

一种模型驱动的工作流过程定义途径^{*})

赵文 袁崇义 张世琨 王立福

(北京大学信息科学技术学院 北京 100871)

摘要 传统的工作流管理系统在互操作性,可复用性,可移植性和开发效率等方面遇到了挑战,MDA 是解决上述挑战和问题的一种途径。在模型驱动的工作流管理系统中,工作流模型处于核心地位,本文首先给出了扩展的工作流元模型和基于 Petri 网的形式化工作流模型过程网。其次,运用模型驱动的途径,依据给出的工作流模型,本文提出了一种模型驱动的,用于快速构造工作流管理系统的框架。针对工作流过程模型,着重讨论了从基于 EPC 的过程 CIM 到基于过程网的过程 PIM 的转换(转换过程和转换规则),并以扩展的 Petri 网标注语言 E-PNML 规约了过程 PIM。
关键词 模型驱动的体系结构,工作流管理系统,事件驱动的过程链,过程网,过程 PIM,过程 CIM,模型转换,PNML

A Model Driven Approach to Defining Workflow Process

ZHAO Wen YUAN Chong-Yi ZHANG Shi-Kun WANG Li-Fu

(School of Electronic Engineering and Computer Science, Peking University, Beijing 100871)

Abstract Interoperability, reusability, portability and productivity are the real challenges for traditional workflow management system, MDA is an approach to meeting the challenges mentioned above. In the research area of model driven workflow management system, workflow model is the very core of the subject. As a start, extended workflow metamodel and Petri net based formal workflow model procedure net are introduced. And then, according to MDA and given workflow model, a model driven framework is proposed for rapidly building workflow management system. Finally, one of the key steps in the framework, i. e. transformation (process and rules) from EPC based process CIM to procedure net based process PIM, is discussed in detail, and E-PNML specification for process PIM is given.

Keywords MDA, Workflow management system, EPC, Procedure net, Process CIM, Process PIM, Model transformation, PNML

1 引言

伴随计算机应用的普及、网络技术的迅猛发展以及业务领域的不断拓展,信息资源异构、分布和松散耦合的特点表现得越发明显。对信息的处理方式也发生了根本的改变,由集中式向着异构分布式、协同工作发展。

工作流技术是一项管理、协调和控制组织内部或组织间相关活动的技术。从概念上讲,工作流是一类能够完全或者部分自动执行的业务过程,它根据一系列过程规则使得文档、信息或任务能够在不同的执行者之间进行传递或执行,以达到一个总体的业务目标^[1]。工作流管理系统是一个软件系统,它完成工作流的定义和管理,并按照在计算机中预先定义好的工作流逻辑推进工作流实例的执行,工作流管理系统通常提供 3 种功能:

1) 建立阶段功能:主要考虑工作流过程和相关活动的定义和建模功能,即完成业务过程的计算机化定义;2) 运行阶段控制功能:在一定运行环境下,执行工作流过程,并完成每个过程中活动的排序和调度功能;3) 运行阶段的人机交互功能:实现各种活动执行过程中用户与 IT 应用工具之间的交互。

随着各种新技术的不断涌现、工作流管理所涉及范围和领域的不断拓展以及业务需求的频繁变更,使得传统的工作流管理机制和方法遇到了挑战。首先,表现在“交互性”方面。

同一工作流过程中的不同活动间、不同过程的活动间需要进行交互。但是,通常活动所调用的应用是跨不同平台或技术的,有的甚至可能是遗产系统;其次,表现在可移植性方面。软件行业的特点之一就是新技术的不断产生和普及,所造成的结果就是旧系统的贬值。如何快捷、平滑地实现从一个平台移植到另一个平台,这是传统工作流管理系统难以做到的。

2002 年 OMG 组织提出了模型驱动的体系结构 MDA (Model Driven Architecture)^[2],通过将软件的业务逻辑和实现技术分离开来,MDA 可以较好地解决交互性、可移植性等问题,并提高软件开发的效率。

依据 MDA 的上述特性以及可实现性的增强,MDA 逐步被应用于工作流管理领域^[3~5]。基于模型驱动工作流管理技术的研究和实践成果,本文给出了一种模型驱动的工作流管理系统框架,并给出了其中关键步骤的实现机制。本文第 2 部分给出了改进的工作流元模型,包括作为核心的过程元模型、组织机构元模型、功能元模型,以及这些元模型间的关系。并给出了基于 Petri 网的形式化工作流模型“过程网”;在第 3 部分,基于基本的 MDA 框架,给出了模型驱动的工作流管理系统框架;第 4 部分详细介绍了框架的核心步骤,即从 CIM 到 PIM 的转换。

