

一种分离业务规则的灵活业务过程建模方法

杜栓柱¹, 黎 夔^{1,2}

DU Shuan-zhu¹, LI Nao^{1,2}

1.中国科学院 软件研究所, 北京 100080

2.中国科学院 研究生院, 北京 100039

1.Institute of Software, Chinese Academy of Sciences, Beijing 100080, China

2.Graduate School of Chinese Academy of Sciences, Beijing 100039, China

E-mail: dusz@intec.iscas.ac.cn

DU Shuan-zhu, LI Nao. Adaptive business process model separating business rules. Computer Engineering and Applications, 2008, 44(30): 232-236.

Abstract: Aiming at the incessant change of business applications, the business process model Business Process Model for Dynamic Business Logic (BPM4DBL) is built. BPM4DBL separates changeable business logic from process control structure, and describes changeable business logic with declarative business rules. Afterwards, business rule definition, classification and execution language are given. Finally, a case study is presented.

Key words: business process model; business rules; adaptivity

摘 要: 针对动态环境中业务应用的不断变化, 将易变的业务逻辑从过程控制结构中分离, 并且通过业务规则的声明性表达式描述, 由此构建了业务过程模型 BPM4DBL (Business Process Model for Dynamic Business Logic)。在建立规则元素的模型定义后, 给出了业务规则的定义、分类和可执行语言描述, 最后给出一个 BPM4DBL 的具体应用实例。

关键词: 业务过程模型; 业务规则; 灵活性

DOI: 10.3778/j.issn.1002-8331.2008.30.071 **文章编号:** 1002-8331(2008)30-0232-05 **文献标识码:** A **中图分类号:** TP311

1 引言

在日趋激烈的全球化市场竞争态势下, 业务过程模型如何适应政府或企业等组织部门不断调整变化的业务逻辑, 建立灵活的业务过程模型, 成为当前业务过程建模研究中面临的主要问题^[1-2]。根据模型的构建目标, 目前代表性的业务过程模型可分为三类^[3-4]: 一类应用于业务过程的一般性描述, 如甘特图和 IDEF3 (Integrated DEFinition Method 3), 一类应用于业务过程重组, 如流程图和 IDEF0 (Integrated DEFinition Method 0), 一类应用于 PAISs (Process-Aware Information Systems) 信息系统的开发和执行, 如 EPC (Event-driven Process Chain)、Petri Net 和 BPMN (Business Process Modelling Notation)。本文研究面向 PAISs 的灵活业务过程模型。

一般来说, 灵活业务过程模型研究在业务逻辑不能预先确定或经常变化情况下, 如何建立灵活的流程结构, 一方面使得静态建立时模型能够描述未能预先确定的流程, 另一方面动态运行时模型能够支持业务逻辑的即时调整。文献[5]区分了声明性和命令性业务过程模型, 其中前者仅定义流程“做什么”(What)而无需定义“如何做”(How), 流程结构在运行时一定约束下动态生成, 因而可以支持不能预先确定的业务逻辑, 由此基于线性时序逻辑 (LTL) 建立了 ConDec 模型。相比较而言, 命

令式模型在流程建立时即需要确定流程中各个任务的执行次序, 其一般通过引入灵活的结构算子来实现模型的灵活性: 文献[6]定义了顺序、分支和包含等 5 个基本约束结构块算子, 支持上述结构块在运行时的即时生成。命令式模型由于易于使用得到了工业界的广泛支持, 而通过在命令式模型中引入声明式语义, 从而获得灵活业务过程模型的方法也得到目前研究者的重视, 成为当前灵活性业务过程建模研究的重点: 文献[7]通过定义过程的状态, 给出了 BPEL4WS (Business Process Execution Language For Web Services) 中触发业务规则的事件点, 并且定义了业务规则的基本表示形式。这里注意到, 当前研究集中于如何在命令式模型的表示层中引入声明性表达式, 以及如何在运行时支持声明性表达式执行, 而未关注于如何从业务过程模型中分离易变业务逻辑的方法, 从而不能有效支持业务逻辑的动态调整。

