息

主要包括：姓名、证件类型、证件号码、手机号码、家庭电话、家庭地址、出生日期，学历，单位名称、单位地址等信息；

- 评分模型计算结果

主要包括：模型评分；

- 黑名单检查结果

主要包括：身份（姓名+证件号码+证件类型组合）信息、家庭电话、单位电话是否在黑名单中；家庭地址、单位地址、单位名称在黑名单中匹配的

相似度。

- 历史申请审批信息

主要包括：姓名、证件类型、证件号码、手机号码、家庭电话、家庭地址、单位名称、单位电话、单位地址、审批结果等信息；

(2) 校验规则

- 模型评分结果检查规则

依据评分模型计算结果值所在的区间，确定欺诈侦测建议结果。校验如下：

1) 当模型评分小于设置的最低阀值时，建议结果为“无欺诈”；

2) 当模型评分大于设置的最低阀值，并且小于设置的最高阀值时，建议结果为“怀疑欺诈”，触发规则组增加“模型评分结果检查规则”，对应原因码为“A001”；

3) 当模型评分大于设置的最高阀值时，建议结果为“高欺诈可能”，触发规则组增加“模型评分结果检查规则”，对应原因码为“A002”；

- 人行征信信息检查规则

检查人行征信信息中的信贷交易信息和公共信息是否存在不良记录。校验如下：

1) 当信贷交易信息中存在逾期或透支信息时，建议结果为“高欺诈可能”，触发规则组增加“人行征信信息检查规则”，对应原因码为“B001”；

2) 当公共信息存在不良记录时，建议结果为“高欺诈可能”，触发规则组增加“人行征信信息检查规则”，对应原因码为“B002”。

- 客户黑名单检查规则

检查申请信息中客户基本信息是否在黑名单中。校验如下：

1) 当姓名，证件号码，证件类型组合信息在黑名单中时，建议结果为“高欺诈可能”，触发规则组增加“客户黑名单检查规则”，对应原因码为“C001”；

2) 当家庭电话在黑名单中时，建议结果为“高欺诈可能”，触发规则组增加“客户黑名单检查规则”，对应原因码为“C002”；

3） 当家庭地址在黑名单中匹配的相似度大于阀值时，建议结果为“高欺诈可能”，触发规则组增加“客户黑名单检查规则”，对应原因码为“C003”。

- 单位黑名单检查规则

检查申请信息客户相关的单位信息是否在黑名单中。校验如下：

1） 当单位电话在黑名单中时，建议结果为“高欺诈可能”，触发规则组增加“单位黑名单检查规则”，对应原因码为“D001”；

2） 当单位地址在黑名单中匹配的相似度大于阀值时，建议结果为“高欺诈可能”，触发规则组增加“单位黑名单检查规则”，对应原因码为“D002”；

3） 当单位名称在黑名单中匹配的相似度大于阀值时，建议结果为“高欺诈可能”，触发规则组增加“单位黑名单检查规则”，对应原因码为“D003”。

- 历史申请检查规则

使用申请信息的客户身份（姓名，证件号码，证件类型组合）信息的在历史申请审批信息检查。校验如下：

1） 使用客户身份信息在历史申请审批信息进行统计。当历史申请审批信息存在拒绝申请时，建议结果为“怀疑欺诈”，触发规则组增加“历史申请检查规则”，对应原因码为“E001”。

- 年龄检查规则

检查申请信息中的年龄是否符合规定。校验如下：

1） 当年龄小于 18 时，建议结果为“怀疑欺诈”，触发规则组增加“年龄检查规则”，对应原因码为“F001”。

- 集团欺诈检查规则

申请信息与历史的申请审批信息使用精确或模糊匹配方法，对进行地址、单位名称、电话等等进行统计，当统计数量大于阀值时，视为存在集团欺诈。校验如下：

1） 申请信息中的地址与历史申请审批信息中所有申请件的地址进行模糊匹配，对相似度大于阀值进行统计。当统计数量小于设置的最低阀值时，建议结果为“无欺诈”；当统计数量大于设置的最

低阀值，并且小于设置的最高阀值时，建议结果为“怀疑欺诈”，触发规则组增加“集团欺诈检查规则”，对应原因码为“G001”；当统计数量大于设置的最高阀值时，建议结果为“高欺诈可能”，触发规则组增加“集团欺诈检查规则”，对应原因码为“G002”；

2) 申请信息中的单位名称与在历史申请审批信息所有申请件的单位名称进行模糊匹配，对相似度大于阀值进行统计。当统计数量小于设置的最低阀值时，建议结果为“无欺诈”；当统计数量大于设置的最低阀值，并且小于设置的最高阀值时，建议结果为“怀疑欺诈”，触发规则组增加“集团欺诈检查规则”，对应原因码为“G003”；当统计数量大于设置的最高阀值时，建议结果为“高欺诈可能”，触发规则组增加“集团欺诈检查规则”，对应原因码为“G004”；

3) 申请信息中的电话与在历史申请审批信息中所有申请件的电话进行精确匹配统计数量。当统计数量小于设置的最低阀值时，建议结果为“无欺诈”；当统计数量大于设置的最低阀值，并且小于设置的最高阀值时，建议结果为“怀疑欺诈”，触发规则组增加“集团欺诈检查规则”，对应原因码为“G005”；当统计数量大于设置的最高阀值时，建议结果为“高欺诈可能”，触发规则组增加“集团欺诈检查规则”，对应原因码为“G006”；

4) 查询申请信息中的电话的相关信息，获得电话安装地址。使用电话安装地址与历史申请审批信息中所有申请件的地址进行模糊匹配，对相似度大于阀值进行统计。当统计数量小于设置的最低阀值时，建议结果为“无欺诈”；当统计数量大于设置的最低阀值，并且小于设置的最高阀值时，建议结果为“怀疑欺诈”，触发规则组增加“集团欺诈检查规则”，对应原因码为“G007”；当统计数量大于设置的最高阀值时，建议结果为“高欺诈可能”，触发规则组增加“集团欺诈检查规则”，对应原因码为“G008”。

(3) 输出结果

规则处理结果主要包括：建议结果、欺诈模型评分、触发规则组及对应原因码。

3.3.系统管理功能需求

3.3.1. 规则管理

规则管理[10]包含规则编辑、规则验证和规则审批等功能。

(1) 规则编辑

规则变更主要包括规则创建、规则修改和规则优化三个功能：

- 规则创建是创建新的规则；
- 规则修改是对已有的规则进行维护；
- 规则优化是通过分析规则监控跟踪报表判断规则的表现情况，及时发现规则所存在的问题并进行优化。

(2) 规则验证

编辑完成的规则需要进行规则验证。规则验证可以对单个规则进行验证，也可以对整个规则流程进行验证。如果规则验证通过才可提交进行规则审批；如果验证不通过就必须进行规则修改后再次规则验证。

(3) 规则审批

反欺诈系统提供规则审批的功能，主要是将通过验证的规则进行审批。如果审批不通过就必须进行规则修改后再次规则验证和审批；如果规则审批通过规则部署到决策引擎中。

3.3.2. 欺诈查询

欺诈查询主要是针对信用卡申请欺诈侦测结果的查询，包括申请欺诈查询和集团欺诈查询功能。具体需求如下：

(1) 申请欺诈查询

对申请欺诈侦测结果进行查询。以列表的形式展现信息。

查询条件：申请人姓名、建议结果、日期。

查询结果信息：申请件编号、姓名、证件号码、建议结果、模型评分、触发规则、原因码、审批结果、日期。

(2) 集团欺诈查询

对集团欺诈侦测结果进行查询。以列表的形式展现信息。

查询条件：单位名称、地址、电话、日期。

查询结果信息：申请件编号、姓名、证件号码、单位名称、地址、电话、日期。

3.3.3. 统计报表

通过统计报表分析欺诈侦测的表现情况以及反欺诈推荐结论的质量情况。保镖需求如下：

(1) 规则监控统计报表

通过分析规则监控统计报表来判断规则的表现情况。规则监控统计报表如表 3-1 所示：

表 3-1 规则监控统计报表

Table3-1 The page of rule monitor report

规则	触发量	触发率	命中率	平均触发率	平均命中率
A01					
A02					
A03					
A04					
……					

报表指标定义如下：

- 触发量

  指标定义：触发每条规则的件数

  计算公式：统计汇总触发每条规则的件数

- 规则触发率

  指标定义：申请欺诈系统每天触发规则进件数占总进件量的比例。

  计算公式：规则触发率＝触发规则数/进件量*100%。

- 规则命中率

指标定义：审核人员对触发规则的申请件调查后确认为欺诈或疑似欺诈的件数占调查总件数的比例。

计算公式：规则命中率＝前端确认卡片数/调查数*100%。

(2) 欺诈管理统计报表

通过申请欺诈率，可以直接反映进件质量。而通过命中率，可以为申请欺诈系统规则的有效性提供依据，直接反映申请欺诈系统规则设置、管理水平，间接反映审核人员调查能力程度。欺诈管理统计报表如表 3-2 所示：

表 3-2 欺诈管理统计报表

Table3-2 The page of fraud report

指标	1 月	2 月	……	12 月
申请总数量				
怀疑欺诈申请数				
怀疑欺诈申请比率				
高欺诈可能申请数				
高欺诈可能申请比率				
调查确认欺诈申请数				
调查确认非欺诈申请数				
调查命中率				
调查命中率基准				