2 工作流模型

MDA 的核心是模型。简单地说,模型是物理系统的抽

^{*}) 国家自然科学基金(No. 60473058);国家重点基础研究发展计划(973)(No. 2002CB312006)。赵文 博士,主要研究领域为软件工程及工作流技术;袁崇义 教授,博士生导师,主要研究领域为并行计算、Petri 网理论及应用;张世琨 研究员,博士生导师,主要研究领域为软件工程和软件体系结构;王立福 教授,博士生导师,主要研究领域为软件工程和信息安全。

象。 workflow 模型是对业务过程的抽象。为了提高模型的正确性以及降低建模的复杂度,依据 ARIS 模型^[6]和 WfMC 的工作流过程定义元模型^[7],本文给出了集成的 workflow 模型^[7]。集成的 workflow 模型由过程模型、组织机构模型和功能模型三个模型组成,其中,过程模型是核心。

对 workflow 模型特别是 workflow 过程模型进行形式化规约是必要的。一方面,可以更准确地描述 workflow 模型;另一方面,也为 workflow 模型的验证和仿真奠定了坚实的基础。通过对 WF-net^[7]的分析,我们提出了基于 Petri 网层次化的 workflow 模型“过程网”^[8,9]。其中,在逻辑层,用 P/T 系统定义过程逻辑;在语义层,引入 C-net 描述过程语义;在执行层,通过对偶网规约引擎的执行。

2.1 工作流元模型

工作流元模型是定义 workflow 模型的语言,用于描述 workflow 模型内部的各个元素、元素之间关系及元素属性等。工作流元模型的核心是过程元模型。集成的 workflow 模型由过程模型、组织机构模型和功能模型组成。下面分别介绍三个相关的元模型及其关系。

过程元模型:主要用于描述 workflow 过程中的活动以及活动之间的关系。根据问题分离原则(Separation of Concerns),对 workflow 管理联盟制定的 workflow 过程定义元模型作了适当的修改和扩展,把汇聚、分支结构和它们的约束(AND、OR、XOR)说明从活动中提取出来,引入了一种新的元模型元素“连接符”(Connector)^[8,10],即把所有控制信息规约从活动规约中独立出来,分别封装,使得变化的影响局部化,这种分离有利于表达复杂的过程逻辑, workflow 模型的动态修改等。

组织机构元模型:基于角色用来定义组织成员、各部门的作用和任务、它们间的相互关系以及与其他模型间关系。

功能元模型:主要用于描述功能以及功能的分解层次关系,即描述过程中活动所具备或调用的功能以及所处理的相关数据。

三类元模型描述了 workflow 的不同侧面,这些侧面之间存在着一定的关联。其中,过程模型处于核心地位,它与组织机构模型和功能模型存在着紧密的关系。

过程模型通常是在一定的组织机构背景下建立的,过程模型中的每个活动应该拥有执行者,即在过程模型中所涉及的组织单元和角色的来源是组织机构模型。过程模型描述的只是活动间的逻辑关系。其中,过程模型的每个活动都会调用一定的功能。这些具体功能间的结构关系和数据关系是通过功能模型给出的。

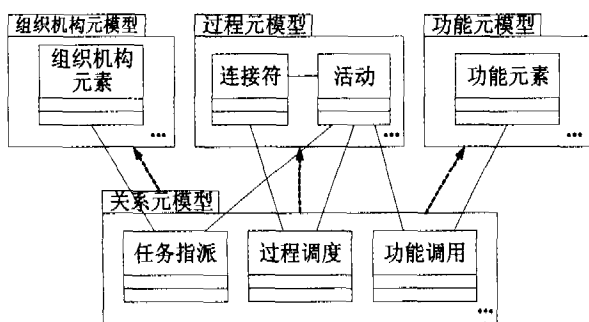


图1 工作流元模型间关系

由于每个元模型都是基于 MOF 的,所以可以建立关系元模型(见图 1)来表达各模型之间的特定关联。例如,关系

元模型中包含的“任务指派”关联了过程模型中的活动和组织机构模型中的组织机构元素。关系元模型中元素代表的操作实际上是 workflow 引擎的核心功能,因此,关系元模型实质上相当于“ workflow 执行元模型”。

2.2 基于 Petri 网 P/T 系统的层次化 workflow 模型“过程网”

在研究和分析现有形式化 workflow 模型的基础上,依据改进的工作流元模型,我们提出了一种基于 Petri 网 P/T 系统层次化的 workflow 模型“过程网”(Procedure net),分为 3 个层次,即逻辑层、语义层和执行层。其本质强调过程逻辑的建立与过程语义规约以及过程执行的分离,目的在于解决 workflow 过程模型的良好性问题,并为过程模型的分析奠定基础。下面介绍过程网的 3 个层次。