为此, 本文在分析业务过程特点基础上, 将易变的业务策略从业务过程结构中分离, 并且通过业务规则将此表示为声明性过程语义, 由此给出了业务过程模型 BPM4DBL (Business Process Model for Dynamic Business Logic) 的表示和分析方法, 随后给出了一个具体的应用实例。

基金项目: 国家自然科学基金 (the National Natural Science Foundation of China under Grant No.60673121)。

作者简介: 杜栓柱 (1971-), 男, 博士, 主要研究方向为业务过程管理理论及技术; 黎夔 (1974-), 女, 硕士, 主要研究方向为知识管理理论及技术。

收稿日期: 2007-11-26 **修回日期:** 2008-02-26

2 业务过程和业务规则基本概念

业务过程是政府或企业等组织内或组织间业务逻辑的反映, 表现为任务之间的有序关系集合, 其中任务之间的关系描述了输入到输出之间的转换, 从而实现输入的增值, 或达到预期的目标^[8]。一般地说, 在目前动态的市场经济环境中, 业务逻辑的动态性主要体现在业务策略的动态性上, 据此可以将业务过程分为两部分, 一是稳定的控制结构, 二是易变的业务策略, 其中前者表现为组成过程的任务、任务间的数据关系和任务间的控制关系等相对稳定, 后者表现为任务和任务间的非功能性约束、任务之间的变迁条件等容易变化。为了适应业务过程中易变业务策略的调整, 本文将通常业务过程模型中混合的业务策略和控制结构相互分离, 并且用业务规则的方式来描述业务策略。

业务规则是人工智能理论中的规则应用于业务领域所形成的概念, 目前尚未有统一、标准的定义, 一个比较公认的定义是由业务规则组织(Business Rule Group, BRG)给出的: 从业务应用的角度来看, 业务规则是支持领域决策, 影响或控制领域业务行为的指示; 从计算机信息系统的角度来看, 业务规则是声明语句, 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^[9]。BRG 定义了结构断言、行为断言和派生三类规则, 其中结构断言规则定义了组织中某方面的概念或者事实; 行为断言规则定义了限制或控制组织行为的约束条件声明; 派生规则定义了从业务中其它知识可以继承的知识声明。目前, 业务规则一般通过 XML 格式的语言来描述, 典型的业务规则语言包括 RuML^[10]、SRML (Simulation Reference Markup Language)^[11]等。本文中将根据从业务过程中分离的业务策略特点来描述业务规则。

例如, 对图 1 所示的科研报销流程来说, 流程所涉及的任务(包括报销单填写、课题负责人审核、课题组长审核、科研会计审核、所级财务主管审核、财务处长审核和出纳支付)、任务之间的结构关系(如课题负责人审核和课题组长审核的输入为报销单填写)等相对稳定, 而任务的非功能性约束(如财务处长审核的执行时间约束 CS2)、任务之间的变迁条件(例如 CO1、CO2)等业务策略则容易发生变化。如随着管理方式的变化, 流程执行中“CO1 和 CO2”可能变化为“报销金额大于 3 000, 并且报销人员不是课题负责人”。在目前的业务过程模型中, 任务及任务之间的结构关系通过活动元素及其之间控制结构描述, 而业务策略则表示为活动属性、活动间变迁的条件、活动的异常

处理约束, 因此局部业务策略的调整不但需要了解全局的流程结构, 而且影响到了整体的业务过程模型, 需要重新生成新的业务过程模型。本文通过用业务规则来描述活动之间变迁条件等易变的业务策略, 可以使得业务人员在流程运行中方便地局部修改业务策略, 而不需要重新生成新的过程模型。

