

基于受限中文的轻型业务规则管理系统的设计与实现^{*}

徐 黎 冯元勇 糜宏斌 孙玉芳

(中科院软件研究所 开放系统与中文信息处理中心 北京100080)

摘 要 业务规则管理是实现企业灵活多变的业务控制和客户服务的理想方案。本文介绍了业务规则管理系统的基本思想,并面向业务分析人员设计了一种受限中文业务规则语言,实现了一个轻型业务规则管理系统。

关键词 业务规则管理系统,业务规则,业务对象模型,业务规则语言

Design and Implementation of a Small Business Rule Management System Based on Constrained Chinese Natural Language

XU Li FENG Yuan-Yong MI Hong-Bin SUN Yu-Fang

(Open System&Chinese Information Processing Center, Institute of Software, Chinese Academy of Sciences, Beijing 100080)

Abstract Business logic is the most dynamic component in business applications. Business Rule Management System captures business logic as rules, and lets users quickly and securely change business rules, redeploying without altering software code. In this paper, the principles of Business Rule are discussed and the design of a small Business Rule Management System based on constrained Chinese natural language is presented.

Keywords Business rule management system, Business rule, Business object model, Business rule language

1 引言

面对快速变化的市场环境,企业自身往往需要不断地调整业务以适应变化的需求。而在传统的应用系统中,业务规则被程序员采用程序设计语言、数据库定义和操纵语言编写,业务规则的变化常常要引起代码的变化,甚至数据库结构的变化,使传统应用系统难以维护,难以及时适应需求的变化。

业务规则管理系统^[1](BRMS, Business Rule Management System)通过对业务规则进行规范化表示、集中的管理和维护、灵活的部署、高效率的执行,将企业内部不确定的业务策略和流程从应用程序中分离出来,使得企业能够方便灵活地对它们进行管理和更改。

本文介绍了业务规则管理系统的原理,并面向业务分析人员设计了一种受限中文业务规则语言,实现了一个轻型业务规则管理系统。

2 业务规则管理系统的原理

2.1 业务规则

业务规则的字面意思是指导业务活动的规则或策略,它广泛存在于工商业活动中。例如银行的贷款申请,商品促销规定、保险条例等。例如:“具有A级信用度的个人客户可以透支1万元”。

业务规则目前尚无工业标准定义,一个比较公认的定义是由业务规则组织^[2](Business Rule Group)给出的,包含两个层面的理解。从企业业务的角度来看,“业务规则是支持企业决策,影响或控制企业业务行为的指示”;从计算机信息系统的角度来看,“业务规则是一条语句,它定义或约束业务的

某些方面。其目的是对业务结构做出断言,或者对业务行为施加控制和影响。”业务规则事实上表达了“条件-动作”的关系,即当某种条件满足时,执行规则规定的动作,包括对数据或状态的约束性检查、修改等。

业务规则语句应当满足原子性(Atomic)、确定性(Unambiguous)、简洁性(Compact)、一致性(Consistent)和相容性(Compatible)^[1]。

业务规则的分类方法很多,按内容可分为三类^[1]:

1) 结构类规则(Structure):表达数据和状态必须满足的约束关系。

2) 动作类规则(Action):测试条件,结果为TRUE时,启动指定的业务事件或发送消息。

3) 定义类规则(Definition):定义术语,以及术语间定量或定性的关系。

2.2 业务规则的表述

业务规则语句的基本结构为 IF<conditions>, THEN<actions>。<conditions>部分通常称为左部(LHS, Left Hand Side),由若干模式(pattern)组成;<actions>部分称为右部(RHS, Right Hand Side),指示当满足LHS的模式时应当执行的操作。

业务规则的表述基于业务术语和业务事实。术语(term)是在特定业务领域具有明确定义的名词或名词短语,例如“客户”、“信用级别”。事实(fact)是对业务状态的描述,例如“编号A001的客户信用额度为5000人民币”。实质上规则左部表示对事实的判断,由若干事实序列组成,规则右部是对事实的修改和创建。考查事实需要理解对象间的关系,为此我们建立业务对象模型(BOM, Business Object Model)来描述信息系统