过程逻辑(process logic)描述的是过程中活动之间的逻辑关系,关注的是活动间相互依赖关系这样实例不变因子,是面向过程的。依据改进的工作流过程元模型,我们尝试用 P/T 系统来描述过程模型的控制逻辑。通过对常用 workflow 模式进行描述,可以得到规约 workflow 过程逻辑所需要的唯一网结构^[8](见图 2)。

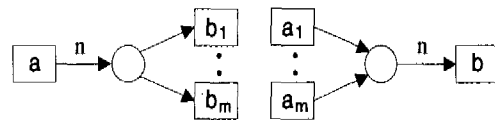


图2 表达过程逻辑的唯一网结构

定义 1 若

$$\begin{aligned} \forall t \in T: |\cdot t| \geq 1 \wedge |t \cdot| &= 1 \wedge \\ \forall p \in P: |\cdot p| \geq 1 \vee |p \cdot| &= 1 \wedge \\ F^+ \cap (F^{-1})^+ &= \phi \end{aligned}$$

则一个有限连结点 Petri 网 $N=(P, T; F)$ 被称为一个基本逻辑网。其中, $F^+ \cap (F^{-1})^+ = \phi$ 要求网中不包含有向圈,每个变迁(workflow 活动)可以有 1 到多个输入,但只能有一个输出。

对于 $p \in P$,若 $|\cdot p| > 0 \wedge |p \cdot| > 0$,则 p 被称为一个连接符。若 $|p \cdot| > 1$,则 p 是一个扇出;若 $|\cdot p| > 1$,则 p 是一个扇入。

每个 $p \in P$ 都代表一个可调加权同步距离^[8,10],它用来规约 $\cdot p$ 中的变迁和 $p \cdot$ 中的变迁是如何同步的。调整同步距离的方法是在弧 (t, p) 或 (p, t) 上赋一个权值(见图 2)。

定义 2 若对于所有连接符 $p \in P$ 以及所有 $t \in T$:

$$\begin{aligned} (p, t) \in F \Rightarrow W(p, t) &\leq |\cdot p| \wedge \\ (t, p) \in F \Rightarrow W(t, p) &\leq |p \cdot| \end{aligned}$$

则一个具有加权函数 $W: F \rightarrow \{1, 2, \dots\}$ 的基本逻辑网 $N=(P, T; F, W)$ 。

对于连接符 $p \in P$,使得 $in(p) = \sum_{t \in \cdot p} W(t, p)$ 并且 $out(p) = \sum_{t \in p \cdot} W(p, t)$ 。若 $in(p) = out(p)$,则 p 是一个 AND 连接符;若 $in(p) \neq out(p)$,则 p 是一个 OR 连接符。

定义 3 $\Sigma=(P, T; F, K, W, M_0)$ 称为 workflow 模型的过程逻辑,如果 $(P, T; F, K, W)$ 为过程网,且 $\forall p \in P: (\cdot p = i \Rightarrow M_0(p) = 1) \wedge (\cdot p \neq i \Rightarrow M_0(p) = 0)$ 。

过程网的变迁规则是在 P/T 系统的变迁规则上增加了若干限制,其中最主要的一条就是每个变迁至多能发生一次。

过程网有两类特殊库所,即开始库所($i=\phi$)和结束库所($o=\phi$)。对于每个过程模型,开始库所只有一个,结束库所可以有多个。

由 WfMC 给出的 workflow 定义可以看出,各种类型的信息在执行者之间传递并接受处理。由此,每个 workflow 过程模型都会需要电子表单,我们称之为 B-Form,其主要作用在于实例化过程,并记录实例中涉及的中间结果和最后结果。我们把网中流动的托肯(token)定义为 B-Form^[8]。

模型语义层涉及的主要内容是活动的调度规则。调度规则是指活动做出分支决定时用于调度后续活动的规则,是面向过程实例的,其中的候选分支由过程逻辑给出。具体地说,调度规则描述的是 OR 连接符等的选择细节,选择的做出可能基于活动执行者的人为决定,也可能基于 B-Form 中所承载的信息(数据)。

显然,选择细节需要进行详细的规约。由于 Petri 网本身不具备数据的读、写能力,而 C-net^[10]在 Petri 网 P/T 系统中引入了新的库所来表示数据、引入新弧分别表示读、写操作。因此,可以基于 C-net 对 workflow 过程模型的语义层进行规约。

定义 4 $CN=(P,V,T;F,R,W_r)$ 称为 C-net, 如果
 $P \cup V \cup T \neq \phi \wedge P \cap V = \phi$
 $\wedge P \cap T = \phi \wedge V \cap T = \phi$
 $\wedge F \subseteq P \times T \cup T \times P \wedge (R, W_r \subseteq V \times T)$
 $\wedge dom(F) \cup cod(F) = F \cup T$
 $\wedge dom(R \cup W_r) = V \wedge cod(R \cup W_r) \subseteq T$