3 分离规则的业务过程模型 BPM4DBL

3.1 BPM4DBL 基本概念

定义 1 (BPM4DBL) BPM4DBL 为三元组, $BPM4DBL=(AS, RS, FS)$, 其中 AS 为活动集, RS 为规则集, $AS \cap RS = \emptyset$, $AS \cup RS \neq \emptyset$, FS 为活动和规则之间的连接关系集, 即 $FS \subseteq (AS \times RS) \cup (RS \times AS)$ 。

从上述定义可以看出, BPM4DBL 中活动表示业务逻辑中需要执行的任务, 规则描述业务逻辑中任务或任务之间的业务策略, 活动和规则元素之间的关系描述了业务逻辑的控制结构。活动元素是 BPM4DBL 的基本实体, 规则元素从属于活动元素, 每个活动分别包括前规则和后规则两类元素, 分别表示活动在执行前和执行后需要满足的业务策略。一个 BPM4DBL 基本单元由活动及其前后规则元素组成, 如图 2, 其是过程中的最小结构单元, 多个基本单元的组合格形成过程模型。

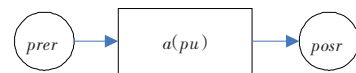


图 2 一个 BPM4DBL 基本单元

定义 2 (BPM4DBL 基本单元) BPM4DBL 基本单元为三元组, $pu=(prer, a, posr)$, 其中 a 为活动元素, $prer$ 和 $posr$ 分别为 a 的前规则和后规则元素, $prer, posr \in RS, a \in AS$ 。

基本单元中规则元素从属于活动元素, 因此 BPM4DBL 控制结构本质上是基本单元中活动之间的控制依赖关系; 又由于每个活动都包括前和后规则, 基本单元组合为顺序、与分支/汇聚、或分支/汇聚、循环等控制结构时, 需要遵循一定的组合策略:

(1) 两个基本单元组合为顺序结构时, 前序单元中的后规则和后序单元中的前规则合并为两个单元的公共规则元素。如图 3(a) 所示, 顺序结构 pu_1 和 pu_2 中的 $posr_1$ 和 $prer_2$ 合并为 $posrer_{12}$ 。

(2) 多个基本单元组合为分支结构时, 分支点前序单元的后规则 and 分支点后序单元的前规则合并为多个基本单元的公

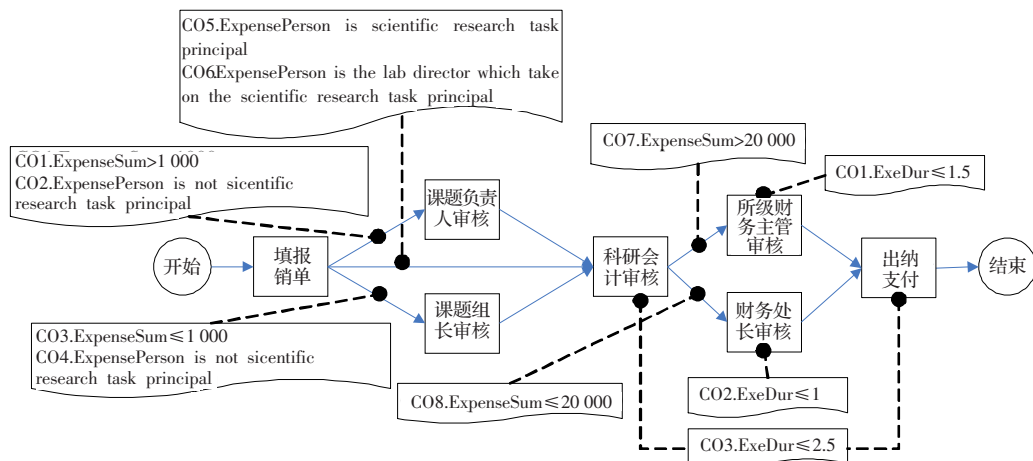


图 1 科研报销流程

共规则元素,合并后的规则元素为新的分支点。如图3(b)所示,前序单元 $pu1$ 的 $posr1$ 、后序单元 $pu2$ 的 $prer2$ 和 $pu3$ 的 $prer3$ 合并为 $posr123$ 。