指标定义如下

- 申请总数量

指标定义：当月申请件总数。

计算公式：当月申请欺诈侦测总件数。

- 怀疑欺诈申请数

指标定义：当月申请拒绝原因为怀疑欺诈的件数。

计算公式：汇总批核标识为“怀疑欺诈”申请件数。

- 怀疑欺诈申请比率

指标定义：当月怀疑欺诈件数占当月申请件总数的比例。其中当月怀疑欺诈件数是当月拒绝原因为怀疑欺诈的件数， 当月申请件总数是申请欺诈侦测总件数。

计算公式：比率=当月怀疑欺诈件数/当月申请件总数*100%。

- 高欺诈可能申请数

指标定义：当月申请拒绝原因为高欺诈可能的件数。

计算公式：汇总批核标识为“高欺诈可能”申请件数。

- 高欺诈可能申请比率

指标定义：当月高欺诈可能申请件数占当月申请件总数的比例。其中当月高欺诈可能申请件数是当月拒绝原因为高欺诈可能的件数， 当月申请件总数是申请欺诈侦测总件数。

计算公式：比率=当月高欺诈可能申请件数/当月申请件总数*100%。

- 调查确认欺诈数

指标定义：审核人员对需要调查的申请件调查后确认为欺诈的件数。

计算公式：确认为欺诈的件数，即申请欺诈系统中标识为“确认欺诈”的申请件。

- 调查确认非欺诈数

指标定义：审核人员对需要调查的申请件调查后确认为非欺诈的件数。

计算公式：确认为非欺诈的件数，即申请欺诈系统中标识为“非欺诈”的申请件。

- 调查命中率

指标定义:审核人员对需要调查的申请件调查后确认为欺诈的件数占调查总件数比。

计算公式：当月确认为欺诈的件数/当月总调查的申请件数*100%。

4. 信用卡申请反欺诈系统设计

4.1.系统设计原则

系统设计将遵循以下原则

(1) 前瞻性

要对信用卡申请反欺诈的发展前瞻性分析，做好相应规划和实施变更。

(2) 与市场发展同步

系统必须对这些市场需求有较高的适应性。

(3) 可操作性：

提供用户WEB访问界面。须以菜单方式提供操作人员使用;菜单设计要合理，菜单分类应按业务种类设置。

(4) 可扩展性：

不仅要满足目前的需要，还要考虑今后业务发展过程中对新需求的适应性。

(5) 安全性：

系统建设必须考虑数据的保密性需求，防止数据的丢失与外泄。密码、证件等关键数据采用加密技术，敏感数据可以根据客户要求进行屏蔽。

(6) 稳定性：

系统运行必须稳定，具备7*24小时的运行能力。

(7) 高效性：

系统建设必须考虑性能需求，满足业务发展的需要。逐层优化解决方案，可以保证系统的综合运行性能。

(8) 异常可查性：

引入审计功能，对关键操作有完善的日志、监控等手段进行跟踪，在出现问题时候可以查找原因。

4.2. 系统架构设计

系统的架构如图4-1所示：

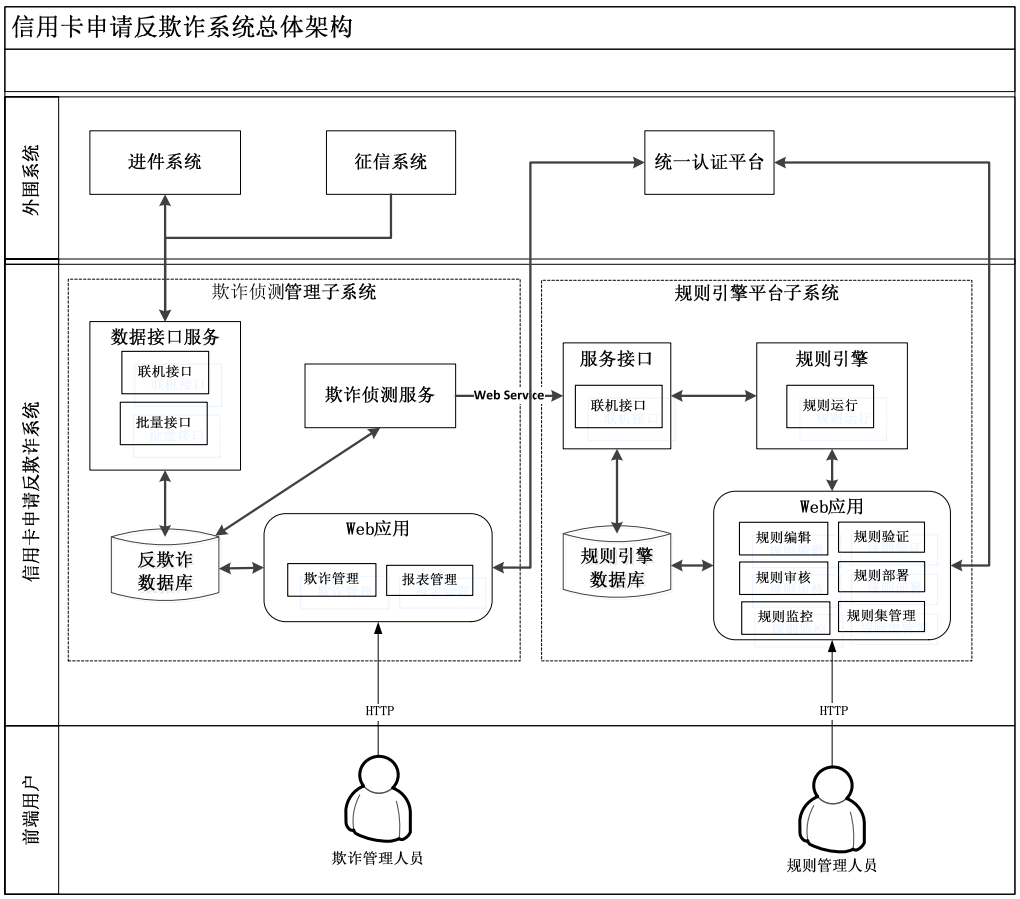

图 4-1 系统架构

Fig.4-1 The architecture of system overall

系统架构说明：

(1) 外围系统

- 进件系统

  是信用卡申请欺诈侦测的请求方；提供申请信息、黑名单信息、历史审批信息；

- 征信系统

  提供申请人的人行征信信息；

- 统一认证平台

  提供用户单点登录及系统权限管理；

(2) 信用卡申请反欺诈系统

- 欺诈侦测管理子系统

  包括数据接口服务、欺诈侦测服务、欺诈管理(Web 应用)、反欺诈数据库等；

- 规则引擎平台子系统

  包括服务接口、规则引擎、规则管理(Web 应用)、规则引擎数据库；

- 子系统接口

  欺诈侦测管理子系统的欺诈侦测服务通过 WebService 方式调用规则引擎平台子系统的服务接口来实现在规则引擎中进行规则处理；

(3) 前端客户

- 欺诈管理人员
- 规则管理人员

4.3. 应用结构设计

系统应用结构设计如图4-2所示：

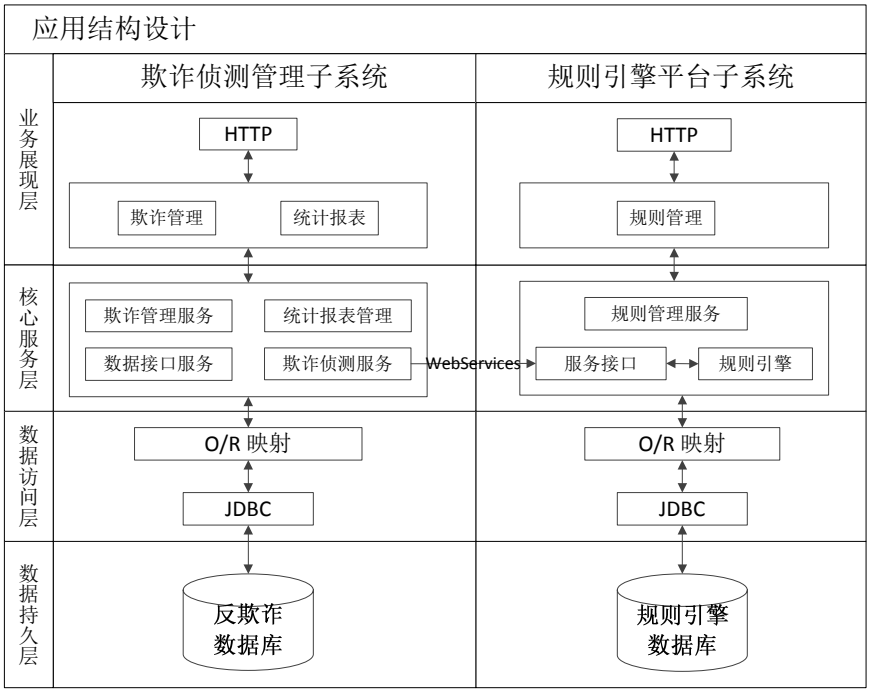

图 4-2 系统应用结构

Fig.4-2 The structure of system application

应用结构描述：

从系统架构及数据处理上分析，可以描述为以下四个层次：

(1) 业务展现层

主要是客户通过HTTP协议访问的Web页面所提供的功能，主要包括规则引擎平台子系统和欺诈侦测管理子系统。

(2) 核心服务层

是系统主要核心组件，通过不同组件模块的组合操作可以完成所有业务功能操作。

(3) 数据访问层

主要是指系统与数据持久层交互实现。

(4) 数据持久层

采用数据库来实现数据的持久化。

应用系统软件如下：

(1) 操作系统

采用成熟稳定的RedHat Linux 9.0及以上作为操作系统；

(2) 数据库管理系统

使用Oracle 10g及以上作为数据库管理系统；

(3) Web 服务

使用目前成熟的B/S构架，Web服务采用Apache Tomcat 6.0及以上，使客户端的使用IE就可以使用系统；

(4) 应用系统平台

使用J2EE构架Sun JDK 1.6及以上作为应用系统平台,支持多操作系统平台；

(5) J2EE 应用框架

系统采用开源项目搭建的J2EE应用框架，使得开发效率大大提高：

- 采用开源 SSH(Struts+Spring+Hibernate) 框架作为系统应用框架；
- 采用开源项目 Drools 来实现规则引擎功能；
- 采用开源项目 CXF 来实现 WebService 服务；

4.4.规则引擎平台子系统后台功能设计

规则引擎平台系统的系统功能设计如图 4-3 所示：

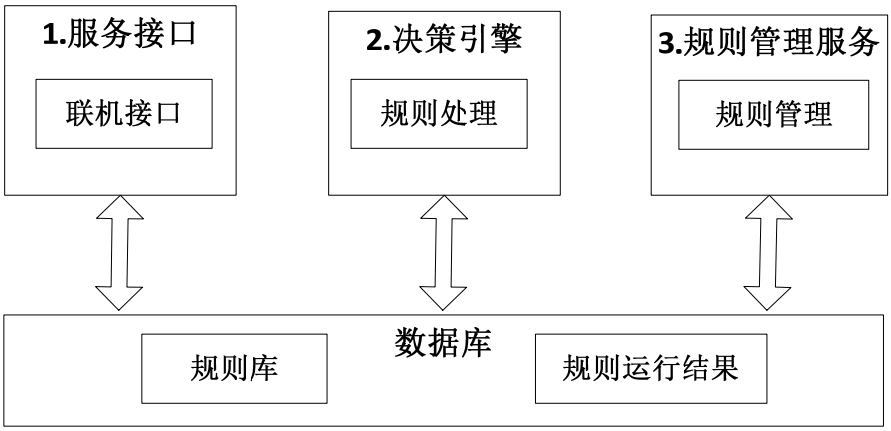

图 4-3 规则引擎平台子系统的系统功能

Fig.4-3 The system function of rule engine platform system

功能图描述：

(1) 服务接口

提供给欺诈侦测管理子系统调用规则引擎服务的接口，主要提供在线的实时服务的联机接口；

(2) 规则引擎

提供规则处理功能，用来执行黑名单检查、模型评分计算和规则计算等规则；

(3) 规则管理

提供规则维护管理功能，主要实现规则集的编辑和规则的编辑、验证、审批、部署、监控和参数维护等功能；

(4) 数据库

- 规则库：由规则管理功能进行维护，供规则引擎使用；
- 规则运行结果：为规则引擎进行规则处理的结果。

4.4.1. 服务接口

服务接口采用 Web Service 方式提供给欺诈侦测管理子系统进行联机访问，是调用规则引擎的联机接口。为了使联机接口具备业务的扩展性，联机接口采用 XML 格式定义的报文进行信息交换。

(1) 功能流程图

服务接口功能流程如图4-4所示：

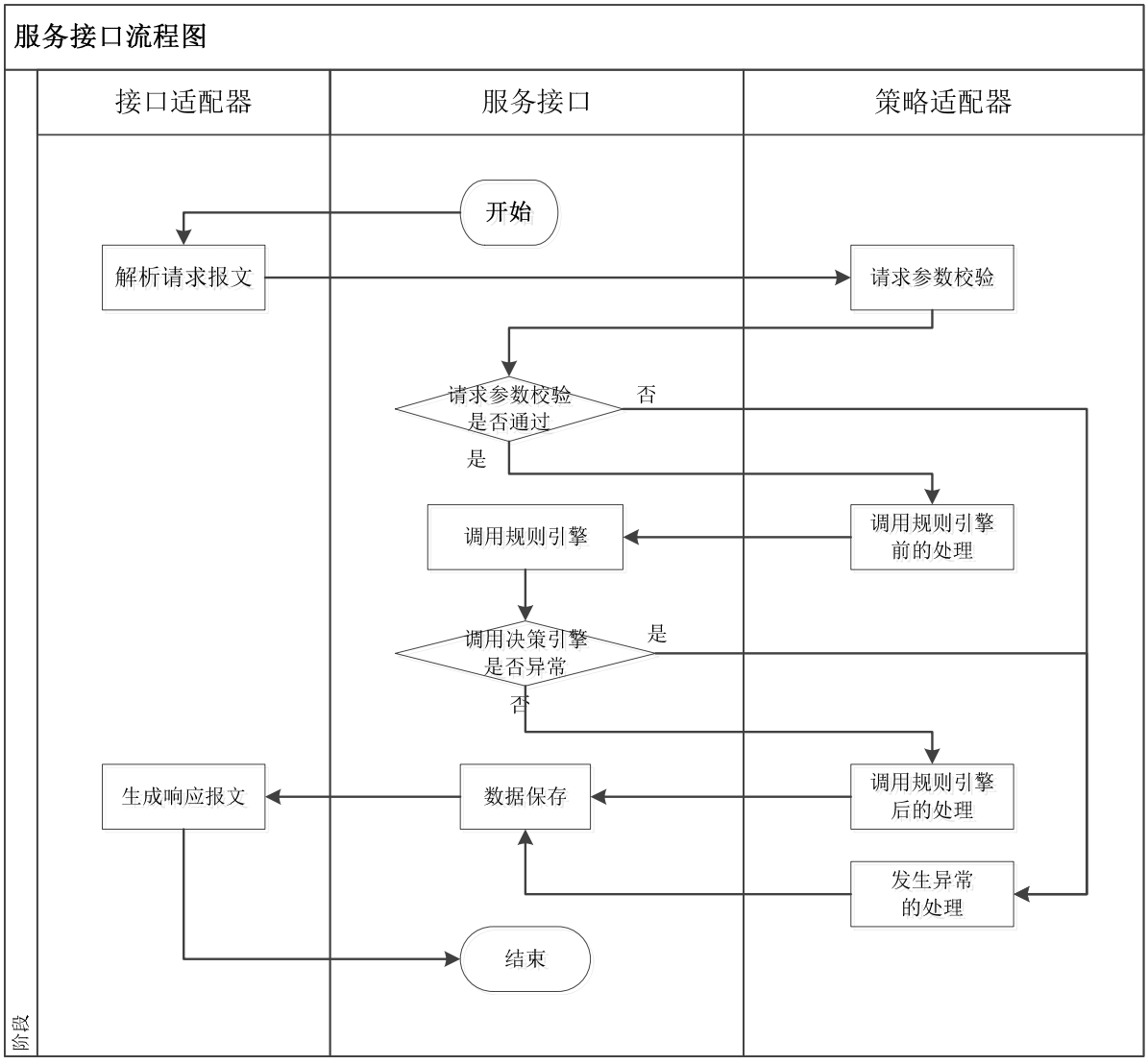

图 4-4 服务接口功能流程图

Fig.4-4 The flow chart of service interface

流程图描述：

- 服务接口

主要功能是实现调用规则引擎的流程逻辑控制，包括：调用接口适配器、调用策略适配器、调用规则引擎、数据保存。其中接口适配器、策略适配器采用适配器模式[19]进行设计；

- 接口适配器

主要功能是解析请求报文；生成响应报文。

- 策略适配器

一个规则集合对应一个策略适配器。主要功能如下：参数校验、调用规则引擎前的处理、调用规则引擎后的处理及发生异常异常的处理。

(2) 类图

服务接口类图如图4-5所示：

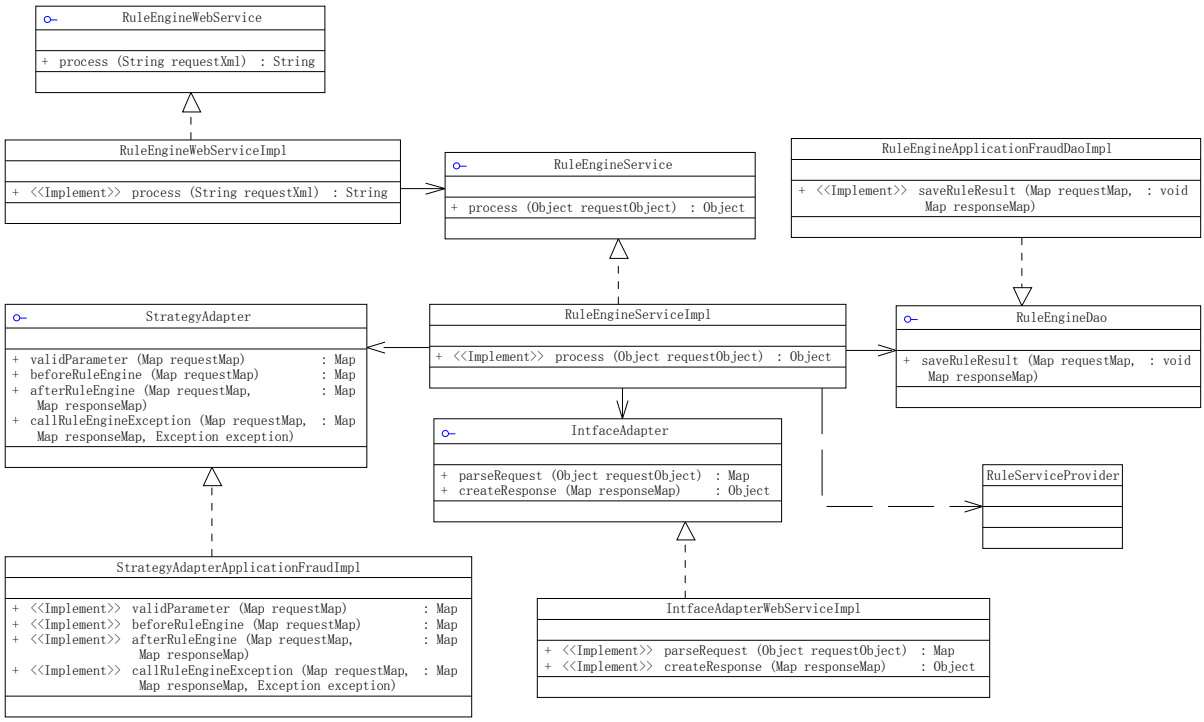

图 4-5 服务接口类图

Fig.4-5 The class diagram of service interface

功能图描述：

- RuleEngineWebService

服务接口的 WebService 接口。是提供给外部系统调用规则引擎的 WebService 服务接口定义。接口包含的方法如下：

1） process：调用规则引擎；

- RuleEngineService

接口服务的定义。主要功能是实现调用规则引擎的流程逻辑控制，协调调用接口适配器、策略适配器、规则处理结果 DAO 和规则引擎运行类、结果保存 DAO 等。其中接口适配器、策略适配器采用适配器模式[19]进行设计。接口包含的方法如下：

1) process：实现调用规则引擎的流程逻辑控制功能；

- IntfaceAdapter

接口适配器。主要功能是对于其他系统进行交互的报文进行解析和组装。接口包含的方法如下：

1) parseRequest：实现请求报文的解析功能；

2) createResponse：实现响应报文的生成功能。

- StrategyAdapter

策略适配器。每个规则对应一个策略适配器。主要实现参数校验、调用规则引擎的相关处理。接口包含的方法如下：

1) validParameter：实现参数校验功能；

2) beforeRuleEngine：实现调用规则引擎前的处理功能；

3) afterRuleEngine：实现调用规则引擎后的处理功能；

4) callRuleEngineException：实现调用规则引擎出现异常的处理功能。

- RuleEngineDao

规则处理结果 DAO。主要功能是将规则处理结果保存到数据库中。接口包含的方法如下：

1) saveRuleResult：实现规则处理结果保存到数据库功能；

- RuleServiceProvider

规则引擎运行类。主要功能是调用规则引擎进行规则的计算。是由规则

引擎提供的运行时 API 提供调用方法。

(3) 系统时序图

服务接口系统时序图如图4-6所示：

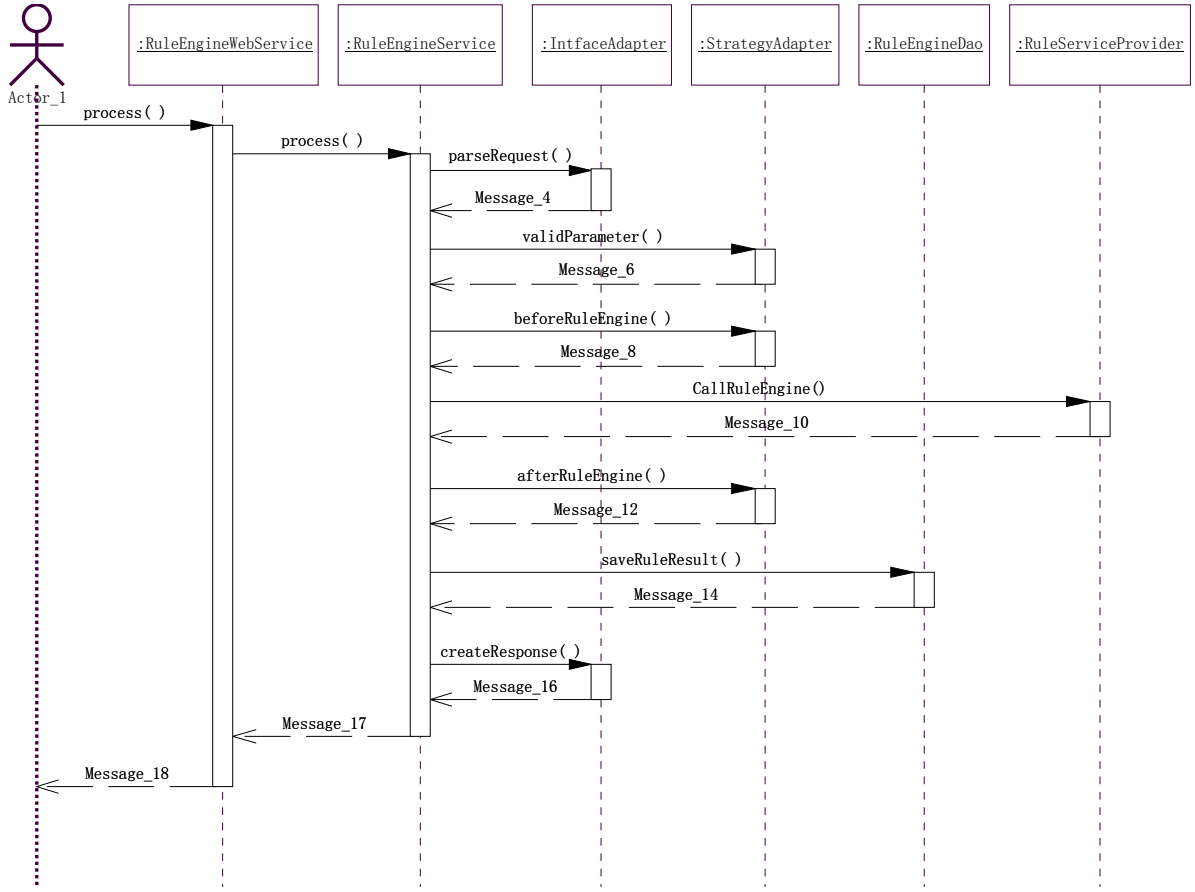

图 4-6 服务接口系统时序图

Fig.4-6 The sequence diagram of service interface

时序图描述：

- 欺诈侦测管理子系统通过 RuleEngineWebService 的 process 方法来调用规则引擎；
- RuleEngineWebService 的 process 方法中调用 RuleEngineService 的 process 方法来进行调用规则引擎的流程处理；
- RuleEngineService 的 process 方法中调用 IntfaceAdapter 的 parseRequest 方法来进行请求报文的解析，并将解析的报文数据返回；
- RuleEngineService 的 process 方法中调用 StrategyAdapter 的 validParameter 方法对解析的请求报文数据进行校验；并将校验的结果