其中, P 是库所集, V 是变量集, R, W_r 是读写关系, T 是变迁集, dom 和 cod 分别是二元关系的定义域和值域。

图 3 中的有向网 $(V, T; R \cup W_r)$ 给出了变量的读写关系,其中 \odot 表示一个变量,连接变量库所和变迁的弧表示读写关系。

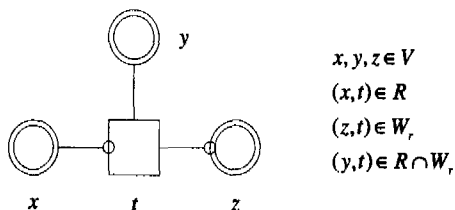


图 3 C-net 变量的读写关系

由此,一个变迁由两部分构成,即哨(Guard)和操作(Operation),见图 4。变迁在有发生权的情况下,要满足哨中的表

达式才可以发生。操作是变迁发生后所执行的功能。

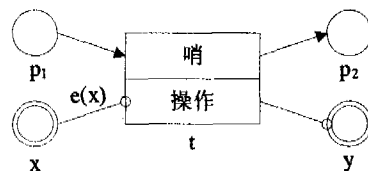


图 4 变迁的结构

模型执行层规约的是 workflow 引擎要完成的任务。例如,任务的调度、资源的指派、表单 B-Form 的处理传递以及异常处理等。过程逻辑中的每个库所正是引擎中的管理任务,而变迁恰是引擎的库所,两者有对偶性。例如,对于过程逻辑中的分支连接符 p 来说,引擎(对偶网中变迁 p)需要执行的操作是把 B-Form 制作成 n 份拷贝发送给后继中的 n 个变迁。

3 模型驱动的工作流管理系统框架

在 MDA 途径中,应用开发和维护的基础是建模,模型变得可执行。MDA 途径为我们提供了更高的可复用和抽象层次。MDA 使用下列模型:

- 与计算无关的模型 CIM: 通常被称为业务模型或领域模型,不涉及任何系统结构的细节,在其规约中使用的是领域术语;
- 与平台无关的模型 PIM: 描述了系统,但不涉及最终实现平台的知识;
- 与平台相关的模型 PSM: 描述了系统,并且包含所需的最终实现平台的知识。

CIM 是通过业务需求来定义的,本身是面向业务的。从 CIM 推导出 PIM 通常会需要人工参与。开发者会把注意力集中在开发 PIM 上,在高抽象层次描述软件系统。然后,开发或选择相应的工具来执行 PIM 的转换。这些工具是按特定的转换定义开发的。转换的结果是 PSM,这个 PSM 再被转换成代码^[11]。

基于 MDA 的基本框架和工作流管理系统的特征,本文给出了模型驱动的工作流管理系统框架,见图 5。首先,在 CIM 的构造方面,由于 ARIS 方法^[6]在企业建模、业务过程重组实践中得到了广泛的应用,所以这里采用 ARIS 方法对企业应用进行建模,从四个侧面分别得到企业的组织视图、数据视图、功能视图和控制视图。集成所有视图的综合模型是基于 EPC 的过程模型,是 ARIS 方法的核心,亦即 CIM 的核心。

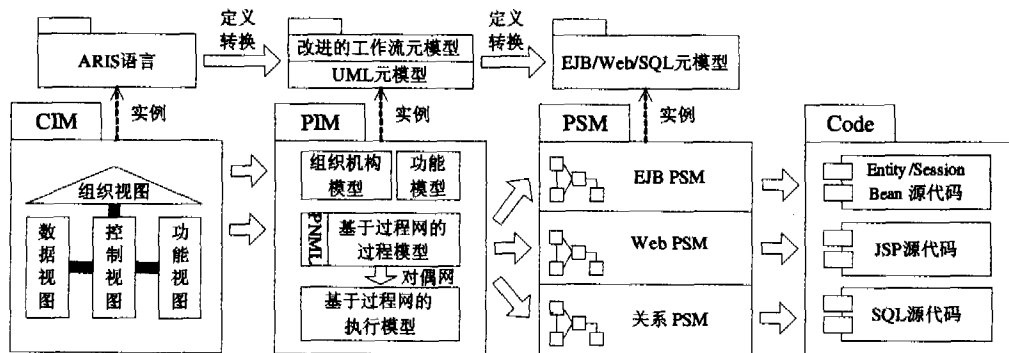


图 5 模型驱动的工作流管理系统框架

在 PIM 的构造方面,本文采用的是基于“过程网”的工作流建模和执行方法。过程网分为三个层次,逻辑层和语义层

用来对 workflow 模型进行形式化描述;执行层通过对管理逻辑和语义的定义来规约 workflow 的执行服务,即 workflow 引擎的执

行。这里把活动调用的功能狭义地定义为表单应用(即过程网中定义的 B Form)。PIM 具体表现为四个平台无关模型及其关系的集合。四个模型是指基于角色的组织机构模型、基于表单的功能模型、PNML^[12,13]形式表示的基于过程网的过程模型以及轻量级的执行服务模型。