(3)多个基本单元组合为汇聚结构时,具体组合方式与(2)类似,合并后的规则元素为新的汇聚点。

(4)多个基本单元组合为循环结构时,循环入点为前或后规则元素,循环出点为与入点元素相同或不同单元的活动元素。

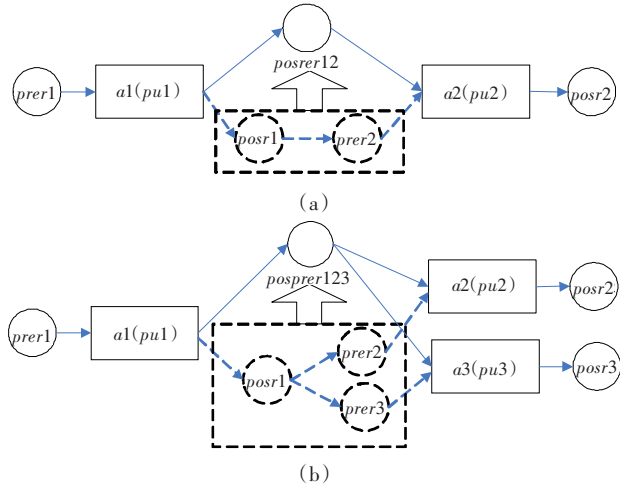


图3 基本单元的组合方式

对于其它控制结构,如K-N分支/汇聚关系等,可以参照上述策略进行组合。需要说明的是,规则元素中一般包括规则描述,因此组合后形成的规则元素中包括了组合前所有规则元素中规则描述的集合。

基本单元中的活动包括一般型和递阶型两类活动,其中一般型活动 a' 是细粒度的,不能够再分解,递阶型活动 a'' 是粗粒度的,包括多个子活动,子活动的类型与活动的类型相同。一般型活动的基本属性包括“活动标识”、“操作”和关联的“规则标识”,其中“活动标识”在模型全局范围内唯一确定了活动,“操作”表示为活动执行的外部程序名或者参与者的手工动作名,关联的“规则标识”确定了活动的前后规则元素。特别地,不同基本单元的活动间控制依赖关系实质是活动状态之间的依赖,即业务过程的执行是通过活动状态之间的变化来实现,活动的状态通过其前后规则元素表示。

定义3 (活动 a' 状态)一般型活动 a' 状态 $a'(state) = \{st | st \in prer \wedge posr\}$, 其中 $a', prer, posr \in pu$ 。

从上述定义可以看出,一般型活动的初始状态通过活动的前规则元素表示,活动变迁后的状态由后规则元素表示。 a' 和 a'' 的状态可以根据具体的业务特点来设计,如就绪、运行、取消、终止和完成等,然后由规则中元素用适当的形式表示。需要说明的是,由于递阶型活动 a'' 是多个活动的集合,因此 a'' 状态依赖于集合中各个子活动的状态,如“当子活动中任一活动取消时,则 a'' 变迁到取消状态”。 a'' 的基本属性及表示方法与 a' 类似。活动状态的变化反映了业务执行的进展过程,其具体表示方法将在下文中说明。

3.2 BPM4DBL 的规则元素概念

BPM4DBL 中规则元素的基本属性包括“规则标识”和“规则集”,其中“规则标识”在模型全局范围内定义了唯一的规则元素;“规则集”是一组规则的集合,表示该规则元素所描述的业务。从结构上看,BPM4DBL 中的规则元素可以看作业务过程