^{*} 基金项目:863专项(编号:2003AA1Z2110);中国科学院知识创新工程方向性项目(编号:KG CX2-SW-504)。徐 黎 硕士研究生,研究方向为中文信息处理、数据库。冯元勇 博士研究生,研究方向为中文信息处理。糜宏斌 副研究员,研究方向为系统软件、中文信息处理。孙玉芳 研究员,博导,研究方向为系统软件、中文信息处理。

中存在的对象类型,以及它们之间的关系。BOM 是描述、理解和执行业务规则的基础。业务规则使用的术语由 BOM 定义,它描述的事实反映 BOM 内业务对象间的约束关系,执行的结果直接表现为对业务对象属性的修改或方法的调用。

规则的表述有三个层次:非形式化描述、技术形式和形式化描述^[1]。一般说来,规则的执行部件(通常称为业务规则引擎, Business Rule Engine)只能理解形式化的业务规则,如 CLIPS^[3,4]。但业务规则来自应用领域,因而必须能够被非技术人员理解和编辑,他们倾向于非形式化的规则描述,如类自然语言。技术形式是规则表述的中间形式,既便于理解,又具备一定的结构性。通常,业务人员采用前一种方式表述规则,开发人员则多采用后两种方式。业务规则管理系统提供了编译工具,将不同表述层次的业务规则语句转换为业务规则引擎的执行语言。

2.3 业务规则管理系统

业务规则管理系统(BRMS)完成对业务规则的建立、管理、维护、部署、运行等。BRMS 的技术基础源于传统的专家系统^[3],主要包括规则的形式化表示、规则编程语言、规则匹配与执行。同时为适应企业应用环境, BRMS 进一步包含了业务规则的开发环境、完整的业务对象模型、与其他系统部件的数据和指令接口,良好的组件化结构等。

业务规则管理系统的体系结构见图1。

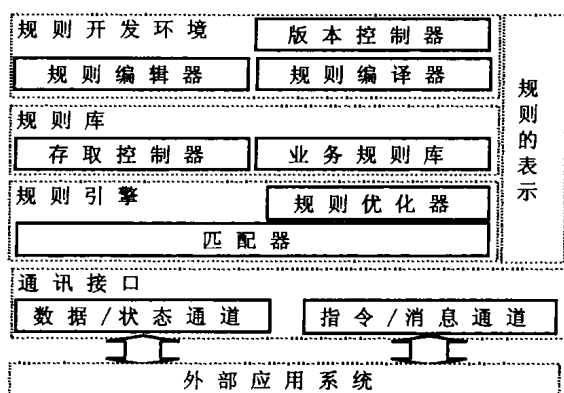


图1 业务规则管理系统的体系结构

1) 规则开发环境是方便用户定制业务规则的部件,并提供了编译工具将用户输入的业务规则翻译成规则引擎能够执行的规则代码。

2) 业务规则引擎是业务规则的执行部件,也是业务规则管理系统的核心。为适应企业应用集成的需要,规则引擎对外提供多种接口方式,主要有三种:嵌入式的紧密集成,规则引擎以库函数的形式被使用;组件式的集成,规则引擎以组件的方式运行在组件容器中;消息集成,规则引擎作为独立的服务器运行。

3) 规则库是存贮业务规则语句及其属性信息的地方。用户录入的规则经编译后保存在规则库中,供规则引擎调用执行。

3 业务规则管理系统的设计

3.1 业务规则管理系统的组成

我们参考业务规则管理的原理,在 Java 平台上设计并实现了一个轻型的业务规则管理系统。系统组成见图2。

规则开发环境中集成了规则编辑器和编译器,图形界面方便用户建立业务对象模型,录入规则,并提供相应工具将规

则语句转换成规则引擎的执行语言。

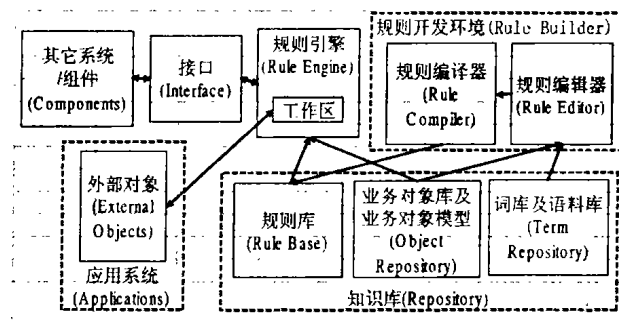


图2 业务规则管理系统的组成

规则引擎是规则的执行部件。我们采用的规则引擎是 JESS6.0^[5],以 Java 类库的方式提供。应用系统将外部对象的引用加载到引擎的工作区,启动引擎。引擎运行时装载规则,高效地完成规则匹配与执行。