返回；

- RuleEngineService 的 process 方法中调用 StrategyAdapter 的 beforeRuleEngine 方法进行调用规则引擎前的参数处理；
- RuleEngineService 的 process 方法中调用 RuleServiceProvider 进行在规则引擎中进行规则计算；
- RuleEngineService 的 process 方法中调用 StrategyAdapter 的 afterRuleEngine 方法进行调用规则引擎后的参数处理；
- RuleEngineService 的 process 方法中调用 RuleEngineDao 的 saveRuleResult 方法将规则处理结果保存到数据库；
- RuleEngineService 的 process 方法中调用 IntfaceAdapter 的 createResponse 方法来生成响应报文；
- RuleEngineService 的 process 方法将响应报文返回给 RuleEngineWebService 的 process 方法；
- RuleEngineWebService 的 process 方法将响应报文返回给欺诈侦测管理子系统。

(4) 接口定义

服务接口采用XML报文形式与其他系统进行信息交换。XML报文由报文头和报文体两个部分组成。报文头字段如表4-1所示；报文体字段如表4-2所示。

表 4-1 报文头字段定义

Table4-1 The field of message head

参数名	中文描述	类型
SERVICE_TYPE	报文类型	String (1)
SERVICE_CODE	交易码	String(8)
SERVICE_TIME	交易时间	String(20)
CONSUMER_ID	调用系统编号	String (4)
CONSUMER_SEQ	调用系统流水号	String (20)
RETURN_STATUS	交易返回状态	String (1)
RETURN_CODE	交易失败返回代码	String (10)

表 4-2 报文体字段定义

Table4-2 The field of message body

参数名	中文描述	类型
APP_NO	申请件编号	String (20)
RES_CODE	建议结果代码	String (1)
MODEL_GRADE	模型评分	String (4)
RULE_CODE	触发规则代码	String (100)
RULE_RESCODE	触发规则原因码	String (300)

XML 报文类型分为请求报文和响应报文，均由报文头和报文体两部分组成。

1） 请求报文（示例）

```
<?xml version="1.0" encoding="UTF-8"?>
<Service>
    <Head>
        <Field name="SERVICE_TYPE">1</Field>
        <Field name="SERVICE_CODE">0101</Field>
        <Field name="SERVICE_ TIME ">2011-03-28 11:11:11</Field>
        <Field name="CONSUMER_ID">0201</Field>
        <Field name="CONSUMER_SEQ">0201201103280001</Field>
    </Head>
    <Body>
        <Record type="default">
            <Field name="APP_NO">123456789</Field>
            ……
        </Record>
    </Body>
</Service>
```

2） 响应报文（示例）

```
<?xml version="1.0" encoding="UTF-8"?>
<Service>
    <Head>
        <Field name="SERVICE_TYPE">2</Field>
        <Field name="SERVICE_CODE">010101</Field>
        <Field name="SERVICE_TIME">2011-03-28 11:11:11</Field>
        <Field name="CONSUMER_ID">000001</Field>
        <Field name="CONSUMER_SEQ">0201201103280001</Field>
        <Field name="RETURN_STATUS">S</Field>
        <Field name="RETURN_CODE"></Field>
    </Head>
    <Body>
        <Record type="default">
            <Field name="APP_NO">123456789</Field>
            <Field name="RES_CODE">C</Field>
            <Field name="MODEL_GRADE">10</Field>
            <Field name="RULE_CODE">A001</Field>
            <Field name="RULE_RESCODE"></Field>
        </Record>
    </Body>
</Service>
```

4.4.2. 规则引擎

本系统采用 Drools 来实现规则引擎。Drools 是 Jboss 公司旗下一款开源的规则引擎，具有一个易于访问企业策略、易于调整以及易于管理的开源业务规则引擎，符合业内标准，速度快、效率高。它完整的实现了 Rete 算法[5]；通过使用其中的 DSL（Domain Specific Language），可以实现用自然语言方式来描述业务规则，使得业务分析人员也可以看懂业务规则代码。

通过 JSR-94（Java 规则引擎 API）来进行规则引擎的调用，JSR-94 由 javax.rules 包定义，是访问规则引擎的标准企业级 API。分为两个主要部分：

（1）规则管理 API

在javax.rules.admin中定义规则管理API, 主要包括：

- 使用类 RuleServiceProvider 来获得 RuleAdministrator（规则管理）接口的实例。
- 提供方法注册执行集。RuleAdministrator（规则管理器）提供了本地和远程的 RuleExecutionSetProvider。
- RuleExecutionSetProvider 负责创建规则执行集。

(2) 运行时 API

在javax.rules包中定义运行时API，主要包括：

- 类 RuleServiceProvider 提供了对具体规则引擎实现的运行时和管理 API 的访问，通过类 RuleServiceProvider 将其规则引擎实现提供给客户；
- 类 RuleServiceProvider 内部实现了规则管理和运行时访问所需的接口；
- 用 RuleServiceProviderManager 注册所有的 RuleServiceProvider；
- 提供用于创建 RuleSession（规则会话）方法，RuleSession 是用来运行规则的；
- 提供访问注册的 RuleExecutionSets（规则执行集）的方法；
- RuleSession 定义了会话类型, 可以选择有状态会话或者无状态会话；
- RuleExecutionSetMetaData 接口提供查找 RuleExecutionSets 的 metadata（元数据）。
- 元数据通过 RuleSession 接口提供给用户。

4.4.3. 规则管理服务

规则管理服务主要包括规则编辑、规则验证、规则审批、规则部署、规则监控等功能。

(1) 业务流程

规则管理及维护的业务流程如图 4-7 所示：

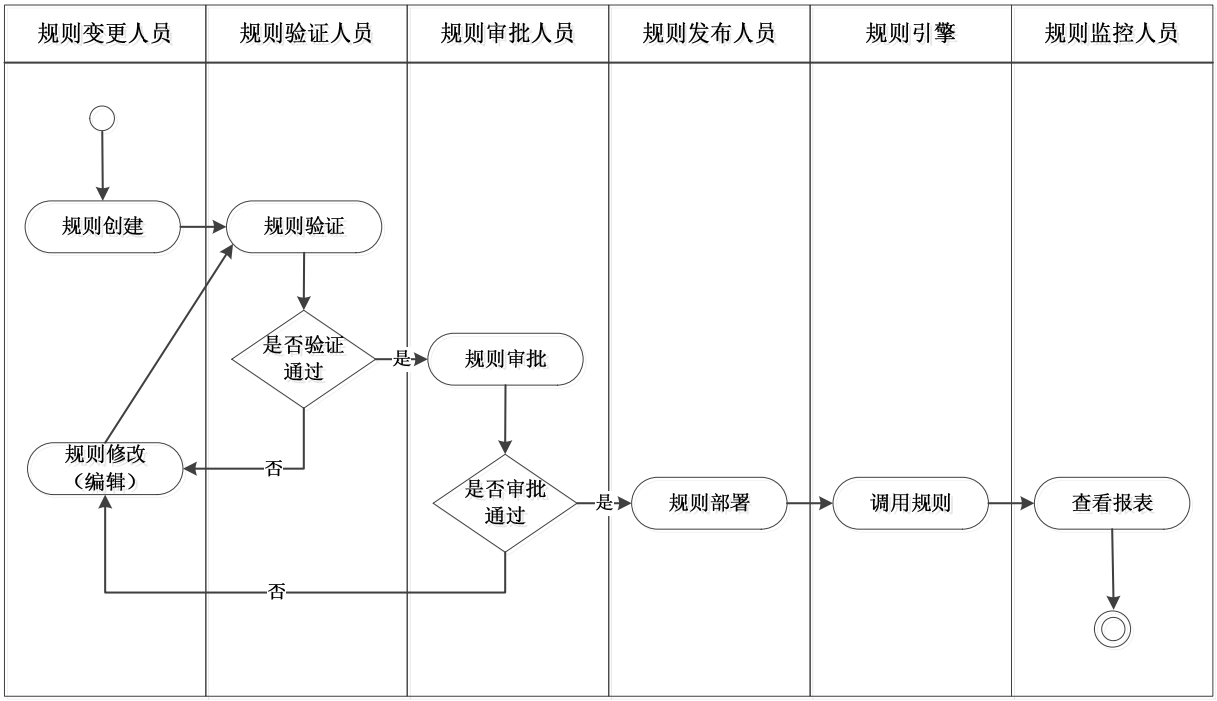

图 4-7 规则管理流程图

Fig.4-7 The flow chart of rule management

流程说明：

- 根据规则开发报告，在系统中创建新的规则；
- 规则验证是对新创建或者修改过的规则进行测试验证；
- 如果验证没通过，则将规则退回进行规则修改；
- 如果验证通过，则由规则审批人员进行规则审批；
- 如果审批没通过，则将规则退回进行规则修改；
- 如果审批通过，则由规则部署人员进行规则部署；
- 发布后的规则由规则引擎调用，执行规则处理；
- 规则监控查是看规则监控跟踪报表，并对规则进行跟踪；
- 依据规则监控跟踪的结果进行规则优化管理；

(2) 功能描述

- 规则版本管理

规则隶属于规则集，每一条规则的可以进行编辑。规则分两大类：

1） 校验规则

对相关数据源中的关联数据项进行规则校验，校验规则可由业务人员进行修改，规则中支持对相关数据项的中文模糊匹配；

2） 模型规则

实现对校验规则的综合结论判定，也可以使用此类型规则实现对模型评分的计算。

规则版本管理即对规则集的版本管理，将规则集存储在存储库中。新建立的规则集可以进行编辑和修改，提交版本后的规则集即成为当前运行中的规则集，当前版本规则可进行编辑和部署。部署的规则集以及历史版本的规则不可编辑。

- 规则编辑

规则变更主要包括规则创建、规则修改和规则优化三个功能：

1） 规则创建

是创建新的规则；

2） 规则修改

是对已有的规则进行维护；

3） 规则优化

是通过分析规则监控跟踪报表判断规则的表现情况，及时发现规则所存在的问题并进行优化。

- 规则验证

编辑完成的规则需要进行规则验证：

1） 可以对单个规则进行验证；

2） 可以对整个规则集进行验证；

3） 如果规则验证通过才可提交进行规则审批；

4） 如果验证不通过就必须进行规则修改后再次规则验证。

- 规则审批

主要是将通过验证的规则进行审批：

1） 如果审批不通过就必须进行规则修改后再次规则验证和审批；

2） 如果规则审批通过规则部署到规则引擎中。

- 规则部署

  主要是将审批通过的规则部署到规则引擎中的规则处理器中。

- 规则监控

  通过分析规则监控跟踪报表来判断规则的表现情况。

4.5.欺诈侦测管理子系统后台功能设计

欺诈侦测管理子系统的系统功能设计如图 4-8 所示：

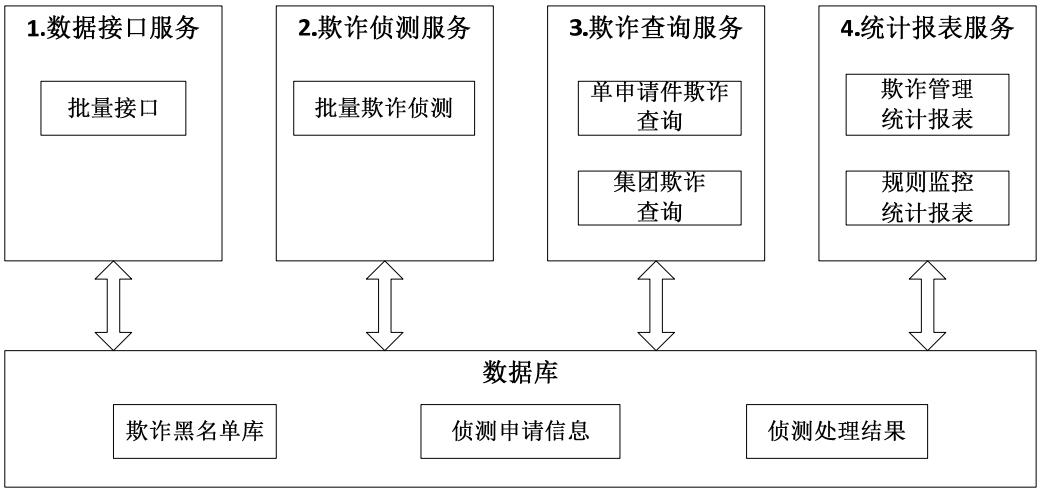

图 4-8 欺诈侦测管理子系统的系统功能

Fig.4-8 The system function of fraud detection management system

功能图描述：

(1) 数据接口服务

负责批量获取各种申请欺诈侦测相关数据，并将这些数据转换为统一的，能够被侦测子系统识别的格式；

(2) 欺诈侦测服务

批量的进行欺诈侦测服务，组织申请欺诈侦测相关数据后调用规则引擎平台子系统进行黑名单检查、模型评分、规则处理等侦测，规则引擎平台子系统给出侦测结果；

(3) 欺诈查询服务

对系统产生的申请欺诈侦测结果进行查询，功能包括申请欺诈查询和集团

欺诈查询；

(4) 统计报表服务

帮助业务和风险管理人员了解欺诈侦测的效果和。同时通过对规则监控分析，可以对现有规则进行评估，对效果不理想的部分进行必要的修正和调整。

4.5.1. 数据接口服务

数据是系统分析的基础，数据采集为系统提供数据支撑。本章阐述系统的数据来源及数据采集的具体内容，并对相关数据进行转换为欺诈侦测系统可识别的格式。

(1) 数据逻辑架构

信用卡申请反欺诈数据逻辑架构设计遵循两大方面原则，一方面要遵循满足目前和未来可能的业务需求原则，另一方面要遵循满足数据仓库系统应用特点的原则。申请反欺诈数据逻辑架构在一个较高层次上解释了数据采集的功能和流程，采用分层、基于原则的方法进行设计。