的路由节点,支持复杂过程控制结构的描述;从功能上看,规则元素可以看作业务过程的决策节点,“规则集”一方面描述活动和活动之间的非功能性约束、活动之间的变迁条件、活动参与者的分配条件,另一方面描述基本单元中的活动状态。需要说明的是,非功能性约束一般描述的是活动属性之间的约束,典型的非功能性约束包括活动的时间属性和成本属性之间的约束,例如图1中的CS1表示从“科研会计审核”执行开始到“出纳支付”执行结束之间的时间应当小于2.5个时间单位;活动之间的变迁条件描述了活动从一状态转换到另一状态时的约束,例如图1中的CO1表示从“填报销单”完成状态到“课题负责人审核”开始状态之间的一项约束;活动参与者的分配条件描述了参与者的动态宿定约束,允许在活动执行中根据约束动态宿定参与者,而不是直接设定活动参与者为具体的个人或者角色,例如“活动A的参与者是流程启动者的直接领导”。

从规则分类的角度来看,“规则集”中规则依赖于活动元素,因此用“活动规则”来统一表示BPM4DBL中的规则类型。活动规则分为断言规则和行为规则,规则的表示依赖于术语和事实,如图4所示。

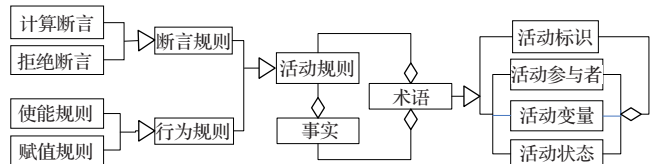


图4 BPM4DBL 中规则分类和依赖关系

定义4 (活动规则) 活动规则 $ActivityRule = \{AsserRule \cup ActRule\} \rightarrow (Term, Fact \rightarrow Term)$, 其中 $AsserRule$ 为断言规则, $ActRule$ 为行为规则, $Term$ 为术语, $Fact$ 为事实。

断言规则表示永真的规则,即BPM4DBL执行中必须满足,当规则满足时流程正常执行,并不触发其它事件(“事件”包括其它规则的触发和变量赋值等,下同);行为规则又称为触发/响应规则,即规则中相关条件满足时触发新的事件。断言规则分为计算断言和拒绝断言,其中计算断言针对活动的时间和成本等属性的数值类计算约束,规则中包含一个或多个算术操作,例如图1中的CS1约束,以及“过程的成本按照所有参与活动的总和计算”;拒绝断言面向活动状态和活动参与者等非数值类计算约束,以及活动时间和成本等属性的非数值类计算约束,例如“活动的成本不能为空”。通常,断言规则涉及到BPM4DBL中活动的时间、成本、状态和参与者相关的约束。根据触发事件类型的不同,行为规则分为使能规则和赋值规则,其中使能规则的动作触发其它的规则执行,例如“当活动参与者的角色为主任时,触发‘主任约束规则’”;赋值规则的动作触发变量的状态或取值的变化,例如“如果递阶活动A的子活动B和子活动C的状态都为‘完成’,则活动A的状态为‘完成’”。一般来说,行为规则用来表示BPM4DBL中活动之间的变迁条件、参与者动态宿定条件和递阶活动的状态约束。从以上规则分类可以看出,本文中活动规则仅仅表示具有严格的控制和约束的规则,而不包括一般意义上的建议和指导等业务规则。

术语是业务过程中的基本实体对象、对象属性以及业务过程资源和外部引用的特定表达,如图4中的活动标识、活动参

与者、活动变量和活动状态基本术语,另外可以根据业务特点定义扩充术语。活动标识、活动参与者、活动变量和活动状态称为类术语,其实例化对象称为实例术语,如“活动A的执行时间”、“活动A的参与者”、“活动A的完成状态”。事实表示术语之间的关系,即 $Fact = \{Rl(Term)\}$, Rl 为术语之间关系符号,其中基本关系包括比较、成员、属性、参与和继承关系。活动规则定义中首先需要根据业务特点定义完备的类术语和实例术语。

下面给出BNF表示的不同类型规则的句法:

计算断言 ::= “IF” “真” “THEN” 数学公式

数学公式 ::= 活动时间类术语 | 活动时间实例术语 | 活动成本类术语 | 活动成本实例术语 | 事实 “=” 活动时间类术语 | 活动时间实例术语 | 活动成本类术语 | 活动成本实例术语 | 活动变量术语 “+” “-” “*” “/” “=” 活动时间类术语 | 活动时间实例术语 | 活动成本类术语 | 活动成本实例术语 | 数值

拒绝断言 ::= “IF” 逻辑表达式 “THEN” “假”

逻辑表达式 ::= 活动时间类术语 | 活动时间实例术语 | 活动成本类术语 | 活动成本实例术语 | 活动状态类术语 | 活动状态实例术语 | 活动参与者类术语 | 活动参与者实例术语 | 活动变量术语 | 事实 “=” “!” “>” “<” “空” | 数值 | 活动状态枚举 | 参与者角色枚举 | 参与者个体枚举 [“^” “v”]

使能规则 ::= “IF” 逻辑表达式 “THEN” {规则名}

赋值规则 ::= “IF” 逻辑表达式 “真” “THEN” 赋值表达式

赋值表达式 ::= 活动时间类术语 | 活动时间实例术语 | 活动成本类术语 | 活动成本实例术语 | 活动状态类术语 | 活动状态实例术语 | 活动参与者类术语 | 活动参与者实例术语 | 活动变量术语 “=” 数值 | 活动状态枚举 | 参与者角色枚举 | 参与者个体枚举

需要说明的是,规则中的“真”和“假”分别表示逻辑永真和永假;类术语和实例术语的名称在不同的BPM4DBL模型中可以不同,只需在同一模型中保持一致。

从活动规则的BNF句法可以看出,活动状态可以通过赋值规则来表示。一般来说,活动的初始状态表示为:赋值规则 ::= “IF” “真” “THEN” 赋值表达式;活动变迁后状态表示为:赋值规则 ::= “IF” 逻辑表达式 “THEN” 赋值表达式。

3.3 BPM4DBL的可执行语言描述

在模型的实际应用中,需要定义BPM4DBL概念的实例化形式,即BPM4DBL的可执行语言。与目前得到学术界和工业界广泛支持的业务过程执行语言XPDL相比,BPM4DBL的基本单元中显式引入规则元素,并且可以通过规则元素描述易变的业务策略。到目前为止,XPDL包括两个版本,一个是2002年10月发布的1.0版本,另一是2005年10月发布的2.0版本。2.0版本中除对少部分元素或属性的修改或替换外,保持向后的兼容性。通过对XPDL2.0的扩展,给出BPM4DBL的可执行语言描述。

BPM4DBL中的AS和FS分别对应于XPDL2.0中的Activities和Transitions元素。注意到,XPDL2.0中<xsd:element name = “Application”>元素的类型包括<xsd:element name = “BusinessRule”>元素,即业务规则作为一类外部应用可以被触发,但不能被XPDL2.0中的其余元素所引用,也即不能表达BPM4DBL基本单元中活动和规则实体元素之间的约束依赖,因此,需要在XPDL2.0中引入RS表示元素,使其作为<xsd:complexType name = “ProcessType”>的子元素。同时,BPM4DBL基本单元中的一个规则元素可以包括多个规则集合,因此,需

要引入“BusinessRuleSets”、“BusinessRuleSet”和“BusinessRule”三类元素,其中“BusinessRuleSets”是“BusinessRuleSet”的集合,同时“BusinessRuleSet”则是“BusinessRule”的集合,下面仅给出“BusinessRule”的描述。