知识库逻辑上分为三部分:规则库、业务对象模型库和语料库。业务规则经过用户分类后,存放在规则库中。业务对象模型描述了业务规则所作用的或者涉及到的业务对象。词库及语料库为用户编辑规则提供可选词汇。

3.2 业务对象模型

业务规则的形成和执行都离不开业务对象。我们建立业务对象模型来描述业务领域中所有的业务对象,它记录了业务对象的名称、属性、方法、参数类型、返回类型,以及对象之间的联系等。业务规则的制定以业务对象模型为基础,对其中的业务对象进行约束。我们的系统以保险为例,部分业务对象模型设计如图3所示。

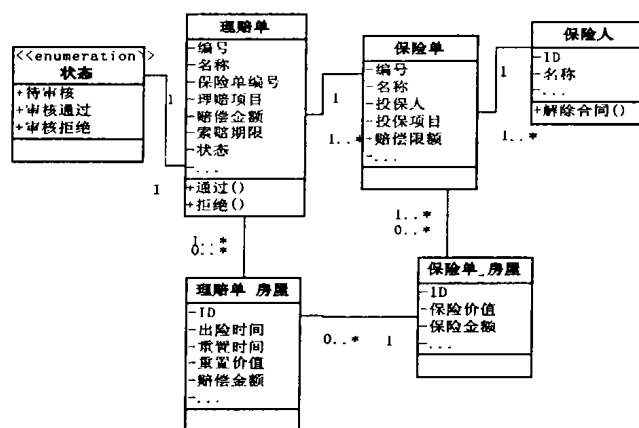


图3 业务对象模型

3.3 业务规则语言的设计与实现

如前所述,业务规则语句的具体表达形式依据用户的不同而不同。我们分别为业务分析人员和开发人员设计了相应的业务规则语言。它们最终都会被系统转换成规则引擎的执行语言。图4展示了我们系统的三层规则语言模型。

3.3.1 受限中文业务规则语言(CCBRL)及其翻译 业务分析人员通常不具备计算机编程知识,只能理解和表述自然语言形式的规则。但是由于中文的开放性和复杂性使得中文自然语言的表达非常灵活,计算机处理起来比较困难。为了降低计算机理解的难度,根据业务规则语句的特点,我们定义了受限中文业务规则语言 CCBRL(Constrained Chinses Business Rule Language),通过提供语法模板和词库的方法,帮助业务分析人员直观、快速地制定规则。

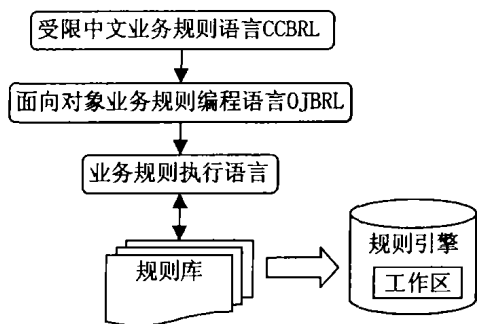


图4 业务规则语言模型

CCBRL 对业务规则语句的表达形式进行了限制,具体表现在两个方面:业务规则语句的句法结构受限,用户需依据 CCBRL 制定的句法模板录入规则;业务规则语句的词汇受限,用户使用的词集必须预先定义。CCBRL 定义了部分关键词来标识语法模板的结构。这种方案既使得规则语句的表达尽可能自然易懂,同时使得规则语句的语法结构比较清晰,有利于计算机确定词语的角色进而确定其语义,从而降低翻译的难度,并使得语义模型容易建立,语料库规模容易控制。

1) CCBRL 的语法定义。CCBRL 的部分语法设计如图5所示。定义业务规则为:〈规则〉::=〈条件〉〈处理〉。〈条件〉是对业务对象属性的测试、比较等,我们定义:〈条件〉::=〈主体〉〈比较〉〈宾体〉