workflow 管理系统要使用特定的技术来实现,针对 J2EE 平台,需要 3 个 PSM。首先,要有一个关系 PSM 来规约数据库,例如,组织机构模型、工作流过程静态模板和运行态实例等。EJB PSM 是基于 UML Profile for EJB 创建的,其中相关的构造型《EJBSessionBean》用来实现执行服务机制(工作流引擎)的业务逻辑和对应用的接口。相关的构造型《EJBEntityBean》用来进行数据处理,主要完成对数据操作的封装(包括对数据的增、删、改、查等),对业务逻辑部分提供数据支持。对应 Web 界面的 PSM 是用一种 UML 变体语言创建的,主要用于生成工作流活动所调用的功能(即表单)。

针对要创建的 3 个 PSM,需要定义如下 3 个从 PIM 到 PSM 的转换:

- PIM 到关系模型的转换:输入是用 UML 创建的模型,输出是 ER 图模型;
- PIM 到 EJB 模型的转换:输入是用 UML 创建的模型,输出是用 UML Profile for EJB 表示的 PSM 模型;
- PIM 到 Web 模型的转换:输入是用 UML 创建的模型,输出是用特殊 Web 界面构造型的 UML 变体表示的 PSM 模型。

从 PSM 到代码模型的转换中,要产生 3 个代码模型,分别用 SQL、Java、JSP 表示。

4 从 CIM 到 PIM 的转换

4.1 ARIS 模型到过程网的转换

CIM 通常被称为业务模型,从而可以明确地与当前的企业建模方法关联起来。目前,已经出现了一些企业建模方法和建模工具。比较具有影响的方法包括 ARIS、IDEF、CIM-OSA 等。其中,ARIS 在企业建模、业务过程重组实践中得到了广泛应用。在 ARIS 体系结构中共包含五个视图,主要的四个是功能、数据、组织和控制视图。集成所有视图的综合模型是基于 EPC 的过程模型(实例见图 6),是 ARIS 方法的核心。

作为一种建模概念,EPC 被用来表达业务过程中的事件和逻辑依赖关系。EPC 包含三种元素,即作为主动元素的功能(Function),作为被动元素的事件(Event)和描述逻辑依赖关系的连接符(Connector)。其中,连接符有三种,AND,OR 和 XOR,可以是分支结构(Split)或汇聚结构(Join)。这些元素通过作为控制流的弧进行连接。图 6 给出的是基于 EPC 的“学生请假”过程模型实例。

对于从 CIM 到 PIM 的转换,输入是 ARIS 的功能视图,数据视图,组织视图和过程模型,输出则是基于改进工作流元模型的组织机构模型、功能模型和过程模型。其中,采用扩展的 Petri 网标记语言 E-PNML^[14]来描述基于过程网的过程模型。这里主要给出从基于 EPC 的 ARIS 过程模型到基于过程网的过程模型间的转换规则和转换过程。转换包含三个步骤,即基本映射、组装衔接和特殊处理。下面分别介绍这三个步骤。

第一步:基本映射

转换主要集中在“功能到变迁(活动)”,“逻辑关系类型到连接符库所”以及“事件到库所”的映射。这里,把普通功能映射为变迁中的操作,但把与事件中条件相关的功能映射为变迁中的哨和操作。把 EPC 中事件的逻辑关系类型映射为过程网中定义为连接符的库所。把 EPC 中的开始事件、结束事件和普通事件(这里特指不含条件的事件)映射为过程网的开始库所、结束库所和普通库所。具体的转换规则见表 1。注意,在 CIM 到 PIM 的映射中,有些工作是需要人工完成的,例如,需要把条件从事件中抽取出来等。