在<xsd:schema>中增加命名空间<xsd:schema xmlns:brdl = “http://www.intec.iscas.ac.cn/businessrule/RuleS1.1”>;其次,增加<xsd:complexType name = “ProcessType”>的子元素<xsd:element ref = “brdl:BusinessRule” minOccurs = “0”/>。在命名空间RuleS1.1中的BusinessRule定义如下:

```
<xsd:element name = “BusinessRule” minOccurs = “0”>
  <xsd:complexType>
    <xsd:sequence >
      <xsd:element name = “Term” type = “xsd:string” minOccurs = “0” />
      <xsd:element name = “Fact” type = “xsd:string” minOccurs = “0” >
        <xsd:complexType>
          <xsd:attribute name = “Type” use = “required”>
            <xsd:simpleType>
              <xsd:restriction base = “xsd:NMTOKEN”>
                <xsd:enumeration value = “Compare” />
                <xsd:enumeration value = “Member” />
                <xsd:enumeration value = “Attribute” />
                <xsd:enumeration value = “Part” />
                <xsd:enumeration value = “nher-it” />
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:attribute>
        </xsd:complexType>
      </xsd:element>
      <xsd:element ref = “brdl:AssertionRule” minOccurs = “0” maxOccurs = “unbounded”>
      <xsd:element ref = “brdl>ActionRule” minOccurs = “0” maxOccurs = “unbounded”>
    </xsd:sequence>
    <xsd:attribute name = “Id” type = “xsd:NMTOKEN” use = “required” />
    <xsd:attribute name = “Name” type = “xsd:string” use = “optional” />
  </xsd:complexType>
</xsd:element>
断言规则 AssertionRule 定义为:
<xsd:element name = “AssertionRule”>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name = “Expression” type = “xpdl:ExpressionType” minOccurs = “0”>
      <xsd:attribute name = “Type” use = “required”>
        <xsd:simpleType>
          <xsd:restriction base = “xsd:NMTOKEN”>
            <xsd:enumeration value = “Computation” />
```

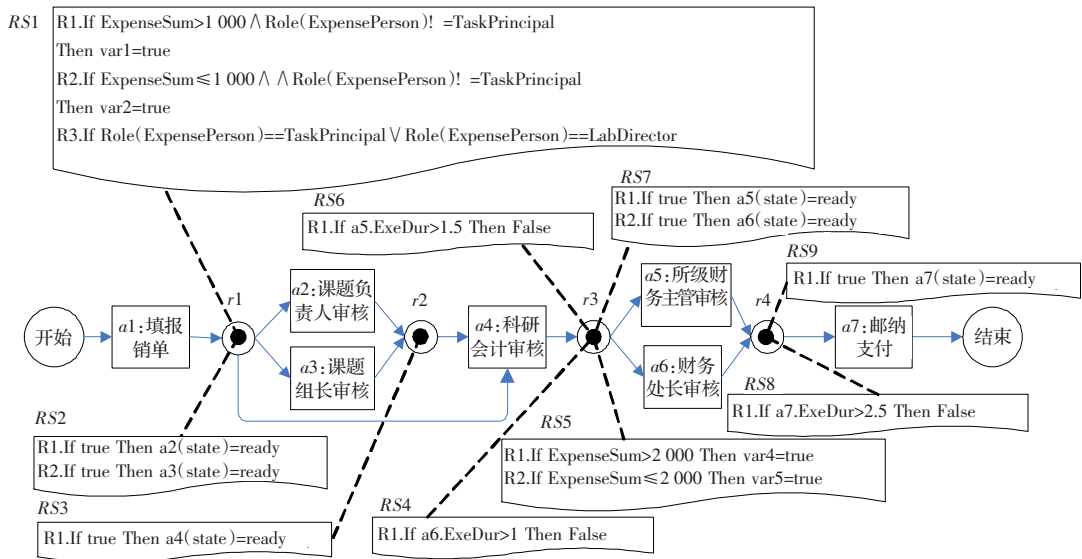


图5 BPM4DBL 表示的科研报销流程

```

<xsd:enumeration value="Decline"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
    
```