例如“理赔单的赔偿金额小于等于保险单的赔偿限额”,其语法分析为:〈主体〉理赔单的赔偿金额/〈主体〉〈比较〉小于等于/〈比较〉〈宾体〉保险单的赔偿限额/〈宾体〉。

定义〈处理〉::=〈设置〉|〈方法调用〉。〈设置〉指对业务对象属性的修改,〈方法调用〉表示对业务对象的方法或系统预定义方法的调用。例如“理赔单的状态设为‘审核通过’”的语法分析结果为〈设置〉〈主体〉理赔单的状态/〈主体〉〈操作〉设为/〈操作〉〈宾体〉‘审核通过’/〈宾体〉/〈设置〉。

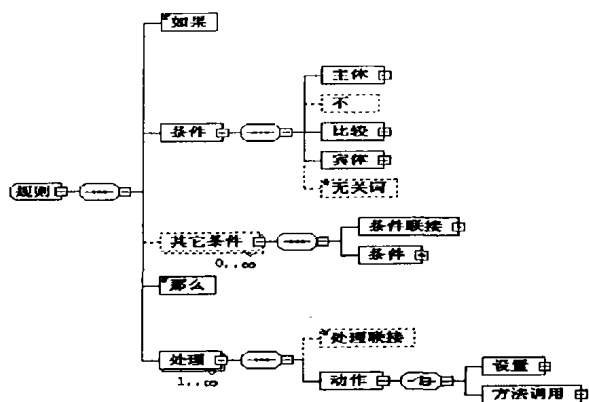


图5 CCBRL 部分语法设计

确定规则语句涉及的业务对象及其嵌套关系是计算机理解业务规则的关键,也是难点。为此,我们在模板的〈主体〉和〈宾体〉内设计了〈上级对象〉标签,记录对象间的上下位关系,方便后面的语义分析和翻译。

2) CCBRL 的语义模型。CCBRL 的部分语义模型设计如图6所示。〈规则对象〉记录该规则语句涉及到的全部业务对象。语法分析的〈条件〉部分被转换成一个或多个〈模式〉的“与”连接。〈模式〉定义为:

〈模式〉::=〈比较量〉〈比较算子〉〈参照量〉

例如“理赔单的赔偿金额小于等于保险单的赔偿限额”,

结合业务对象模型,其语义可分解成两个模式,分别为〈比较量〉理赔单. 保险单编号/〈比较量〉〈比较算子〉==/〈比较算子〉〈参照量〉保险单. 编号/〈参照量〉和〈比较量〉理赔单. 赔偿金额/〈比较量〉〈比较算子〉≤/〈比较算子〉〈参照量〉保险单. 赔偿限额/〈参照量〉。

语法分析的〈处理〉部分统一转换成〈动作〉,包括运算、函数调用和业务对象方法调用。

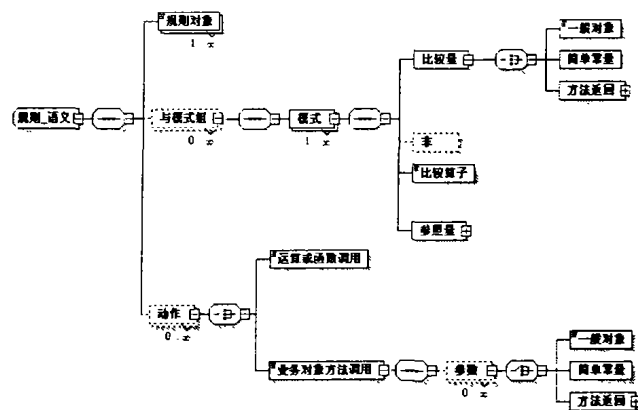


图6 CCBRL 部分语义设计

3) CCBRL 的词库。业务规则离不开具体的业务领域,它反映并约束了业务领域中业务对象的状态,因而业务规则语句使用的术语也必须能够对应到相关业务领域中。在 CCBRL 中,业务规则使用的词汇分为两部分:专业词汇和通用词汇。专业词汇来自应用领域,由业务对象模型提供。通用词汇,如代词、连词、系统预定义的关键词等则由通用词库定义。这种方法确保了规则语句的合法性,降低了分词的难度。但词集受限,容易出现词汇不能满足要求的情形。因此系统提供了词集维护的功能,主要是通过用户交互的方式,增加或删除词库中的术语。