反欺诈数据逻辑架构如图 4-9 所示：

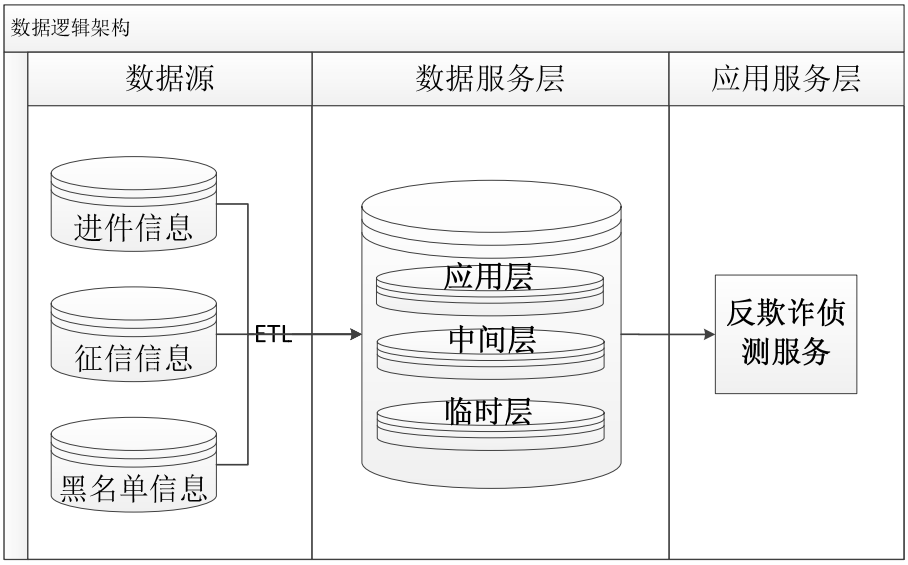

图 4-9 数据逻辑架构图

Fig.4-9 The logic structure diagram of data

架构描述：

- 源数据层

为整个数据分析应用提供原始数据，作为数据整合层的数据抽取源。源系统为申请反欺诈原始数据相关业务系统，包括进件信息，征信信息，黑名

单信息等数据。另外，还可通过数据补录的方式加载分析所用的外部数据。

- 数据服务层

作为申请反欺诈数据核心部分，它负责存储和管理来自各种源数据系统的数据，并为用户访问提供数据服务。反欺诈数据主要包括：

1） 临时层

主要将源系统数据进行临时保存，为后续中间层数据处理做准备。主要进件信息，征信信息，黑名单信息以及其他外部数据；

2） 中间层

主要为系统提供基础数据而设计，主要申请欺诈侦测所需的变量数据。

3） 应用层

主要为欺诈侦测结果数据。

- 应用服务层

主要包括申请反欺诈的应用服务。

(2) 系统数据源

数据来源于进件系统及行外第三方信息。数据主要包括：申请件信息；历史申请审批信息；行内黑名单信息；银联黑名单信息；人行征信信息；固定电话运行商提供的电话相关信息。

- 申请件信息

主要包括：申请编号、姓名、证件类型、证件号码、手机号码、家庭电话、家庭地址、出生日期、学历、工作年限、单位名称、单位电话、单位地址等；

- 历史申请审批信息

主要包括：申请编号、姓名、证件类型、证件号码、手机号码、家庭电话、家庭地址、出生日期、学历、工作年限、单位名称、单位电话、单位地址、审批结果等；

- 黑名单信息

主要包括：身份（姓名+证件号码+证件类型组合）黑名单、电话黑名单、

地址黑名单、单位名称黑名单等。

- 人行征信信息

主要包括：姓名、证件类型、证件号码、手机号码、家庭电话、家庭地址、出生日期，学历，单位名称、单位地址、信贷交易信息、公共信息等；

- 电话相关信息

主要包括：电话号码、安装地址、安装时间等；

(3) 系统返回结果信息

系统返回的结果信息主要包括：申请编号、建议结果、欺诈模型评分、触犯规则和原因码。

(4) ETL 设计

在数据进入系统并能够提供欺诈侦测管理子系统使用之前，需要使用 ETL 技术[20],[21]对这些数据进行处理工作：数据源确定，主要是数据源数据结构和软硬件平台的确定；数据映射，主要是源数据与本系统数据库中对应关系；数据获取，主要是源数据获取方式的确定；数据转换，主要是将异构的数据进行转换；数据加载，主要是将源数据加载到本系统的数据库中，实现与质量检核系统的相关功能的相互调用，实现整个 ETL 流程的自动化，以及实现 ETL 过程的异常处理机制。

- ETL 逻辑架构

ETL 逻辑架构设计如图 4-10 所示

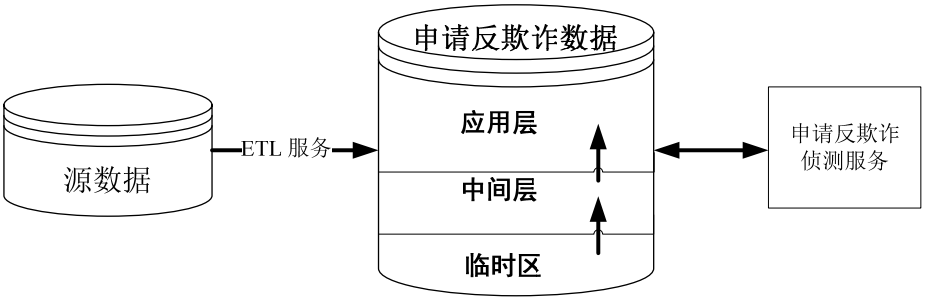

图 4-10 ETL 逻辑架构图

Fig.4-10 The logic structure diagram of ETL

- ETL 流程调度

ETL 流程调度是指管理员可以通过 ETL 工具的任务调度功能对 ETL 任

务进行运行间隔设置，自动在规定条件满足时启动 ETL 任务，数据的加工处理过程体现自动化原则，减少人工处理过程；源系统和反欺诈数据集市之间的数据流转应该自动化、无缝衔接的。在 ETL 任务调度层加入程序性控制，以控制任务在自动调度时，必须满足符合的条件，否则不予执行。ETL 流程自动调度执行时，可有以下触发条件：

1） 时间触发－指定 Job 在特定的时间点开始运行。

2） 事件触发－发生特定的事件后 Job 自动运行。

3） 时间和事件的结合－多个条件组合都满足后自动运行 Job。

- ETL 任务监控

数据集市系统的维护中最重要的就是随时监控 ETL 任务的运行情况，为了降低系统维护人员的劳动强度，提供友好的 ETL 监控功能是必要的。在功能上，ETL 监控应提供如下功能：

1） 全程全方位监控

采用专业的 ETL 运行及管理工具，可以用来监控系统的运行状况，完成以下管理及运行功能：查看 ETL Job 运行结果，如完成，异常等；控制 ETL Job 的运行，如启动，停止，重置 Job 等；查看 ETL Job 运行详细日志，导出详细日志到文本。通过 Job 监控可以实时监控 ETL Job 的运行状态

2） 多种手段

任务运行失败－ETL 工具可以自动取消整个控制进程，也可以转入异常处理流程；ETL 工具允许从断点处继续执行 Job，而不必将所有的 Job 重新运行；任务运行结束时间延迟－Job 运行时间超长，可以释放 Job；如果 Job 被锁定，需要先释放相关资源重置 Job。

3） 自动报错

邮件通知－Job 运行情况可以通过 Email 通知管理员，主要用于每日指定的时间检查当日的 Job 运行状况并通过 Email 的形式将检查结果发送给运行维护人员

- ETL 功能划分

ETL 功能模块的目标是具体实现从源到目标的的 ETL 任务程序，包括抽取源数据、数据加载、数据质量检查、数据转换及数据质量检查 2。在具体实现上，应遵循以下原则：

1） ETL 过程数据尽量少落地；

2） 制定清晰明确且不冗余的数据接口，使 ETL 过程尽量少传重复数据；

3） 模型设计要分层、数据库设计要分区；

4） 编写 SQL 脚本要考虑性能；设计开发要具备规范性

以下对 ETL 各个功能进行说明：

1） 抽取源数据

先从源系统上根据约定的采集周期采集全量或增量数据，生成相应的接口数据表。在采集过程中可能涉及系统内或跨系统的数据关联获取。这些接口数据表的结构与源数据基本相同。

2） 数据加载

实现将数据接口表加载入目标数据库临时区的功能。此阶段可以细分为如下三个阶段：数据加载前：实现完整性检查、Drop 索引的功能；数据加载：实现将数据加载入数据库里的功能；数据加载后：实行重建索引的功能

加载数据时，根据需要，可以采用 ETL 工具实现，也可以通过编写脚本调用数据库工具来实现。数据的追加策略根据数据的抽取策略以及业务规则确定，一般有以下三种类型：直接追加、全部覆盖、更新追加。

3） 数据转换

实现将临时区数据按照数据映射转换为目标表数据结构的功能。具体逻辑为连接一个或多个临时区表，去除不通过数据质量检核的记录，根据数据映射逻辑，经过数据复制、数据翻译、数据聚合、复杂计算以及数据匹配等处理过程，整合到目标数据表中。具体包含以下过程：符缺省值、代理键的生成、字段合并与拆分、数据排序、数据翻译、数据聚合。

由于此阶段是实现主要逻辑运算的阶段，功能一般比较复杂，为保证性能与可维护性，可以进一步拆分为多个程序进行处理，每一个程序输出的结果作为下一步程序读入的数据。

进行数据转换时：如果是数据抽取过程中，必须考虑对数据抽取过程和业务系统的性能影响；如果是以文件方式进行异步加载数据处理时，要考虑磁盘的存储量和整个 ETL 流程的协调性工作，还有大量的非 SQL 语句编程；如果是在数据加载过程时，必须考虑数据加载性能影响；

如果是先数据装载完成后再处理时，必须考虑数据库系统的海量数据处理性能。

4） 数据质量检查

数据质量检查，检查源表关键字段内容是否为 NULL。对于那些不能通过数据质量检查的数据，我们将其存入相应的拒绝数据表中，以便作为下一步骤的输入和将来数据检查人员跟踪检查。

进行相关业务规则的数据质量检核。反欺诈数据集市数据质量问题主要来源于以下几种情况：源数据质量问题：源系统中的数据信息不真实、不符合业务规则或数据约束条件，或者源系统导出的接口数据文件不符合接口标准或格式等；数据从源系统到数据集市的抽取、传输过程中造成数据失真、丢失，或在整合过程中对数据的取舍存在误判；这类问题主要来自于 ETL 体系本身，可以通过各类技术手段进行避免。数据质量检查包括记录级检查以及业务指标检查。

记录级检查主要包括：数据类型、数据格式、主外键及关联、编码规范、数据值域。

业务指标检查主要包括：比较同一业务指标在临时区与中间层数值，判断在数据转换时是否遗漏或异常；利用业务指标的参考值与该数值相比较，判断数据是否正确；

数据质量检查过程中，如果发生数据转换的异常现象，则需要相关业务人员进行相应的数据修复处理，主要是数据源文件重新生成过程；如果是数据源文件本身有误，则数据源进行修改。本系统不变更任何数据。

- 数据接口方案

1） 控制文件

控制文件可以是不含任何内容的空文件，它的作用在于：一旦出现该文件，表示特定的某个或某些数据文件传输完毕且成功，否则，即使这些数据文件已经出现，ETL 只认为这些文件处于传输的过程中，并没有传输完毕。

2） 数据暂存区

数据暂存区是存放各个系统临时数据的区域，其形式可以是数据文件也可以是数据库表。

由于数据暂存区的空间是有限的，长时间的运行，会造成该空间内存放大量过时的接口数据，因此，必须定期清除对该空间内的过时接口数据。各个系统自身的数据暂存区由各个系统自己负责，定期完成对过时接口数据的清除工作。

4.5.2. 欺诈侦测服务

欺诈侦测服务主要是对信用卡申请欺诈侦测的处理流程进行控制，欺诈侦测服务功能流程如图 4-11 所示：

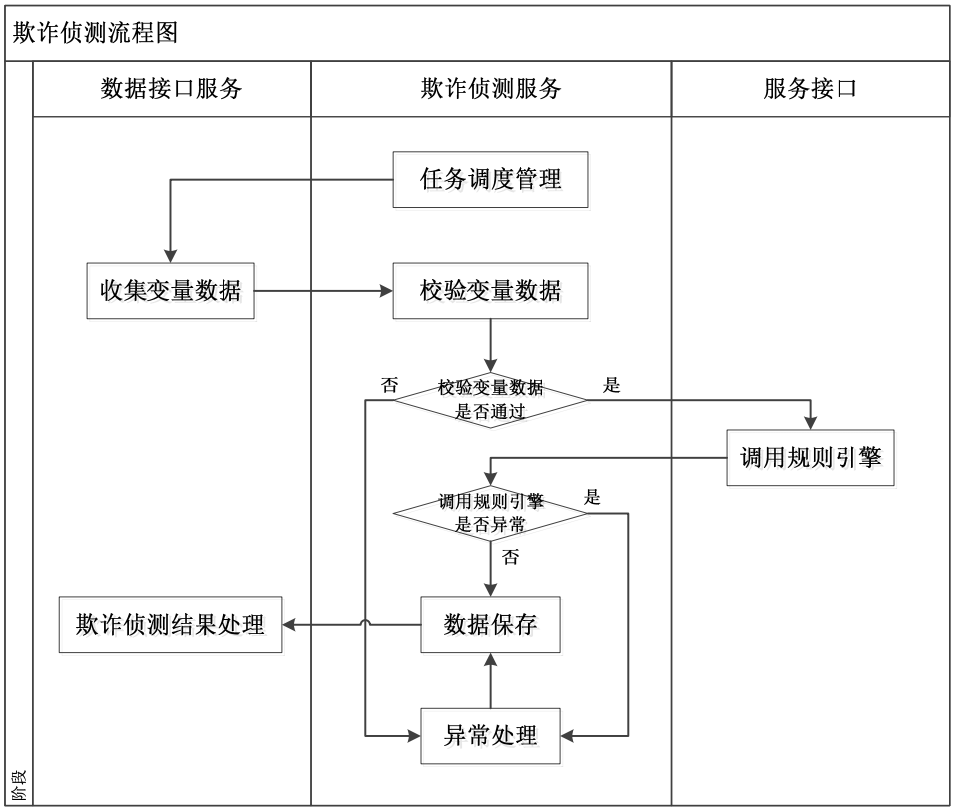

图 4-11 欺诈侦测服务功能流程图

Fig.4-11 The flow chart of fraud detection service

功能流程说明：

(1) 任务调度管理

欺诈侦测服务是后台服务，是通过任务调度器定时的进行调用欺诈侦测服务的批量欺诈侦测，调用周期是每日零点调用一次；

(2) 收集变量数据

欺诈侦测服务被调用时，先调用数据接口服务的批量接口进行收集变量数

据。将前一日的申请信息以及欺诈侦测相关的数据进行数据抽取、数据加载、数据转换及数据质量检查等数据处理，最终加工出欺诈侦测所需要的变量数据；

(3) 校验变量数据

欺诈侦测服务对收集的变量数据进行校验，校验数据是否符合调用规则引擎的要求，如果符合要求则调用规则引擎，如果不符合要求则进行异常处理；

(4) 调用规则引擎

当变量数据校验通过时，使用变量数据通过规则引擎平台子系统的服务接口来调用规则引擎，在规则引擎进行相应的黑名单检查、评分模型计算、规则处理等欺诈侦测处理，规则引擎会将欺诈侦测结果返回给欺诈侦测服务。

(5) 异常处理

当欺诈侦测服务处理过程是出现异常时，进行异常信息的生成或转换为欺诈侦测处理的结果。异常主要包括：变量数据校验不通过、调用规则引擎异常；

(6) 数据保存

将欺诈侦测结果或异常处理的结果信息保存到数据库中，并将这些数据返回给数据接口服务；

(7) 欺诈侦测结果处理

数据接口服务接收欺诈侦测服务返回的欺诈侦测结果，在由数据接口服务将欺诈侦测结果返回给信用卡进件系统。

4.5.3. 欺诈查询服务

欺诈查询服务主要是对欺诈侦测结果进行查询，主要功能描述如下：

(1) 查询数据源

欺诈侦测服务所产生的欺诈侦测结果。

(2) 查询功能

- 申请欺诈查询

  主要是针对申请欺诈侦测结果的组合条件查询。能够查询到申请的欺诈侦测建议结果、模型评分等等。

- 集团欺诈查询

  主要是针对集团欺诈侦测结果的组合条件查询。能够查询到集团欺诈侦

测结果和集团欺诈关联的历史申请信息。

(3) 查询结果

- 申请欺诈查询

查询结果主要包括：申请件编号、姓名、证件号码、建议结果、模型评分、触发规则、原因码等。

- 集团欺诈查询

集团欺诈侦测结果主要包括：申请件编号、姓名、证件号码、单位名称、地址、电话等。

集团欺诈关联的历史申请信息主要包括：申请件编号、姓名、证件号码、单位名称、地址、电话、建议结果、审批结果、日期等。

4.5.4. 统计报表服务

统计报表服务主要是对给用户所展现报表所需的数据进行数据预加工处理的后台服务。统计报表服务功能流程如图 4-12 所示：

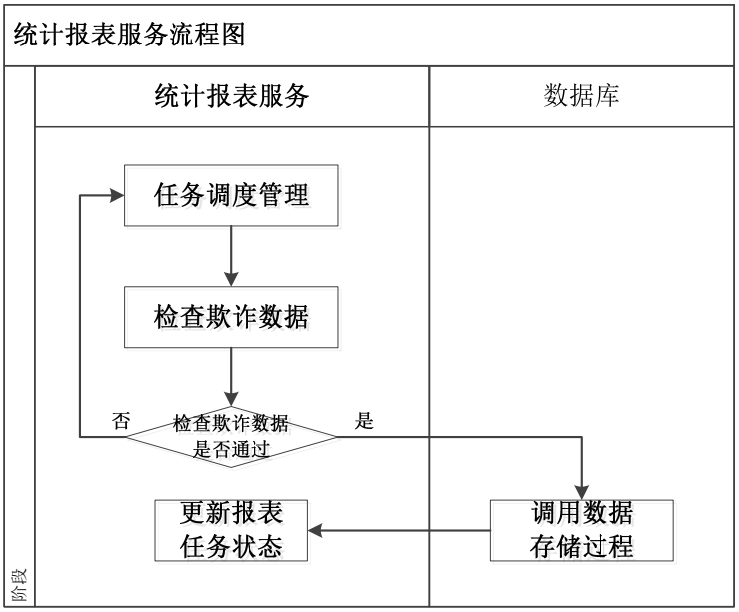

图 4-12 欺诈侦测服务功能流程图

Fig.4-12 The flow chart of report service

功能流程说明：

(1) 任务调度管理

统计报表服务是后台服务，是通过任务调度器定时的进行调用统计报表服

务进行报表数据的预加工处理，调用周期是每日凌晨四零点至八点时间段进行调用；

(2) 检查欺诈数据

主要是检查当日的欺诈侦测服务调度任务是否运行，欺诈侦测结果数据是否已经产生。如果没有通过检查欺诈数据，则通知任务调度管理一小时后再次调用；如果通过检查欺诈数据，则调用数据存储过程进行报表数据预加工处理；