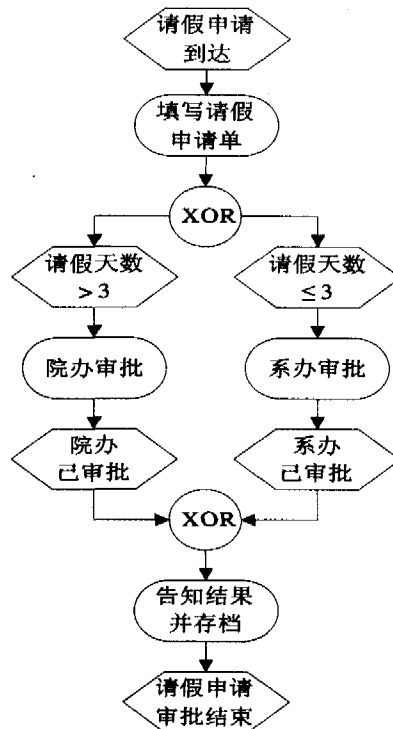


图 6 基于 EPC 的“学生请假”过程模型实例

表 1 基于 EPC 的过程模型到过程网间的转换规则

EPC 元素	过程网元素
逻辑关系类型 	连接符
功能和事件中的条件 	变迁
事件 	库所
开始事件 	开始库所
结束事件 	结束库所
普通事件 	普通库所

其中,从逻辑关系类型到连接符库所的基本映射规则如下:

- 如果 $m=n$ 且 $m, n > 1$,则表示“与”(AND)关系;
- 如果 $m \neq n$,则表示“或”(OR)关系;
- 如果 $n=1$,且 $m > 1$,则表示“异或”(XOR)关系。

第二步:组装衔接

经过第一步的映射后,形成了各自独立 Petri 网元素的集合。本步骤就是要依据规则,形成元素间的关系,从而形成初

步的过程网模型。具体的规则如下：

组装规则 1: 如果两个相邻的元素, 一个是普通库所, 一个是连接符库所, 则把普通库所“溶”入到连接符库所中, 实例见图 7。

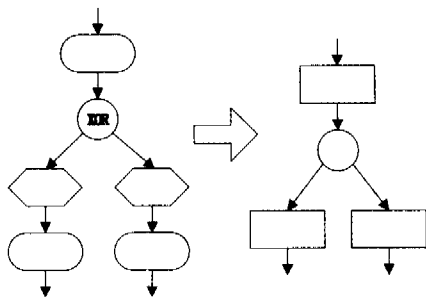


图 7 组装规则 1 实例

组装规则 2: 如果两个相邻的元素都是连接符库所, 则需要在两者之间增加相应的辅助变迁, 以保持 Petri 网的性质, 实例见图 8。

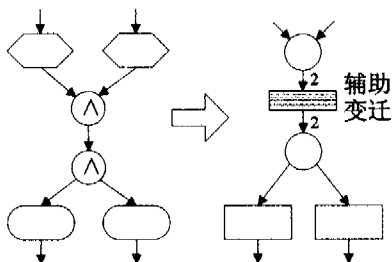


图 8 组装规则 2 实例

组装规则 3: 如果两相邻元素, 一个是库所, 一个是变迁, 则合并两者的输出连接弧和输入连接弧, 形成新连接弧连接库所和变迁。

第三步: 特殊处理

我们可以从过程网的定义中看出, 它只允许有唯一的开始库所(逻辑上的唯一入口), 但允许有多个结束库所(代表不同的处理结果); 而 EPC 并没有限制开始事件和结束事件的数量, 也就是说可以有多个开始事件和结束事件。如果基于 EPC 的过程模型中存在多个开始事件, 则需要为经过第一步和第二步“基本映射”和“组装衔接”得到的网结构, 增加唯一的“真”开始库所, 用辅助变迁和 XOR 分支连接符来替代各个“伪”开始库所。从而, 形成一个完整的过程网。具体实例见图 9。

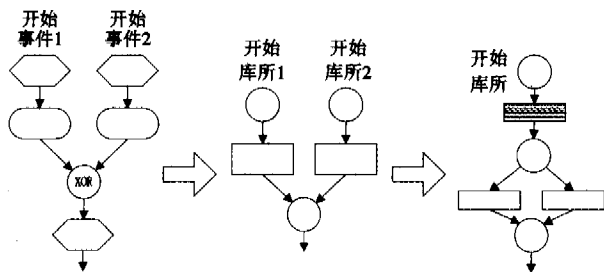


图 9 添加唯一的开始库所

基于 EPC 建立的“学生请假”过程模型实例经过转换, 便得到了图 10 中所示基于过程网的过程模型 Σ 。针对 Σ , 还需要生成相应的 E-PNML 文件, 才可以部署到 workflow 引擎上执行。图 10 中 Σ' 给出的是 workflow 引擎的执行逻辑^[8,9]。