4) CCBRL 的翻译。业务规则语句通常简短,且含义清晰明确,出现歧义的可能性很小,语句中的词语可负担的语义角色不多。因此,我们设计 CCBRL 的翻译流程如图7所示。

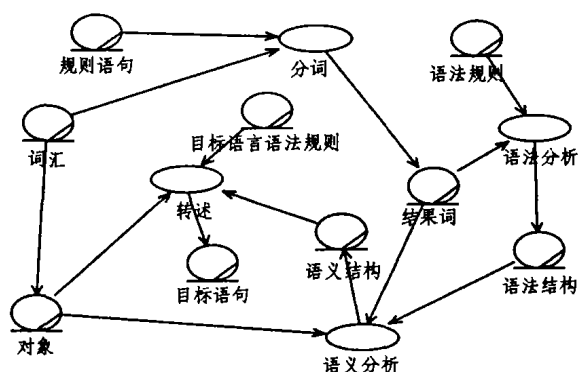


图7 CCBRL 的翻译流程

它遵循通常的中文翻译过程,大致分为:分词、语法分析、语义分析、转述。翻译的目标是将源语句转述成面向对象业务规则编程语言 OBJRL 的形式。

1) 分词:依据业务对象模型和通用词库定义的词汇,对业务规则语句进行分词。分词策略为正向最大匹配。

2) 语法分析:CCBRL 定义了部分关键字,使得我们能够比较容易地判断语句的语法结构。任何词在规则语句中都只有一个语法功能。语法分析根据 CCBRL 的语法定义,为每个词语加上语法标签。

(下转第99页)

体可以帮助编辑人员更规范、更清楚地描述其所要表达的概念和知识。

Ontology 领域本体是本系统的一个技术关键。一般说来,本体是对一个领域中的公认的知识及其属性、它们之间的关系的一种明确的、形式化的描述。本体明确定义了一个领域中不同的人 and 计算机都能理解的基本概念(Concept)以及概念之间基本关系。通过本体可以实现人与人之间、人与计算机之间,计算机与计算机之间信息共享。

(2)数据层。数据层包括多媒体库及其索引库、元数据库及其索引库。

多媒体数据库实现对多媒体内容的存储和管理,多媒体数据库将采用“对象-关系”模型来实现。在多媒体数据库后面将有一个可以根据需要不断扩展的后端处理模块,负责对多媒体文件添加水印,利用成熟的工具或者算法实现基于多媒体内容的聚类索引,直接为未来的检索服务,提高检索效率。

元数据库实现对元数据的存储和管理。从便于元数据互操作的角度出发,元数据可以采用 Native XML 数据库。但是,考虑到目前该技术不是非常成熟,在我们以前的试应用中效果不太理想。因此,我们采用传统的关系数据库进行管理,但提供元数据转换成 XML 文件的工具。

(3)服务层。服务层包括综合检索模块和显示模块。

综合检索模块是本系统最复杂和最具特色的一项功能。综合检索模块将实现基于用户提交的关键词(概念)、基于元数据属性以及基于多媒体内容的综合检索。基于关键词的检索将以领域本体为基础,研究和基本实现以本体为基础的信息检索方式。基于内容的检索将以图像为基础,实现基于图像

颜色、形状、纹理等的特征检索。视频和音频的基于内容检索技术作为研究的方向。

显示模块实现对多媒体内容的浏览和显示,包括动态生成缩略图(Thumbnail)等。

结论 通过研究基于内容的多媒体检索技术,本系统将实现对多媒体内容元数据的抽取、编辑和管理等功能,利用数据库实现对多媒体内容的集中管理,利用元数据实现对多媒体内容的查询、检索和定位。在此基础上,研究和引入领域本体,研究和开发基于概念、属性和内容的多媒体综合检索系统。

经过初步的研究和实践,本论文提出了多媒体检索系统的框架设计,在具体实现过程中还有许多问题需要深入研究,比如:多媒体数据存储和管理技术、多媒体元数据互操作技术、多媒体元数据自动分析和抽取技术、领域本体的研究等。同时,这套设计方案本身还需要进一步改进和完善。