(3) 调用数据存储过程

使用数据库的数据存储过程来进行报表数据预加工处理，在数据库空闲时间充分利用数据库资源进行报表数据的预加工处理，这样可以使报表数据加工效率最好。数据存储过程主要包括规则监控统计报表数据加工和欺诈管理统计报表数据加工；

(4) 更新报表任务状态

当统计报表数据加工调度调度任务完成时，对调度任务的状态信息进行保存，状态信息主要包括：开始时间、结束时间、任务状态等等。

4.6.系统非功能设计

4.6.1. 系统安全设计

(1) 用户安全

- 密码修改策略

用户首次登录系统时，系统强制进行操作员密码修改；用户在前一次修改密码 30 天后再次登录系统，系统强制进行操作员密码修改。该密码过期期限可参数化控制，以“天”为设置单位；修改后的新密码不能与前一次使用过的密码重复；密码输错 6 次用户即被锁定，即使操作员使用正确的密码也无法再登录，需要管理员重置后用户才能够登录。

- 密码复杂度策略

密码长度控制为 6-12 位;密码复杂度控制为至少 1 位数字及字母,字母不区分大小写，操作员密码重置时不受此限制。

- 登陆超时策略

系统实现参数化控制登录超时功能，以“分钟”为设置单位。操作员登

录系统后，在参数设置的时间内未进行任何操作，系统在下一个交易进行操作时，自动签退系统，恢复至登录页面，并提示“登录超时，请重新登录系统”。自动签退时，系统正在处理但未提交确认的交易，操作不成功。

(2) 应用安全

对系统采用用户认证，二级用户角色管理、访问权限控制、系统监控、系统备份等手段来对系统的运行提供支持。

- 系统访问

系统访问用户可以通过 IE 浏览器进行系统访问，在进入系统时，需要通过用户名和密码的方式来进行用户的身份认证，同时系统可以提供 IP、MAC 地址绑定的选项，限制访问系统的用户的所用机器 IP；

- 角色管理控制：

系统提供用户角色层的控制，实现用户的角色的设定和管理。通过用户所属角色来获取该角色绑定的系统访问资源（系统功能）。用户和角色建立多对多关联关系；

- 访问权限控制

建立一组与业务相关的固定式的角色列表，系统管理员或部门内的系统管理员可以维护 角色和所管辖权限的关联关系。从而达到 当用户拥有某个角色岗位，就可以访问这个岗位所拥有资源（功能）。

(3) 数据安全

对敏感数据，有些不能从生产环境卸下来，比如密码，有些数据在传输和存储过程中要加密，同时为了防止数据丢失必须对数据按一定周期进行安全备份。

4.6.2. 系统监控设计

系统拥有完备的运行日志、用户操作监控日志、用户登录日志以及系统异常日志等。

(1) 日志类型

应用系统日志；用户操作日志；数据库日志；发布日志；安全日志；错误日志。

(2) 日志记录信息

用户 ID；日期，时间（至少精确到秒）；终端身份和位置（IP 或 MAC 地

址）；操作及操作对象:（登入，登出，创建，删除，修改，提交，审批，授权，增删用户，修改密码）；主要标识（如客户号码，机构名称）；行动记录（事前及事后数据）；系统错误信息。

(3) 日志信息保护

防篡改 禁止非授权用户访问日志；禁止编辑或删除日志；任何对日志文件的访问尝试都必须被记录。无特殊声明日志至少保存一年。必须有容量监控。

(4) 日志收集

保证日志记录功能在任何时候都能正常运行。在系统出现无法恢复的异常前，记录预警信息。

4.6.3. 可扩充性设计

(1) 应用服务可扩充性设计

主要将通过软件、硬件的扩充，以及增加应用服务控制器，来调度、平衡系统服务；

为应对系统后续规模扩容和应用开发升级，可采用的“集群技术”和“分散布署”的方式来应对：

- 集群计算

  可对 WEB 应用服务器进行集群布署

- 分散布署

  可将两个字系统分散布署到不同的机器上。

(2) 数据库可扩充性设计

采用方法如下：

- 数据库分区存储设计

  以利于后续多节点数据库服务器的并行访问

- 系统级的“集群技术”

  增加数据库服务器节点,可以让不同的数据库服务器同时访问不同的分区数据；

4.6.4. 可靠性设计

硬件架构中采用双机冗余架构，应用在任何一台设备上运行都能满足业务的需要，软硬件配置在发生切换情况下需满足现实运行的要求（包括高峰期）；应用系统的主机、应用、网卡方面具备高可用性。

可由 WEB 应用服务器和工具服务器做双机冗余备份，可安装同样的应用软件、工具软件及中间件，可以在两台服务器上任一台运行另一台的应用

5. 信用卡申请反欺诈系统实现与验证

5.1.关键功能实现

关键功能实现是规则引擎平台子系统和欺诈侦测管理子系统的前台Web功能的实现，关键功能主要包括规则引擎平台系统的规则管理、欺诈侦测管理子系统的欺诈查询和统计报表三个功能。

5.1.1. 规则管理

规则管理功能的系统实现主要如下：

(1) 类实现

规则管理的前台关键功能类图设计如图5-1所示：

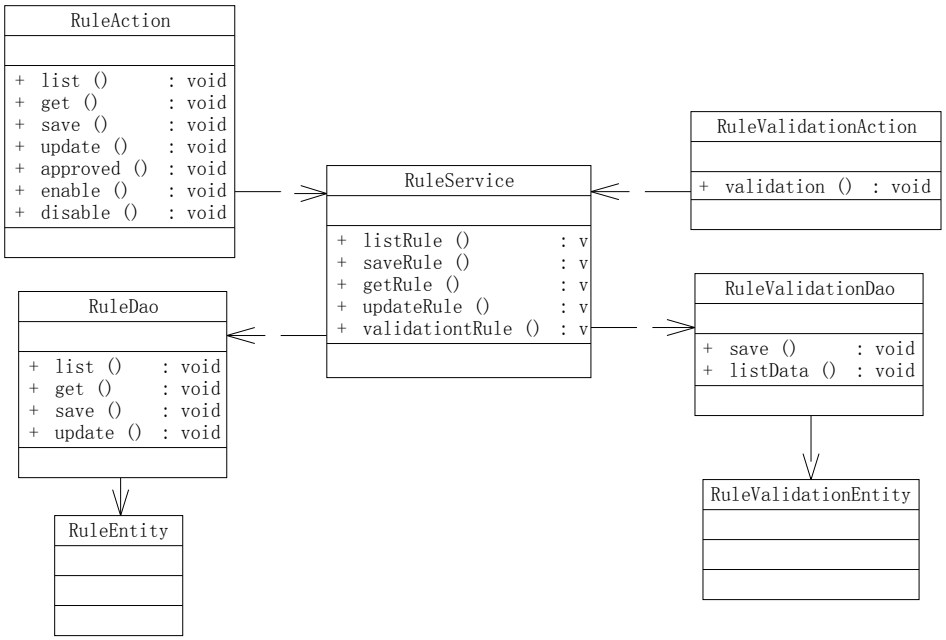

图 5-1 规则管理类图

Fig.5-1 The class diagram of rule management

类图描述：

- RuleAction

实现规则管理的页面交互，主要包括规则的查询、创建、修改、保存、更新、审批、启用、停用等功能。

- RuleValidationAction

实现规则验证管理的页面交互，主要包括规则验证查询和测试等功能；

- RuleService

规则管理服务的实现类，主要实现规则编辑、验证、审批、部署等功能；

- RuleDao

实现规则管理功能的对数据库操作，支持规则数据的查询、增加、修改等操作；

- RuleValidationDao

实现规则验证管理功能的对数据库操作，支持规则验证数据的查询、增加、修改等操作；

- RuleEntity

规则数据的实体类；

- RuleValidationEntity

规则验证数据的实体类。

(2) 数据库实现

规则管理功能实现的主要数据表如图5-2所示：

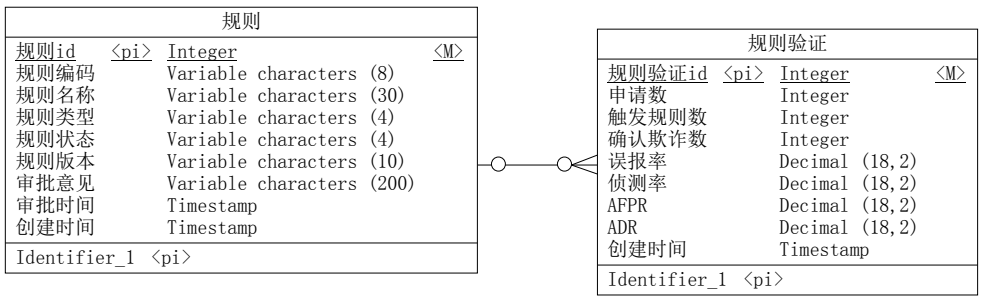

图 5-2 规则管理表

Fig.5-2 The table of rule management

数据表结构描述：

- 规则(rule)表的字段如表 5-1 所示：

表 5-1 规则集表字段

Table5-1 The field of rule table

字段中文名称	字段英文名称	字段类型
规则 id	rule_id	INTEGER
规则集 id	rule_set_id	INTEGER
规则编码	rule_code	VARCHAR2(8)
规则名称	rule_name	VARCHAR2(30)
规则类型	rule_type	VARCHAR2(4)
规则状态	rule_state	VARCHAR2(4)
规则版本	rule_version	VARCHAR2(10)
审批意见	approved_content	VARCHAR2(200)
审批时间	approved_time	TIMESTAMP
创建时间	update_time	TIMESTAMP

- 规则验证(rule_validation)表的字段如表 5-2 所示：

表 5-2 规则集表字段

Table5-2 The field of rule validation table

字段中文名称	字段英文名称	字段类型
规则验证 id	rule_validation_id	INTEGER
规则 id	rule_id	INTEGER
申请数	vali_count	INTEGER
触发规则数	vali_rule_count	INTEGER
确认欺诈数	vali_fraud_count	INTEGER
误报率	vali_error_rate	NUMBER(18,2)
侦测率	vali_detection_rate	NUMBER(18,2)
AFPR	vali_afpr	NUMBER(18,2)
ADR	vali_adr	NUMBER(18,2)
创建时间	update_time	TIMESTAMP

(3) 功能实现

规则管理功能的菜单如图5-3所示：

图 5-3 规则管理页面

Fig.5-3 The page of rule management

- 规则编辑

点击“规则管理”菜单中的“规则编辑”子菜单，进入规则查询页面。输入查询条件，点击“查询”按钮时，调用RuleAction类的list方法查询规则，规则查询的时序图如图5-4所示 ：

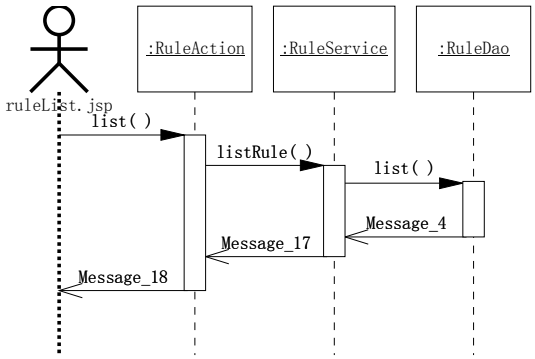

图 5-4 规则查询时序图

Fig.5-4 The sequence diagram of rule query

时序图描述：

1） 规则查询页面（ruleList.jsp）通过RuleAction的list方法查询规则；

2） RuleAction的list方法中调用RuleService的listRule方法查询规则；

3） RuleService的listRule方法中调用RuleDao的list方法查询规则；

4） RuleDao的list方法查询数据库的规则表中规则，并将规则查询结果返回到RuleService的listRule方法中；

5） RuleService的listRule方法中将规则查询结果返回到RuleAction的list方法中；

6） RuleAction的list方法中将查询结果在规则查询页面显示。显示查询结果如图5-5所示：

图 5-5 规则查询结果页面

Fig.5-5 The page of rule query result

在规则查询页面中点击“新增”按钮，进入规则创建（ruleAdd.jsp）页面。规则创建页面如图5-6所示：

图 5-6 规则新建页面

Fig.5-6 The page of create rule

在规则创建页面输入相关规则信息；点击“提交”按钮时，调用RuleAction类的save方法保存规则，规则保存的时序图如图5-7所示 ：

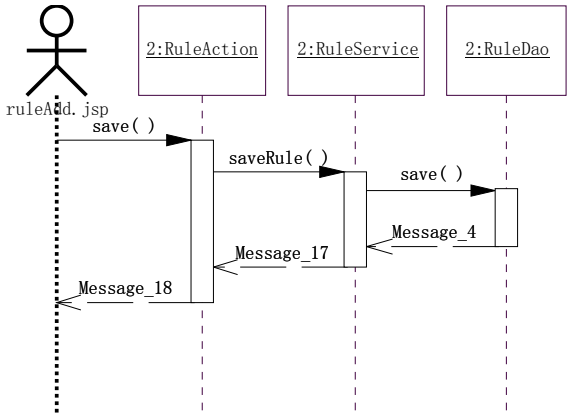

图 5-7 规则保存时序图

Fig.5-7 The sequence diagram of rule save

时序图描述：

1) 规则创建页面通过RuleAction的save方法保存规则；

2) RuleAction的save方法中调用RuleService的saveRule方法保存规则；

3) RuleService的saveRule方法中调用RuleDao的save方法保存规则；

4) RuleDao的save方法将规则保存到数据库的规则表中；

在规则查询页面中选择需要编辑的规则，点击“编辑”按钮，调用RuleAction类的get方法查询单个规则信息。时序图如图5-8所示 ：

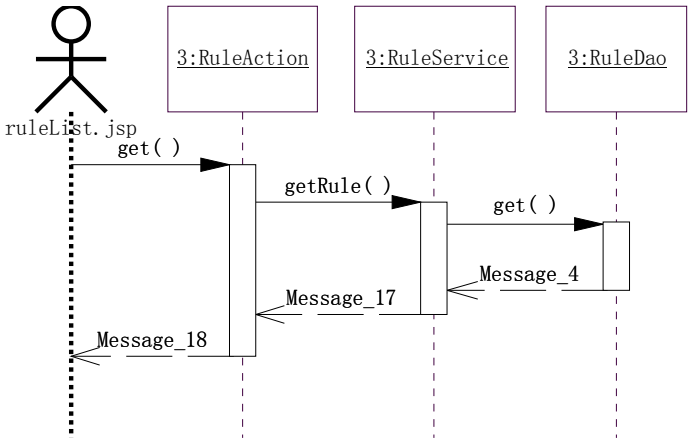

图 5-8 规则保存时序图

Fig.5-8 The sequence diagram of rule get

时序图描述：

1） 规则查询页面通过RuleAction的get方法查询规则；

2） RuleAction的get方法中调用RuleService的getRule方法查询规则；

3） RuleService的getRule方法中调用RuleDao的get方法查询规则；

4） RuleDao的get方法查询数据库的规则表中的规则，并将规则查询结果返回到RuleService的getRule方法中；

5） RuleService的getRule方法将结果返回到RuleAction的get方法中；

6） RuleAction的get方法将结果显示在规则编辑（ruleEdit.jsp）页面。规则编辑页面如图5-9所示：

图 5-9 规则编辑页面

Fig.5-9 The page of edit rule

在规则编辑页面修改相关规则信息；点击“提交”按钮时，调用RuleAction类的update方法更新规则，规则更新的时序图如图5-10所示 ：

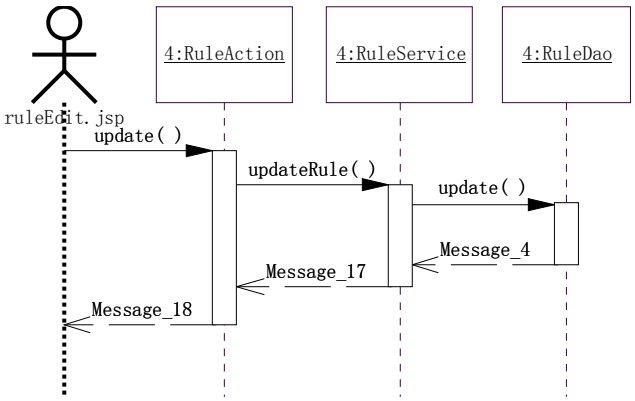

图 5-10 规则保存时序图

Fig.5-10 The sequence diagram of rule update

时序图描述：

1) 规则创建页面通过RuleAction的update方法更新规则；

2) RuleAction的update方法中调用RuleService的updateRule方法更新；

3) RuleService的updateRule方法中调用RuleDao的update方法更新；

4) RuleDao的update方法将规则更新到数据库的规则表中；

在规则查询页面中选择需要编辑规则定义的规则，点击“规则定义”按钮时，调用规则定义的编辑页面进行规则的具体定义编辑。

- 规则验证

点击“规则管理”菜单中的“规则验证”子菜单，进入规则验证页面（ruleValidationList.jsp）。进入规则验证页面时调用RuleAction类的list方法查询规则，时序图参考如图5-5所示。规则验证页面如图5-11所示：