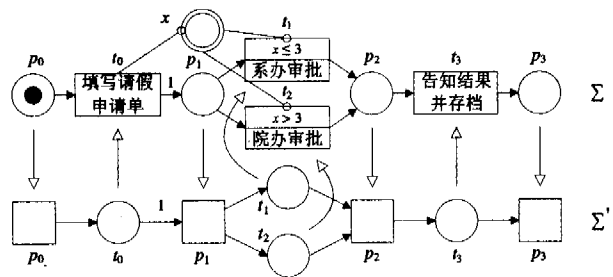


图 10 基于过程网的“学生请假”实例及执行逻辑

4.2 生成 PNML 文件

WfMC 提出的参考模型中, 定义了五类接口, 其中接口 1 的主要功能是过程定义的导入与导出, 从而实现了定义和执行两个阶段的分离。为了实现这一接口特性, 需要定义一种 workflow 过程定义语言。由于提出的形式化 workflow 模型过程网是基于 Petri 网的, 所以这里采用 Petri 网标注语言 PNML 来规约 workflow 过程模型。因为过程网引入了 C-net 来描述 workflow 语义, 而 C-net 在 P/T 系统基础上有引入了变量库所和读写关系, 所以我们对 PNML 核心元模型进行了扩展, 得到了新的元模型, 见图 11。

对于扩展的 PNML(即 E-PNML)来说, 标签(Label)是一个比较重要的概念。每个扩展的 PNML 核心元模型元素都有标签与之相对应。除了要为库所、变迁、弧和变量等元素定义本身的专有属性外, 还需定义标识和名称等一些基本属性。

在元模型扩展和标签定义的基础上, 我们给出了过程网的类型定义 E-PNTD, 即过程定义语言 E-PNML 的语法约束。E-PNTD 是特定于过程网的合法标签以及合法标签的组合, 确定了过程网的合法文件格式。

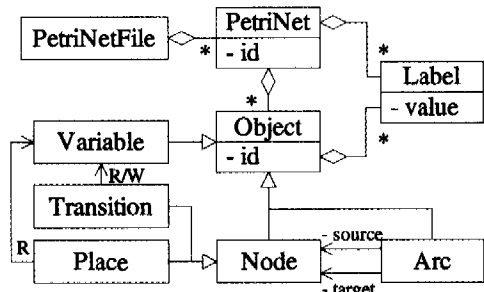


图 11 扩展的 PNML 核心元模型

下面, 给出的是“学生请假”过程实例的 E-PNML 描述。

```

<!--“请假”过程的 E-PNML 描述-->
<Pnml>
  <Name>学生请假</Name>
  <StartPlaceId>p0</StartPlaceId>
  <Places>
    <!--库所 p0, 省略了库所 p1, p2, p3-->
    <Place id=p0>
      <Name>Start</Name>
      <Ptmarking>1</Ptmarking>
      <Capacity>1</Capacity>
    </Place>
    .....
  </Places>
  <Transitions>
    <!--变迁 t2, 省略了变迁 t0, t1, t3-->
    <Transition id=t2>
      <Name>院办审批</Name>
      <Guard/>(x, GT, 3)</Guard>
      <Body>
        <Type>4</Type><AppId>2</AppId>
      </Body>
      <Participant>
        <Type>1</Type><Pid>3</Pid>
        <AssignMethod>0</AssignMethod>
      </Participant>
    </Transition>
  </Transitions>
</Pnml>
  
```

```

    </Participant>
  </Transition>
  .....
</Transitions>
</Arcs>
<! --连接 t0 和 p1 的弧, 省略了其它连接弧-->
<Arc id=a1>
  <SourceId>t0</SourceId>
  <TargetId>p1</TargetId>
  <Weight>1</Weight>
</Arc>
.....
</Arcs>
</Variables>
<! --变量 x-->
<Variable id=v1>
  <Name>x</Name><Type>int</Type>
  <Scope>global</Scope>
</Variable>
</Variables>
.....
</Pnml>

```

结束语 模型驱动的体系结构(MDA)是OMG提出的新方法学,是一种新的系统开发途径,强调整个系统开发过程由对软件系统的建模行为所驱动。运用MDA,从 workflow管理的特点出发,本文给出了模型驱动的工作流管理系统框架。基于该框架,我们开发了一套集成的可视化建模工具 WorkflowStudio,主要用来建立平台无关的模型,例如,过程模型、组织模型和功能模型(表单)等。这些模型可以部署到 workflow引擎上执行。通过该框架,可以初步生成基于J2EE平台的工作流引擎。正如OMG体系结构工作组的Frankel所说“MDA的潜力要得到充分的发挥还需要若干年的艰苦工作”,MDA和工作流技术本身并不成熟,这也体现在我们的理论研究和实践中。因此,在进一步的研究工作中,需要着重解决几个方面的问题,主要包括完善形式化工作流模型过程网,完善从CIM到PIM,从PIM到PSM以及从PSM到代码的转换规则定义等。