参考文献

- 1 Kiranyaz S, Caglar K, Guldogan E, et al. MUVIS: A Content-based Multimedia Indexing and Retrieval Framework[J]. IEEE, 2003(2)
- 2 Chang S-F, Chen W, Meng H J, Sundaram H, Zhong Di. VideoQ: An Automated Content Based Video Search System Using Visual Cues[J]. ACM, 1997(2)
- 3 Smith J R, Chang S-F. VisualSEEK: a fully automated content-based image query system[J]. ACM, 1996(1)
- 4 Weber R, Bolliger J, Gross T, Schek H J. Architecture of a Networked Image Search and Retrieval System[J]. ACM, 1999(11)
- 5 MPEG-7 [S]. <http://ipsi.fhg.de/delite/Projects/MPEG7/>
- 6 Virage. www.virage.com

(上接第89页)

3) 语义分析:依据语法分析的结果,以及规则语句中每个词语的语义关系(在业务对象模型中反映),确定规则语句的语义。语义分析为词语添加语义标识,并将中文表示转换成系统内部引用时的英文标识。

4) 转述:根据语义分析的结果,将〈模式〉和〈动作〉翻译成 OBJRLL 形式,完成整个中文规则语句的转换。

3.3.2 面向对象业务规则编程语言 OBJRLL 开发人员具备计算机对象设计和编程知识,为了方便他们定制规则,加速业务规则的开发,我们采用类 Java 语言的文法,定制了面向对象业务规则编程语言 OBJRLL (Object-oriented Java-like Business Rule Language)。

在 OBJRLL 中,业务规则的基本形式为:IF 〈Pattern〉 THEN 〈Action〉。在〈Pattern〉和〈Action〉中采用类 Java 的规范访问对象。部分语法如下:

```
(RuleStatement)::=[ruleHeader] IF 〈Pattern〉 THEN 〈Action〉
(ruleHeader)::= ruleName = 〈string〉; [priority = 〈integer〉];
(argumentList)::= argument 〈parameterList〉
(Pattern)::= 〈objectExpression〉 {〈logicLinkWord〉 〈objectExpression〉}
( objectExpression ) :: = [ 〈 variable 〉; ] 〈 object 〉 (
  〈conditionExpression〉 {〈object〉}
(Action)::= 〈ActionExpression〉+
( ActionExpression)::= 〈variable〉. 〈object〉(〈par〉); {〈object〉. 〈object〉(〈par〉);
```

该语言支持面向对象的功能,并根据业务规则的特点,具备变量指代、集合运算等功能。例如下面这条网上购物时的规则:“如果购物车中有鱼,则在购物车中添加鱼片赠品。”

```
ruleName = "ruleAddSampleFlakedFish";
IF ?sc:ShoppingCart { 'fish' IN Items[]}
```

```
THEN ?sc.addItem("SampleFlakedFish");
```

3.3.3 业务规则执行语言(Rule Execution Language)

是规则引擎的执行语言。我们系统采用的规则引擎是 JESS6.0,相应的执行语言是 CLIPS。用 OBJRLL 编写的规则需要编译成 CLIPS 的形式才能被规则引擎执行。我们定制了词法、语法文件,采用相关编译器构造技术生成了 OBJRLL 的编译器。

总结 本文介绍了业务规则管理系统的原理,并设计了一个三层业务规则语言模型,实现了一个轻型业务规则管理系统。系统为业务分析人员定制了受限中文业务规则语言,提供实用方便的规则开发工具,将企业的业务逻辑及其实现相分离,并具备良好的集成性,使得企业的信息系统能够更快更好地适应业务需求的变化。在今后的工作中,我们将进一步完善该系统的功能,如为业务规则引擎提供更多的封装方式,方便系统的应用集成;完善中文支持功能,为业务人员定制灵活方便的中文业务规则模板,方便规则的录入。

参考文献

- 1 Morgan T. Business Rules and Information Systems. Boston: Addison-Wesley, 2002
- 2 Business Rule Group. <http://www.businessrulesgroup.org/>, 2004
- 3 Giarratano J, Riley G 著. 印鉴, 刘星成, 汤庸译. 专家系统原理与编程. 北京: 机械工业出版社, 2000
- 4 Giarratano J C. CLIPS User's Guide. v6. 20. Boston: PWS, 2002. 3
- 5 JESS. <http://herzberg.ca.sandia.gov/jess/>. 2004. 3