图 5-11 规则验证页面

Fig.5-11 The page of rule validation

在规则验证页面。选中所需验证的规则，输入测试时间周期，点击“测试”按钮时，调用RuleValidationAction类的validation方法验证规则，时序图如图5-12所示 ：

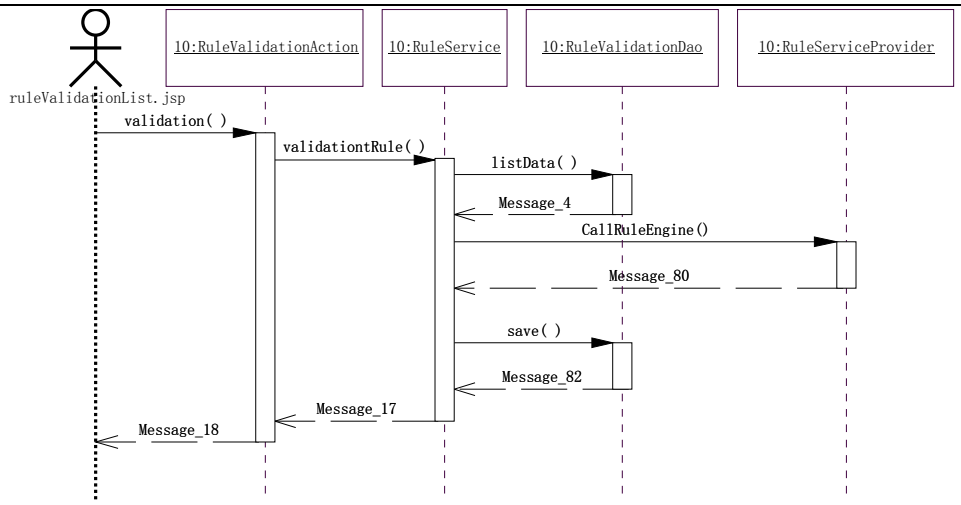

图 5-12 规则验证时序图

Fig.5-12 The sequence diagram of rule validation

时序图描述：

1） 规则验证页面通过RuleValidationAction的validation方法验证规则；

2） RuleValidationAction的validation方法中调用RuleService的validationRule方法验证规则；

3） RuleService的validationRule方法中调用RuleValidationDao的listdata方法根据测试时间周期在数据库中查询测试的样本数据；

4） RuleService的validationRule方法中调用RuleServiceProvider进行在规则引擎中对选中的规则进行计算；

5） RuleService的validationRule方法中调用RuleValidationDao的save方法将规则验证结果保存到数据库中；

6） RuleService的validationRule方法将规则通过样本数据验证结果返回到RuleValidationAction的validation方法中；

7） RuleValidationAction的validation方法将规则验证结果在规则验证结果页面显示。页面如图5-13所示：

图 5-13 规则验证结果页面

Fig.5-13 The page of rule validation result

- 规则审批

点击“规则管理”菜单中的“规则审批”子菜单，进入规则审批页面（ruleApprovedList.jsp）。进入规则审批页面时调用RuleAction类的list方法查询规则，时序图参考如图5-5所示。规则审批页面如图5-14所示：

图 5-14 规则审批页面

Fig.5-14 The page of rule approved

在规则审批页面。输入审批信息，点击“完成审批”按钮时，调用RuleAction类的approved方法完成审批规则，时序图如图5-15所示 ：

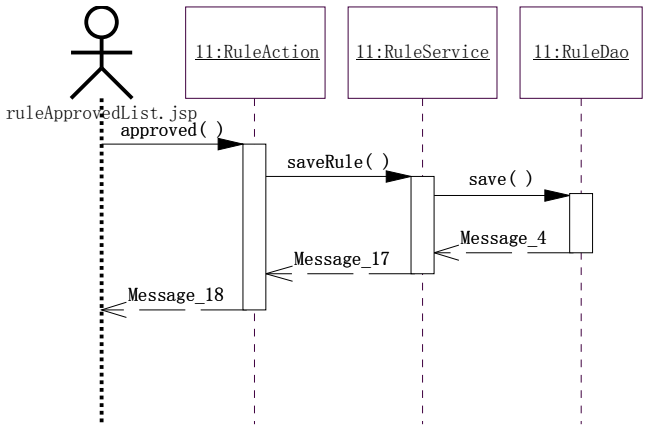

图 5-15 规则审批时序图

Fig.5-15 The sequence diagram of rule approved

时序图描述：

1) 规则审批页面通过RuleAction的approved方法更新规则审批信息；
2) RuleAction的approved方法中调用RuleService的updateRule方法更新规则审批信息；
3) RuleService的updateRule方法中调用RuleDao的update方法更新规则审批信息；
4) RuleDao的update方法将规则审批信息更新到数据库的规则表中；

- 规则部署

点击“规则管理”菜单中的“规则部署”子菜单，进入规则部署页面（ruleDeploymentList.jsp）。进入规则审批页面时调用RuleAction类的list方法查询规则，时序图参考如图5-5所示。规则部署页面如图5-16所示：

图 5-16 规则部署页面

Fig.5-16 The page of rule deployment

在规则部署页面。选择需启用的规则，点击“启用”按钮时，调用RuleAction类的enable方法完成启用规则，时序图如图5-17所示 ：

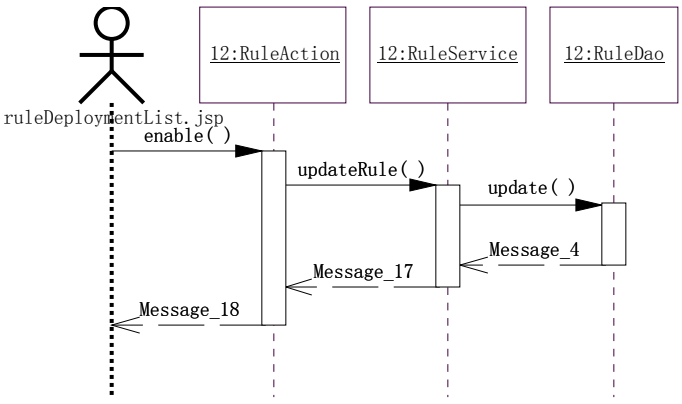

图 5-17 规则启用时序图

Fig.5-17 The sequence diagram of rule enable

时序图描述：

1) 规则部署页面通过RuleAction的enable方法更新规则部署信息；

2) RuleAction的enable方法中调用RuleService的updateRule方法更新规则部署信息；

3) RuleService的updateRule方法中调用RuleDao的update方法更新规则部署信息；

4) RuleDao的update方法将规则部署信息更新到数据库的规则表中；

在规则部署页面。选择需停用的规则，点击“停用”按钮时，调用RuleAction类的disable方法完成停用规则，时序图如图5-18所示 ：

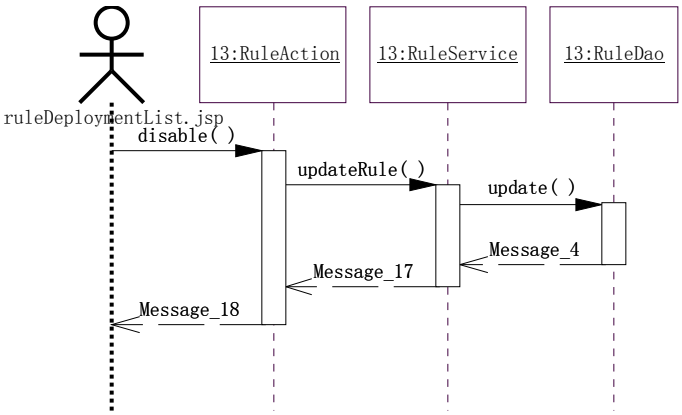

图 5-18 规则启用时序图

Fig.5-18 The sequence diagram of rule enable

时序图描述：

1) 规则部署页面通过RuleAction的disable方法更新规则部署信息；

2) RuleAction的enable方法中调用RuleService的updateRule方法更新规则部署信息；

3) RuleService的updateRule方法中调用RuleDao的update方法更新规则部署信息；

4) RuleDao的update方法将规则部署信息更新到数据库的规则表中；

5.1.2. 欺诈查询

欺诈查询功能的系统实现主要如下：

(1) 类实现

欺诈查询的前台关键功能类图设计如图5-19所示：

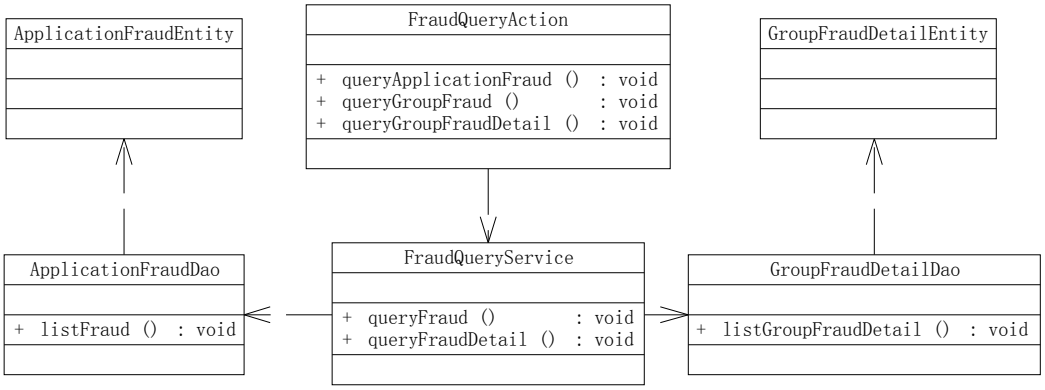

图 5-19 欺诈查询类图

Fig.5-19 The class diagram of fraud query

类图描述：

- FraudQueryAction 实现欺诈查询的页面交互；
- FraudQueryService 欺诈查询服务的实现类，主要实现规则但申请件和集团欺诈查询服务功能；
- ApplicationFraudDao 实现申请件欺诈查询的对数据库操作；
- GroupFraudDetailDao 实现集团欺诈详细数据查询的对数据库操作；
- ApplicationFraudEntity 申请欺诈侦测结果的实体类；
- GroupFraudDetailEntity 集团欺诈侦测结果详细信息的实体类；

(2) 数据库实现

欺诈侦测管理子系统的主要数据表如图5-20所示：

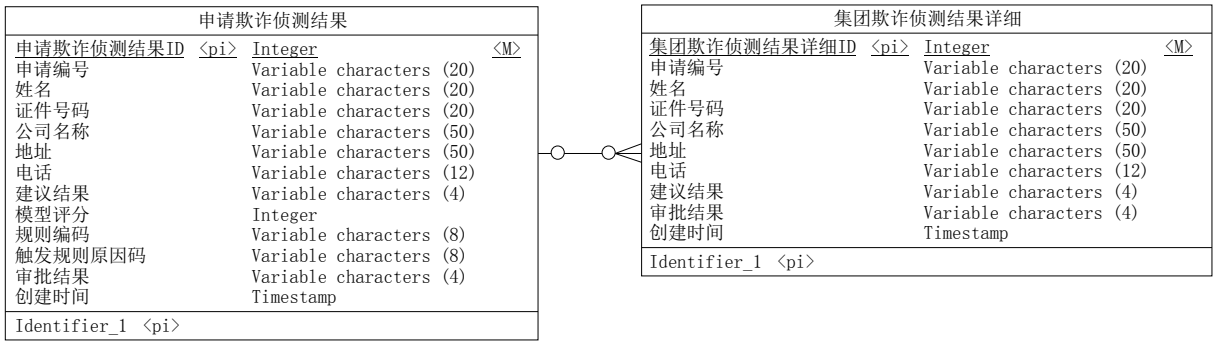

图 5-20 申请查询管理表

Fig.5-20 The table of fraud query management system

数据表结构描述：

- 申请欺诈侦测结果(app_fraud_result)表的字段如表 5-3 所示：

表 5-3 申请欺诈侦测结果表字段

Table5-3 The field of application fraud result table

字段中文名称	字段英文名称	字段类型
申请欺诈侦测结果 ID	app_fraud_result_id	INTEGER
申请编号	app_no	VARCHAR2(20)
姓名	name	VARCHAR2(20)
证件号码	cert_no	VARCHAR2(20)
公司名称	group_name	VARCHAR2(50)
地址	group_addr	VARCHAR2(50)
电话	group_tel	VARCHAR2(12)
建议结果	detection_result	VARCHAR2(4)
模型评分	model_score	INTEGER
规则编码	rule_code	VARCHAR2(8)
触发规则原因码	rule_res_code	VARCHAR2(8)
审批结果	approved_result	VARCHAR2(4)
创建时间	update_time	TIMESTAMP

- 集团欺诈侦测结果详细(group_fraud_result_detail)表的字段如表 5-4 所示：

表 5-4 集团欺诈侦测结果详细表字段

Table5-4 The field of group fraud result detail table

字段中文名称	字段英文名称	字段类型
集团欺诈侦测结果详细 ID	group_fraud_result_detail_id	INTEGER
申请欺诈侦测结果 ID	app_fraud_result_id	INTEGER
申请编号	app_no	VARCHAR2(20)
姓名	name	VARCHAR2(20)
证件类型	cert_type	VARCHAR2(4)
证件号码	cert_no	VARCHAR2(20)
公司名称	group_name	VARCHAR2(50)
地址	group_addr	VARCHAR2(50)
电话	group_tel	VARCHAR2(12)
建议结果	detection_result	VARCHAR2(4)
审批结果	approved_result	VARCHAR2(4)
创建时间	update_time	TIMESTAMP

(3) 功能实现

欺诈查询功能主要包括：申请欺诈查询和集团欺诈查询。欺诈查询功能的菜单如图 5-21 所示：

图 5-21 欺诈查询页面

Fig.5-21 The page of fraud query

- 申请欺诈查询

点击“欺诈查询”菜单中“申请欺诈查询”子菜单，进入申请欺诈查询页面。输入查询条件，点击“查询”按钮时，调用FraudQueryAction类的queryApplicationFraud方法查询申请欺诈侦测结果，时序图如图5-22所示 ：

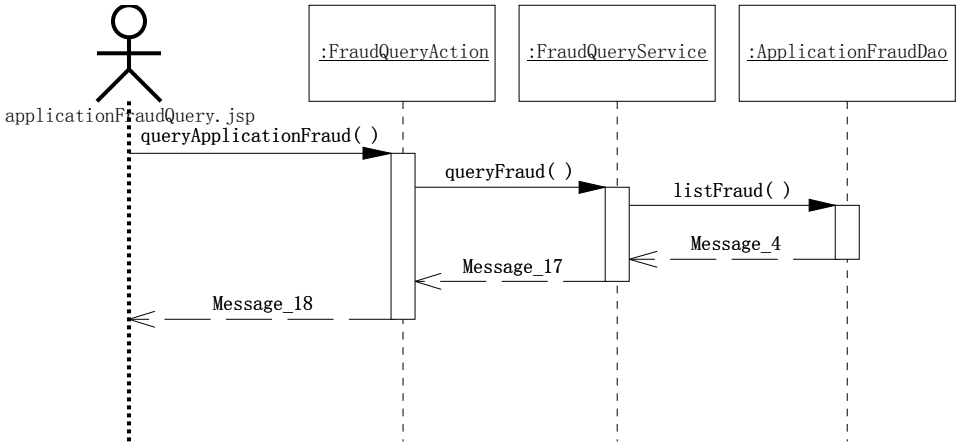

图 5-22 申请欺诈查询时序图

Fig.5-22 The sequence diagram of application fraud query

时序图描述：

1） 申请欺诈查询页面（applicationFraudQuery.jsp）通过调用FraudQueryAction的queryApplicationFraud方法查询侦测结果；

2） FraudQueryAction的queryApplicationFraud方法中依据查询条件调用FraudQueryService的queryFraud方法查询申请欺诈侦测结果；

3） FraudQueryService的queryFraud方法中调用ApplicationFraudDao的listFraud方法查询申请欺诈侦测结果；

4） ApplicationFraudDao的listFraud方法查询数据库的申请欺诈侦测结果表中的数据，将查询的欺诈侦测结果返回到FraudQueryService的queryFraud方法中；