致谢 在此,对北京大学信息科学技术学院杨芙清院士举办,袁崇义教授和王立福教授主持的讨论班上的同学和老师表示感谢。

参 考 文 献

- Hollingsworth D. The Workflow Reference Model. Document Number: TC00-1003, Document Status - Issue 1.1. Workflow Management Coalition, 19-Jan-1995
- Mukerji J, Miller J. MDA Guide Version 1.0.1. Document Number: OMG/2003-06-01. Object Management Group, 12-Jun-2003
- Hur Wonchang, Jung Jae-yoon, Kim Hoontae, et al. Model-Driven Approach to Workflow Execution. International Conference on Business Process Management (BPM 2004). In: Desel J, Pernici B, Weske M. Eds. LNCS 3080, 2004. 261~273
- Breton E, Bezivin J. Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International, 8-12 Oct. 2001. 225~230
- 林嵩,赵建华,李宜东,等.从EDOC模型到J2EE程序:一个MDA工具的实现.计算机学报,2004,31(9):106~109
- Scheer A-W. ARIS - Business Process Modeling. 3rd Edition. Springer Verlag, April 2000
- van der Aalst W M P. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 1998, 8(1):21~66
- 赵文.工作流建模及模型验证技术研究:[学位论文].北京:北京大学,2004
- 袁崇义. Petri网原理与应用.北京:电子工业出版社,2005
- 赵文,胡文蕙,张世琨,等.工作流元模型的研究与应用.软件学报,2003,30(6):1052~1059
- Kleppe A, Warmer J, East W. MDA Explained: The Practice and Promise of The Model Driven Architecture. 1st Edition. Addison-Wesley Professional, April 25, 2003
- Jungel M, Kindler E, Weber M. The Petri Net Markup Language. Petri Net Newsletter, 2000, 59:24~29
- Billington J, et al. The Petri Net Markup Language: Concepts, Technology, and Tools. In: van der Aalst W, Best E, eds. Proc. Int'l Conf. Applications and Theory of Petri Nets 2003. LNCS, 2679. Springer, 2003. 483~505
- 迟美娜.基于JB-PNML的青鸟过程部署工具的设计与实现:[学位论文].北京:北京大学,2005

(上接第9页)

- Foster I, Kesselman C, Nick J M, et al. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 2002
- Introduction to Grid Computing with Globus[EB/OL]. <http://www.ibm.com/redbooks>, 2003
- The Globus Project[EB/OL]. <http://www.globus.org/>, 2004
- 徐志伟,李伟.织女星网格的体系结构研究.计算机研究与发展,2002,39(8)
- HUANG Li-Can+, WU Zhao-Hui, PAN Yun-He. A Grid Architecture for Scalable e-Science and Its Prototype. Journal of Software, 2005, 16(4):577~586
- Allen G, Davis G K, Dolkas K N, et al. Enabling applications on the grid: A Gridlab overview. International Journal of High Performance Computing Applications, 2003, 17(4):449~466
- Buyya R, Abramson D, Giddy J. A Case for Economy Grid Architecture for Service-Oriented Grid Computing. In: Proceedings of the International Parallel and Distributed Processing Symposium: 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), April 23, 2001, San Francisco, California, USA: IEEE CS Press, USA, 2001
- Ardaiz O, Sanjeevan K, Sanguera R. A 3-tier Grid Architecture and Interactive Applications Framework for Community Grids. ICCS, 2004. 67~74
- Green Grid Technical Overview. <http://grid.dartmouth.edu/docs/architecture.php>

- docs/architecture.php
- Zheng Xiao-lin, Chen De-ren, Lu Lin. Research of architecture for grid manufacturing. Proceedings of the Ninth International Conference, Computer Supported Cooperative Work in Design, 2005, 1:345~349
- Czajkowski K, Fitzgerald S, Foster I, et al. Grid information services for distributed resource sharing. The 10th IEEE Int'l Symp on High2Performance Distributed Computing, San Francis2co, California, 2001
- DataGrid Information and Monitoring Services Architecture: Design, Requirements and Evaluation Criteria: [Technical Report]. DataGrid
- Tierney B, Aydt R, Gunter D, et al. A Grid Monitoring Architecture. The Global Grid Forum GWD-GP-16-2
- Welch V, Foster I, Kesselman C, et al. X.509 proxy certificates for dynamic delegation. 3rd Annual PKI R&D Workshop, Apr 2004
- Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications, 2001, 15: 200~222
- Foster I, Kesselman C, Lee C. A Distributed Resource Management Architecture that Supports Advance Reservation and Co-Allocation. In: Proceedings of the International-Workshop on QoS, 1999, 27~36
- Leff A, Rayfield J T, Dias D M. Service-Level Agreements and Commercial Grids. IEEE Internet Computing, 2003, 7(4):44~50