行为规则 ActionRule 与 AssertionRule 的定义相比,只是表达式的枚举类型不同,其类型分别为“Enabled”和“Evaluated”。

4 BPM4DBL 应用示例

应用 BPM4DBL 描述图 1 中的科研报销流程时,需要引入规则元素 r1、r2、r3 和 r4,如图 5 所示。r1 中包括 RS1 和 RS2 规则集,其中 RS1 中规则对应于 CO1、CO2、CO3、CO4、CO5 和 CO6,RS2 中为 a2 和 a3 的状态规则;r2 中规则集为 RS3 为 a4 的状态规则;r3 中 RS4、RS5 和 RS6 规则分别对应于 CS2、CO7 和 CO8、CS1;r4 中规则集 RS8、RS9 分别对应于 CS3 和 a7 的状态规则。

下面给出 BPM4DBL 中 r1 元素及其 RS1 中“R1”的可执行语言描述。r1 元素的描述为:

```

< brdl:BusinessRuleSets Id="r1" Name="r1">
  < brdl:BusinessRuleSet Id="RS1" Name="RS1">
    < brdl:BusinessRuleSet Id="RS2" Name="RS2">
  </ brdl:BusinessRuleSets>
“r1”中“RS1”的具体描述为:
  < brdl:BusinessRuleSet id="RS1" name="RS1">
    < brdl:Term >ExpenseSum< / brdl:Term >
    < brdl:Term >ExpensePerson< / brdl:Term >
    < brdl:Term >TaskPrincipal< / brdl:Term >
    < brdl:Term >LabDirector< / brdl:Term >
    < brdl:Term > var1< / brdl:Term >
    < brdl:Fact Type="Attribute">Role </brdl:Fact >
    < brdl:ActionRule id="R1" name="R1">
      < brdl:Expression Type ="Enabled" > If
ExpenseSum>1 000 ^ Role (ExpensePerson)! =TaskPrincipal Then
    
```

```

var1=true</brdl:Expression>
</brdl:ActionRule>
</brdl:BusinessRuleSet>
    
```

在 RS1 的语言描述中,如果“R1”中的条件为真,则术语“var1”被赋予真值“true”;在活动“填报销单”到“课题负责人审核”之间的变迁中,其变迁条件表示为“<Condition Type=“CONDITION”> var1&et;= true </Condition>”。在规则元素“r3”中包括 RS4、RS5、RS6、RS7 规则集,其中 RS5 面向活动的变迁条件,其应用方式与 RS1 类似;RS4 和 RS6 分别作为活动“所级财务主管审核”和“财务处长审核”的执行时间约束,被活动的扩展属性所引用,如“<ExtendedAttribute Name=“bRule”Value=“RS2”/>”。需要指出的是,活动扩展属性中引用规则的触发方式由 workflow 引擎的确定,不同的 workflow 引擎可以设计和选择规则的不同触发时间点。

5 结束语

随着组织机构所处内外部环境的经常变化,业务过程模型需要适应业务逻辑的动态调整,使得一定范围内易变业务逻辑的变化并不需要从整体上重构业务过程模型。本文通过对业务过程中易变业务逻辑特点的分析,定义了分离易变业务策略的业务过程模型 BPM4DBL,给出了易变业务策略的分类和具体表达式,从而一方面可以通过简洁的方式表达复杂的业务逻辑,另一方面使得业务人员在流程运行中方便地局部修改业务策略,而不需要重新生成新的过程模型。

后续工作中,将研究 BPM4DBL 的形式化表示方法,并且由此对其进行正确性、合理性和性能等分析。

参考文献:

[1] Sun Y Q, Meng X X, Liu S J, et al. Flexible workflow incorporated with RBAC[C]/LNCS 3865, 2006: 525-534.
 [2] Ly L T. Integration and verification of semantic constraints in adaptive process management systems[J]. Data & Knowledge Engineering, 2007(6): 7.