5） FraudQueryService的queryFraud方法中将查询的欺诈侦测结果返回到FraudQueryAction的queryApplicationFraud方法中；

6） FraudQueryAction的queryApplicationFraud方法中将查询结果在申请欺诈查询页面显示。显示查询结果页面如图5-23所示：

图 5-23 申请欺诈查询页面

Fig.5-23 The page of application fraud query

- 集团欺诈查询

点击“欺诈查询”菜单中“集团欺诈查询”子菜单，进入集团欺诈查询页面。输入查询条件，点击“查询”按钮时，调用FraudQueryAction类的queryGroupFraud方法查询集团欺诈侦测结果，时序图如图5-24所示 ：

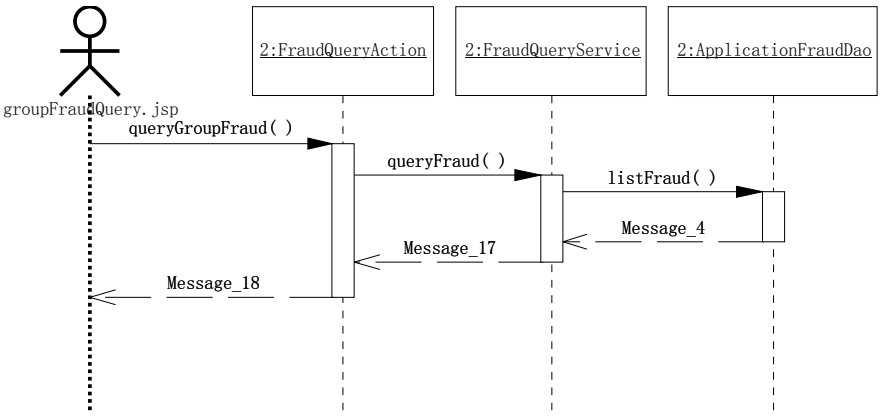

图 5-24 集团欺诈查询时序图

Fig.5-24 The sequence diagram of group fraud query

时序图描述：

1） 集团欺诈查询通过FraudQueryAction的queryGroupFraud方法查询集团欺诈侦测结果；

2） FraudQueryAction的queryGroupFraud方法中依据查询条件调用FraudQueryService的queryFraud方法查询集团欺诈侦测结果；

3） FraudQueryService的queryFraud方法中调用ApplicationFraudDao的listFraud方法查询集团欺诈侦测结果；

4） ApplicationFraudDao的listFraud方法在数据库的申请欺诈侦测结果表中查询触发集团欺诈规则的申请欺诈侦测结果，将查询的集团欺诈侦测结果返回到FraudQueryService的queryFraud方法中；

5） FraudQueryService的queryFraud方法中将查询的集团欺诈侦测结果返回到FraudQueryAction的queryGroupFraud方法中；

6） FraudQueryAction的queryGroupFraud方法中将查询结果在集团欺诈查询页面（groupFraudQuery.jsp）显示。显示查询结果页面如图5-25所示：

图 5-25 集团欺诈查询页面

Fig.5-25 The page of group fraud query

在集团欺诈查询页面。选中查询结果，点击“查看”按钮时，调用FraudQueryAction类的queryGroupFraudDetail方法查询集团欺诈侦测结果详

细信息，时序图如图5-26所示 ：

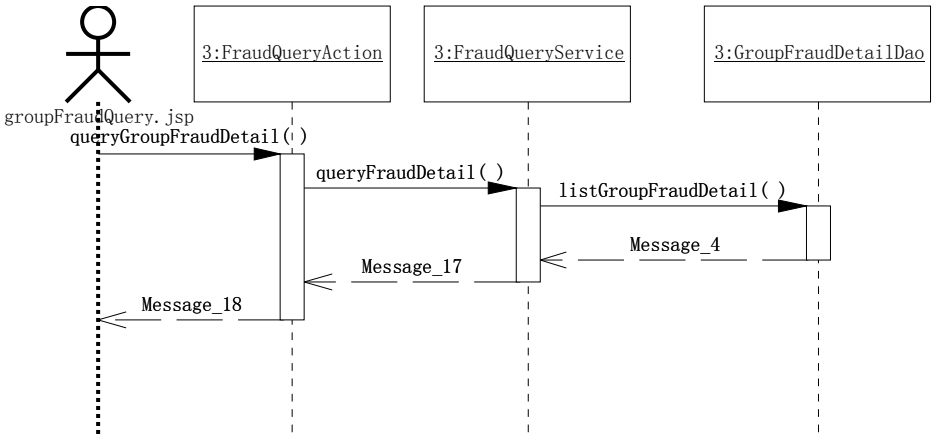

图 5-26 集团欺诈详细查询时序图

Fig.5-26 The sequence diagram of group fraud detail query

时序图描述：

1） 集团欺诈查询通过FraudQueryAction的queryGroupFraudDetail方法查询集团欺诈详细数据；

2） FraudQueryAction的queryGroupFraudDetail方法中调用FraudQueryService的queryFraudDetail方法查询详细数据；

3） FraudQueryService的queryFraudDetail方法中调用GroupFraudDetailDao的listGroupFraudDetail方法查询详细数据；

4） GroupFraudDetailDao的listGroupFraudDetail方法在数据库的集团欺诈侦测结果详细表中查询，将查询结果返回到FraudQueryService的queryFraudDetail方法中；

5） FraudQueryService的queryFraudDetail方法中将查询结果返回到FraudQueryAction的queryGroupFraudDetail方法中；

6） FraudQueryAction的queryGroupFraudDetail方法中将查询结果在集团欺诈查询详细页面（groupFraudDetailQuery.jsp）显示。显示查询结果页面如图5-27所示：

图 5-27 集团欺诈查询详细页面

Fig.5-27 The page of group fraud detail query

5.1.3. 统计报表

统计报表包括规则监控统计报表和欺诈管理统计报表。系统实现主要如下：

(1) 类实现

统计报表的前台关键功能类图设计如图5-28所示：

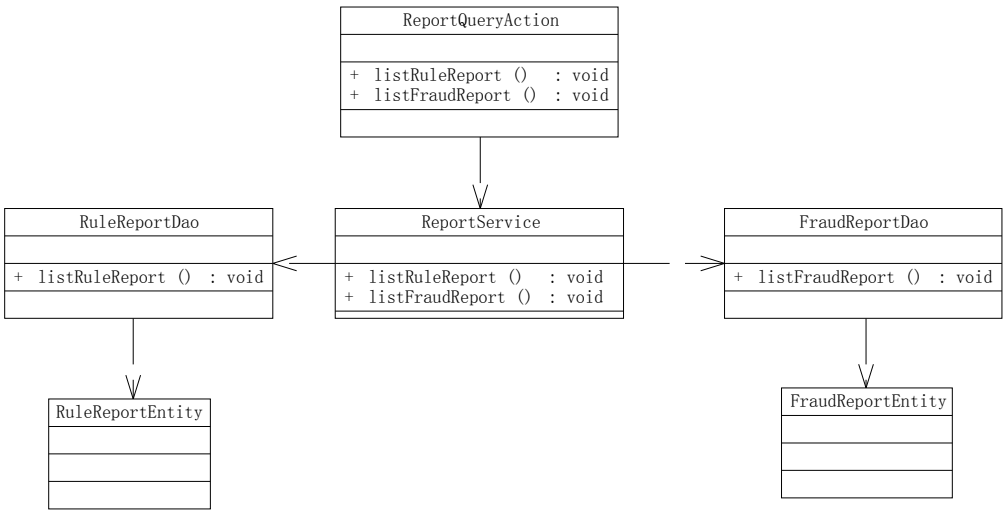

图 5-28 统计报表类图

Fig.5-28 The class diagram of report query

类图描述：

- ReportQueryAction

实现统计报表的查询页面交互，主要包括规则监控统计报表和欺诈管理统计报表查询；

- ReportService

统计报表服务的实现类，主要实现规则监控和欺诈管理统计报表查询；

- RuleReportDao

实现规则监控统计报表查询功能的对数据库查询操作；

- FraudReportDao

实现欺诈管理统计报表查询功能的对数据库查询操作；

- RuleReportEntity

规则监控统计报表数据的实体类；

- FraudReportEntity

欺诈管理统计报表数据的实体类；

(2) 数据库实现

统计报表管理的主要数据表如图5-29所示：

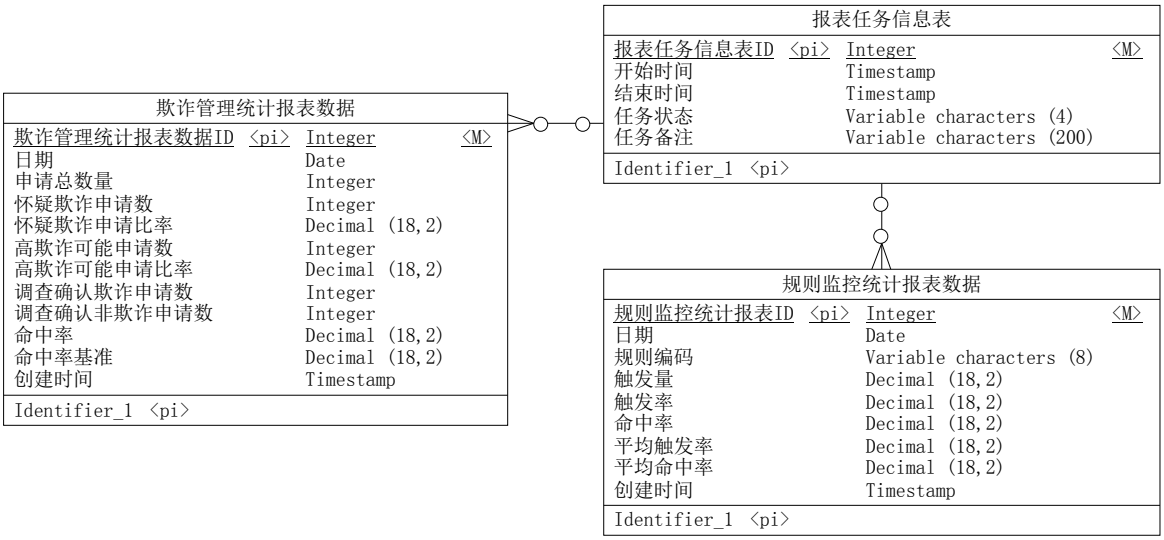

图 5-29 统计报表管理表

Fig.5-29 The table of report query

数据表结构描述：

- 报表任务信息(report_task_info)表的字段如表 5-5 所示：

表 5-5 报表任务信息表字段

Table5-5 The field of report task info table

字段中文名称	字段英文名称	字段类型
报表任务信息表 ID	report_task_info_id	INTEGER
日期	report_date	DATE
开始时间	start_time	TIMESTAMP
结束时间	end_time	TIMESTAMP
任务状态	task_state	VARCHAR2(4)
任务备注	task_remark	VARCHAR2(200)

- 规则监控统计报表数据(rule_report_info)表的字段如表 5-6 所示：

表 5-6 规则监控统计报表数据表字段

Table5-6 The field of rule report table

字段中文名称	字段英文名称	字段类型
规则监控统计报表 ID	rule_report_id	INTEGER
报表任务信息表 ID	report_task_info_id	INTEGER
规则编码	rule_code	VARCHAR2(8)
触发量	trigger_count	NUMBER(18,2)
触发率	trigger_rate	NUMBER(18,2)
命中率	hit_rate	NUMBER(18,2)
平均触发率	avg_trigger_rate	NUMBER(18,2)
平均命中率	avg_hit_rate	NUMBER(18,2)
创建时间	update_time	TIMESTAMP

- 欺诈管理统计报表数据(fraud_report_info)表的字段如表 5-7 所示：

表 5-7 欺诈管理统计报表数据表字段

Table5-7 The field of fraud report table

字段中文名称	字段英文名称	字段类型
欺诈管理统计报表数据 ID	fraud_report_id	INTEGER
报表任务信息表 id	report_task_info_id	INTEGER
申请总数量	app_count	INTEGER
怀疑欺诈申请数	susp_fraud_count	INTEGER
怀疑欺诈申请比率	susp_fraud_rate	NUMBER(18,2)
高欺诈可能申请数	hight_fraud_count	INTEGER
高欺诈可能申请比率	hight_fraud_rate	NUMBER(18,2)
调查确认欺诈申请数	is_fraud_count	INTEGER
调查确认非欺诈申请数	no_fraud_count	INTEGER
命中率	hit_rate	NUMBER(18,2)
命中率基准	hit_rate_base_line	NUMBER(18,2)
创建时间	update_time	TIMESTAMP

(3) 功能实现

统计报表功能的菜单如图 5-30 所示：

图 5-30 统计报表页面

Fig.5-30 The page of report query

- 规则监控统计报表

点击“统计报表”菜单中“规则监控统计报表”子菜单，进入规则监控统计报表查询页面。点击“查询”按钮调用ReportQueryAction类的listRuleReport方法查询规则监控统计报表，时序图如图5-31所示 ：

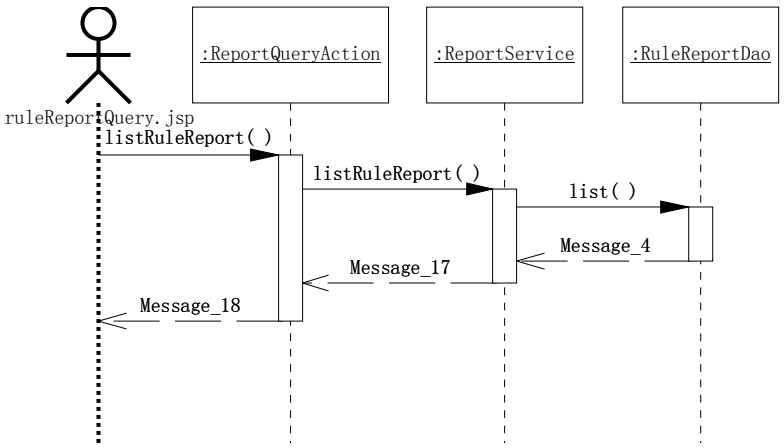

图 5-31 规则监控统计报表查询时序图

Fig.5-31 The sequence diagram of rule report query

时序图描述：

1） 规则监控统计报表查询页面（ruleReportQuery.jsp）通过调用ReportQueryAction的listRuleReport方法查询规则监控统计报表；

2） ReportQueryAction的listRuleReport方法中调用ReportService的listRuleReport方法查询规则监控统计报表；

3） ReportService的listRuleReport方法中调用RuleReportDao的list方法查询规则监控统计报表；

4） RuleReportDao的list方法查询数据库的规则监控统计报表数据表中的数据，将查询结果返回到ReportService的listRuleReport方法中；

5） ReportService的listRuleReport方法中将查询的统计报表结果返回到ReportQueryAction的listRuleReport方法中；

6） ReportQueryAction的listRuleReport方法中将查询结果在规则监控统计报表查询页面显示。查询结果页面如图5-32所示：

图 5-32 规则监控统计报表查询页面

Fig.5-32 The page of rule report query

- 欺诈管理统计报表

点击“统计报表”菜单中“欺诈管理统计报表”子菜单，进入欺诈管理统计报表查询页面。点击“查询”按钮调用ReportQueryAction类的listFraudReport方法查询欺诈管理统计报表，时序图如图5-33所示 ：

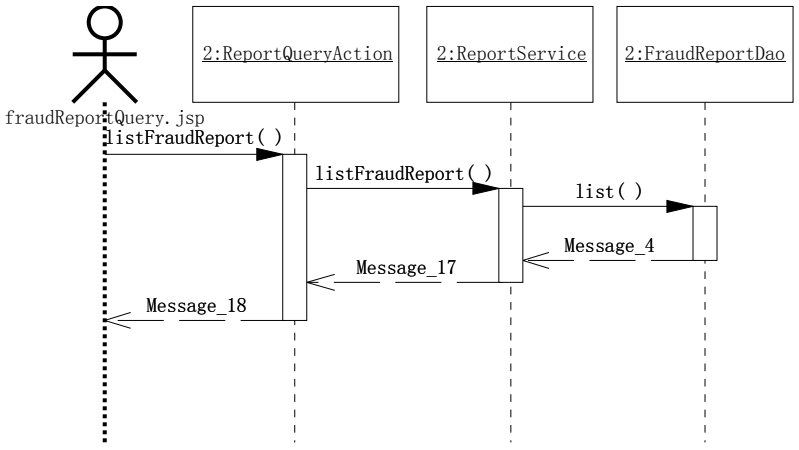

图 5-33 欺诈管理统计报表查询时序图

Fig.5-33 The sequence diagram of fraud report query

时序图描述：

1） 欺诈管理统计报表查询页面（fraudReportQuery.jsp）通过调用ReportQueryAction的listFraudReport方法查询欺诈管理统计报表；

2） ReportQueryAction的listFraudReport方法中调用ReportService的listFraudReport方法查询欺诈管理统计报表；

3） ReportService的listFraudReport方法中调用FraudReportDao的list方法查询欺诈管理统计报表；

4） RuleReportDao的list方法查询数据库的欺诈管理统计报表数据表中数据，将查询结果返回到ReportService的listFraudReport方法中；

5） ReportService的listFraudReport方法中将查询的统计报表结果返回到ReportQueryAction的listFraudReport方法中；

6） ReportQueryAction的listFraudReport方法中将查询结果在欺诈管理统计报表查询页面显示。查询结果页面如图5-34所示：

图 5-34 欺诈管理统计报表查询页面

Fig.5-34 The page of fraud report query

5.2.关键功能验证

关键功能验证是设计测试用例数据来验证信用卡申请反欺诈系统的对欺诈识别的有效性；功能验证流程如下：

(1) 设计三条测试用例：第一条测试用例的预期的建议结果为“无欺诈”；第二条测试用例的预期的建议结果为“怀疑欺诈”；第二条测试用例的预期的建议结果为“高欺诈可能”；

(2) 使用测试用例数据依次调用欺诈侦测管理子系统的数据接口服务和欺诈侦测服务，其中欺诈侦测服务会实时的调用规则引擎平台系统的服务接口；

(3) 欺诈侦测服务调用完成后，将欺诈侦测结果与测试案例的预期结果进行对比来验证欺诈识别的有效性。

5.2.1. 验证用例

由于信用卡申请欺诈侦测的需要外部数据源，所以必须对外部数据数据文件进行测试用例设计来验证欺诈侦测的评分模型和规则的有效性。

信用卡申请信息的测试用例数据文件格式：申请编号，姓名，证件号码，家庭电话，家庭地址，单位名称，单位电话，单位地址。数据文件内容如图5-35所示：

```
[root@localhost 20130715]# pwd
/app/ApplicationFraud/data/20130715
[root@localhost 20130715]# cat application_info.txt
10200701,张三,11111111111111111,021-11111111,高科路11弄11号101,金融发展公司,021-66666666,龙东路1号
10200702,李四,222222222222222222,021-22222222,高科路22弄22号202,浦东发展公司,021-77777777,徐东路1号
10200703,王五,333333333333333333,021-33333333,高科路33弄33号303,徐汇发展公司,021-88888888,高东路1号
[root@localhost 20130715]#
```

图 5-35 申请件数据文件

Fig.5-35 The file of application info

个人征信报告的测试用例数据文件格式：姓名、证件号码、家庭电话、家庭地址、单位名称、单位地址。数据文件内容如图5-36所示：

```
[root@localhost 20130715]# pwd
/app/ApplicationFraud/data/20130715
[root@localhost 20130715]# cat pboc.txt
张三,11111111111111111,021-11111111,高科路11弄11号101,金融发展公司,龙东路1号
李四,222222222222222222,021-22222222,陇西路22弄22号202,浦东发展公司,徐东路1号
王五,333333333333333333,021-33333333,高科路33弄33号303,徐汇发展公司,高东路1号
[root@localhost 20130715]#
```

图 5-36 个人征信报告数据文件

Fig.5-36 The file of personal credit report info

黑名单的测试用例数据文件格式：黑名单类型、黑名单内容。其中黑名单类型取值为“04”的是地址黑名单。数据文件内容如图5-37所示：

```
[root@localhost 20130715]# pwd
/app/ApplicationFraud/data/20130715
[root@localhost 20130715]# cat blacklist.txt
04,徐汇区高东路1号
[root@localhost 20130715]#
```

图 5-37 黑名单数据文件

Fig.5-37 The file of blacklist info

其中：第一个测试用例数据是申请信息和个人征信报告的信息一致且不在黑名单数据中；第二个测试数据的申请信息和个人征信报告的家庭地址不一致且不在黑名单数据中；第三个测试数据的申请信息的单位地址和黑名单中的地址黑名单相似。

5.2.2. 功能验证

由于申请欺诈侦测是通过后台批量进行处理，所以需要查看各个服务的日志来确认申请欺诈侦测处理正常运行。

查看欺诈侦测管理子系统的数据接口服务日志内容如图5-38所示：

```
[root@localhost log]# pwd
/app/ApplicationFraud/log
[root@localhost log]# cat data-service.log
2013-07-15 21:46:00,163 INFO [DataServiceImpl] - find date:20130715 app data file: application_info.txt
2013-07-15 21:46:00,185 INFO [DataServiceImpl] - load data start!
2013-07-15 21:46:01,340 INFO [DataServiceImpl] - load 3 row data
2013-07-15 21:46:02,077 INFO [DataServiceImpl] - load data end!
2013-07-15 21:47:00,160 INFO [DataServiceImpl] - find date:20130715 pboc data file: pboc.txt
2013-07-15 21:47:00,177 INFO [DataServiceImpl] - load data start!
2013-07-15 21:47:01,245 INFO [DataServiceImpl] - load 3 row data
2013-07-15 21:47:01,877 INFO [DataServiceImpl] - load data end!
2013-07-15 21:48:00,144 INFO [DataServiceImpl] - find date:20130715 blacklist data file: blacklist.txt
2013-07-15 21:48:00,167 INFO [DataServiceImpl] - load data start!
2013-07-15 21:48:00,220 INFO [DataServiceImpl] - load 1 row data
2013-07-15 21:48:00,465 INFO [DataServiceImpl] - load data end!
2013-07-15 21:50:00,106 INFO [DataServiceImpl] - create date:20130715 fraud data file: fraud.txt
2013-07-15 21:50:00,133 INFO [DataServiceImpl] - create data file start!
2013-07-15 21:50:00,220 INFO [DataServiceImpl] - save 3 row data
2013-07-15 21:50:00,445 INFO [DataServiceImpl] - create data file end!
[root@localhost log]#
```

图 5-38 数据接口服务日志文件

Fig.5-38 The log file of data intface service

查看欺诈侦测管理子系统的欺诈侦测服务日志内容如图5-39所示：

```
[root@localhost log]# pwd
/app/ApplicationFraud/log
[root@localhost log]# cat detection-service.log
2013-07-15 21:49:00,102 INFO DetectionServiceImpl] - find app_no=10200701 application info
2013-07-15 21:49:00,104 INFO DetectionServiceImpl] - valid parameter is ok! app_no=10200701
2013-07-15 21:49:00,213 INFO DetectionServiceImpl] - call rule engine! app_no=10200701
2013-07-15 21:49:01,077 INFO DetectionServiceImpl] - rule engine return: RES_CODE = C, app_no=10200701
2013-07-15 21:49:01,160 INFO DetectionServiceImpl] - save fraud data! app_no=10200701
2013-07-15 21:49:01,193 INFO DetectionServiceImpl] - find app_no=10200702 application info
2013-07-15 21:49:01,197 INFO DetectionServiceImpl] - valid parameter is ok! app_no=10200702
2013-07-15 21:49:01,340 INFO DetectionServiceImpl] - call rule engine! app_no=10200702
2013-07-15 21:49:02,134 INFO DetectionServiceImpl] - rule engine return: RES_CODE = S, app_no=10200702
2013-07-15 21:49:02,160 INFO DetectionServiceImpl] - save fraud data! app_no=10200702
2013-07-15 21:49:02,332 INFO DetectionServiceImpl] - find app_no=10200703 application info
2013-07-15 21:49:02,337 INFO DetectionServiceImpl] - valid parameter is ok! app_no=10200703
2013-07-15 21:49:02,454 INFO DetectionServiceImpl] - call rule engine! app_no=10200703
2013-07-15 21:49:03,211 INFO DetectionServiceImpl] - rule engine return: RES_CODE = H, app_no=10200703
2013-07-15 21:49:03,398 INFO DetectionServiceImpl] - save fraud data! app_no=10200703
[root@localhost log]#
```

图 5-39 欺诈侦测服务日志文件

Fig.5-39 The log file of fraud detection service

查看规则引擎平台系统的服务接口日志内容如图5-40所示：

```
[root@localhost log]# pwd
/app/RuleEngine/log
[root@localhost log]# cat ruleEngine-service.log
2013-07-15 21:49:00,307 INFO DetectionServiceImpl] - request: app_no=10200701
2013-07-15 21:49:00,998 INFO DetectionServiceImpl] - response: app_no=10200701, RES_CODE = C
2013-07-15 21:49:01,456 INFO DetectionServiceImpl] - request: app_no=10200702
2013-07-15 21:49:02,067 INFO DetectionServiceImpl] - response: app_no=10200702, RES_CODE = S
2013-07-15 21:49:02,578 INFO DetectionServiceImpl] - request: app_no=10200703
2013-07-15 21:49:03,112 INFO DetectionServiceImpl] - response: app_no=10200703, RES_CODE = H
[root@localhost log]#
```

图 5-40 规则引擎服务接口日志文件

Fig.5-40 The log file of rule engine service intface

从三个服务的日志文件显示：验证的测试用例数据文件加载无异常；三条测试用例数据均被欺诈侦测；三条测试用例数据被规则引擎进行规则计算处理，无系统异常；产生返回给进件系统的欺诈侦测结果文件。

在欺诈侦测管理子系统中使用申请欺诈查询功能对验证数据查询, 查询结果如图5-41 所示：

图 5-41 申请欺诈查询页面

Fig.5-41 The page of application fraud query

查看欺诈侦测结果数据文件内容。欺诈侦测结果数据文件格式：申请编号，建议结果，欺诈模型评分，触发规则，原因码。数据文件内容如图5-42所示：

```
[root@localhost 20130715]# pwd
/app/ApplicationFraud/data/20130715
[root@localhost 20130715]# cat fraud.txt
10200701,C,11,,
10200702,S,58,B001,
10200703,H,96,C003,
[root@localhost 20130715]#
```

图 5-42 欺诈侦测结果数据文件

Fig.5-42 The file of fraud detection result

5.2.3. 验证结论

欺诈侦测结果对比如下：

(1) 第一条测试用例的欺诈侦测结果中欺诈模型评分小于 20，且没有触发规则，所以建议结果为“无欺诈”。与预期结果（“无欺诈”）对比一致；

(2) 第二条测试用例的欺诈侦测结果中欺诈模型评分大于 20 并且小于 80，所以建议结果为为“疑似欺诈”。原因是触发征信对比规则，申请信息与个人

征信报告信息数据不一致，建议人工调查。与预期结果（“疑似欺诈”）对比一致；

(3) 第三条测试用例的欺诈侦测结果中欺诈模型评分大于 80，所以建议结果为“高欺诈可能”。原因是触发地址黑名单检查规则，申请信息的单位地址和黑名单中的地址黑名单相似度高于设定值，建议拒绝。与预期结果（“高欺诈可能”）对比一致；。

经过侦测结果的对比，欺诈侦测结果与测试用例的预期结果一致，所以信用卡申请反欺诈系统能够有效的对欺诈进行识别。

6. 结 论

6.1.总结

本文信用卡申请反欺诈系统设计进行了详细描述。对信用卡申请反欺诈系统的各个功能进行了设计。主要是进行系统总体设计和系统功能设计。实现了银行信用卡申请反欺诈系统的设计和实现。

本文取得的主要成果有:

(1) 规则引擎平台子系统设计与实现。

并提供规则管理系统功能来实现业务人员的申请反欺诈规则的维护工作，提供统计报表来监控规则质量如何；

(2) 欺诈侦测管理子系统设计与实现。

建立信用卡申请反欺诈日常管理系统，完善信用卡审批流程；

(3) 达到了系统预期设计目标。

经过系统试运行与使用，对信用卡申请反欺诈的防范水平有所提高。

6.2.改进建议

信用卡申请反欺诈系统在信用卡申请审核过程中发挥重要作用，减少了审核人员的工作压力，降低了审核成本。但是依然存在一些问题，需要进一步完善和改进：

(1) 报表功能相对简单

随着信用卡申请审核管理的逐步加强，对统计报表功能要求越来越高。建议引入统计报表相关的开源项目，来增强统计报表功能；

(2) 集团欺诈管理功能不完善

集团欺诈信息查询管理功能还不能对集团关联的申请历史信息进行查询，只是查询相对独立的集团信息比如：单位名称、单位地址等。需要完善集团欺诈关联信息的查询管理功能，能够查询到集团欺诈相关关联历史申请信息。

参考文献

[1] 陈建，现代信用卡管理[M]，北京，中国财政经济出版社，2005，11-321.

[2] 杨德忠，信用卡欺诈申请的现状、成因及防控措施[J]，中国信用卡，2010(12)，50-53.

[3] 刘燕，丁辉，防范信用卡申请业务欺诈风险的中美对比[J]，银行家，2012(11)，100-101.

[4] 陈建，信用评分模型技术与应用[M]，北京，中国财政经济出版社，2005，11-87.

[5] Han Lu, Han Liyan, Zhao Hongwei, Credit Scoring Model Hybridizing Artificial Intelligence with Logistic Regression[J], Journal of Networks, 2013, 8 (1), 253-261.

[6] 田田，基于逻辑回归的信用卡申请评分模型研究与实现[D]，北京，中国地质大学，2011.

[7] David West, Neural network credit scoring models [J], Computers and Operations Research, 2000, 27(11)

[8] 王静，王延清，何德权，基于多层前馈神经网络的个人信用评分模型[J]，经济师，2004(12),20-21

[9] 李国乐，Java 规则引擎与其 API(JSR-94)[J/OL]，http://www.ibm.com/developerworks/cn/java/j-java-rules/.

[10] Shouhong Wang, Hai Wang, Business Rule Management for Enterprise Information Systems, Information Resources Management Journal (IRMJ) [J], 2010, 23(1), 53-73.

[11] CIS587: The RETE Algorithm[J/OL], http://www.cis.temple.edu/~ingargio/cis587/readings/rete.html.

[12] 张渊，夏清国，基于 Rete 算法的 JAVA 规则引擎[J]，科学技术与工程，2006(11)，1548-1550.

[13] Carlos García-Montoro, Mario González, Emilio Vivancos, Vicente J. Botti, Comparing the Execution Time of the Rete and Arlips2 Pattern Matching Algorithms[J], Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, 2006, l(10), 23-30.

[14] The Java Community Process(SM) Program, JSR 94: JavaTM Rule Engine API[J/OL], http://jcp.org/en/jsr/detail?id=94.

[15] JBoss Community Projects, Drools Introduction and General User Guide [J/OL], http://docs.jboss.org/drools/release/5.5.0.Final/droolsjbpm-introduction-docs/html_single/index.html.

[16] IBM，Business Rules Management System (BRMS) [J/OL], http://www-01.ibm.com/software/websphere/products/business-rule-management/.

[17] FICO, Blaze Advisor® business rules management [J/OL], http://www.fico.com/en/Products/DMTools/Pages/FICO-Blaze-Advisor-System.aspx.

[18] 程昌秀，于滨，一种基于规则的模糊中文地址分词匹配方法[J]，地理与地理信息科学，2011, 27(3), 26-29.

[19] 阎宏，Java 与模式[M]，北京，电子工业出版社，2002, 10-359.

[20] Vishal Gour, Dr. S.S.Sarangdevot, Govind Singh Tanwar, Anand Sharma, Improve Performance of Extract, Transform and Load (ETL) in Data Warehouse[J], International Journal on Computer Science and Engineering, 2010, 2(3), 786

[21] 孙安健，王星，闫晓瑜，通用 ETL 工具的研究与实现[J]，计算机应用与软件，2012，29(12)，175-178.

致 谢

在此，向在我论文研究和撰写过程中曾给予我关心和帮助的人们致以最诚挚的感谢!

我要衷心感谢我的导师步丰林教授，在整个论文撰写过程中，步丰林教授给予了我悉心的指导和帮助，经常在周末休息时间约见我，了解论文进度情况，解答论文研究和撰写过程中的疑问，使我的论文能够顺利完成。步丰林教授不仅传授给我论文研究的方法和思考方式。他对教学严谨的态度以及科学研究的热爱使我获益匪浅，受益终身，在此我向步丰林教授表示最由衷的敬意和最真诚的感谢!

感谢我的第二导师张艳红教授，在我的论文研究过程当中，给予我很多无私的关怀和帮助。在此我向张艳红教授表示真诚的由衷的感谢!

感谢我的公司同事，在我的论文的研究和撰写过程中，公司同事帮我查找资料，解答我的问题，谢谢你们在我攻读硕士学位期间的大力支持。

感谢上海交通大学研究生学院和软件学院的老师们的辛勤工作，使我在攻读硕士学位能够顺利的完成学业!感谢所有传授知识给我的老师们！感谢所有给予我帮助的老师们！感谢所有给予我帮助的同学们!

攻读学位期间发表的学术论文目录

[1] 张峰，银行信用评分模型开发平台的设计，上海交通大学内部网络公示，2013