

基于着色 workflow 网的工作流模型的设计与实现

邬家炜 陈志涛

(华南师范大学计算机学院 广州 510631)

摘要 本文在扩展 Petri 网的基础上,提出了一个新的工作流模型——基于着色 workflow 网的工作流模型,并对其进行了详细设计。在此模型的基础上,本文实现了一个可视化的工作流建模工具,设计人员通过该工具建立业务过程,并根据模型的定义保存业务过程定义。

关键词 工作流,工作流模型, Petri 网,着色 workflow 网,工作流建模工具

Design & Implementation of Workflow Model Based on Colored Workflow Net

WU Jia-Wei CHEN Zhi-Tao

(Computer School of South China Normal University, Guangzhou 510631)

Abstract Based on Petri net, this paper presents a new workflow model, which is colored workflow net, and puts forward a detailed model design. In addition, this paper realizes a visual workflow modeling tool, by which the designer can create the process flow diagram and save the process definition according to the model definition.

Keywords Workflow, Workflow model, Petri net, Colored WF-net, Workflow modeling tool

1 引言

随着信息技术和管理理论的发展以及计算机和网络的广泛应用,现代企业的信息系统越来越表现出自治、异构、分布的特点,相应的企业信息资源也分布在异构的计算机环境中,信息源之间的连接表现出松散耦合的特点^[1],使得企业的业务过程变得越来越复杂。在这种情况下,工作流管理技术逐渐成为企业过程建模和过程管理的核心技术,工作流管理系统成为企业在复杂信息环境下实现业务流程自动执行的必要工具。

工作流管理系统的最重要的功能之一就是工作流建模。一个好的工作流模型是实现一个具有高度的灵活性、可靠性、可用性、可伸缩性和互操作性的工作流管理系统的关键和基础。目前各种各样的工作流建模技术,如基于活动网络的建模方法^[2]、使用图形符号语言^[3]、基于对话模型的建模方法^[4]、基于事务模型的建模方法,如嵌套事务模型^[5]、Sagas^[6]、基于 ECAA 规则的方法^[7]、基于工作流定义语言的方法,如 WfMC 提出的 XPD (XML Process Definition Language)^[8]、BEA、IBM、Microsoft 等提出的 BPEL4WS (Business Process Execution Language for Web Services)^[9] 等等,这些方法大多数都不能以形式化的方式描述工作流业务流程,对业务过程没有函数化,功能化,也不清楚工作流执行的状态,而且几乎没有提供对工作流概念模型的分析。

Petri 网在 1962 年由德国学者 C. A. Petri 作为一种过程建模和分析的工具提出,它是一种图形化描述过程的强有力工具。Petri 网作为离散事件系统建模技术具有形式化的数学定义、提供了统一的图形表示方法来描述系统的各个特性、基于状态而非基于事件以及拥有比较成熟的分析方法等优点^[10],对业务过程的复杂性,如不确定性、并发性和资源共享等问题可以较好地解决,也为工作流的评估分析、性能分析、死锁分析等提供了理论上的依据。

2 基于着色 workflow 网的工作流模型

工作流建模需要指出哪些任务需要执行、按什么顺序执行;执行的顺序决定了在模型中需要指明任务间的关系——条件。用 Petri 网描述工作流模型时,用变迁节点表示工作流任务,库所节点表示条件,弧表示工作流的逻辑关联形式,标记表示实例,而标记的分布表示工作流实例的执行状态。

定义 1 三元组 $N = (P, T; F)$ 称作 Petri 网当且仅当^[11]:

$$(1) P \cup T \neq \emptyset, P \cap T = \emptyset;$$

$$(2) F \subseteq P \times T \cup T \times P ("X" 为笛卡尔乘积);$$

$$(3) \text{dom}(F) \cup \text{cod}(F) = P \cup T. \text{ 其中,}$$

$$\text{dom}(F) = \{x \mid \exists y : (x, y) \in F\},$$

$$\text{cod}(F) = \{y \mid \exists x : (x, y) \in F\}, x, y \in P \cup T$$

它们分别为 F 的定义域和值域。

这里, P 表示库所节点的集合, T 表示变迁节点的集合, F 为库所节点与变迁节点间的有向弧集合(流关系)。

定义 2 Petri 网 $PN = (P, T; F)$ 称作工作流网^[12], 当且仅当:

(1) PN 有两个特殊的库所: i 和 o 。其中, 库所 i 是一个输入库所, 满足 $i = \emptyset$; 库所 o 是一个输出库所, 满足 $o^* = \emptyset$;

(2) 如果在 PN 中加入一个新的变迁 t^* , 使 t^* 连接 o 与 i , 即 $t^* = \{o\}, t^{* *} = \{i\}$, 这时所得的 PN 是强连接的。

工作流模型在运行过程中,需要同时运行多个实例,可能出现两个实例同时开始、同时运行两个并行活动的情况,工作流需要同时处理多个实例,每一个实例在该模型中就存在一个或多个标记,如图 1 所示。但传统的 Petri 网和工作流网都不能区分不同的标记,即多个在同一库所中的标记无法通过定义进行区分,这通常是无法接受的。一种解决方法是通过扩展 Petri 网,给每个标记都提供一个值,通过这个值就能对标记进行区分。通过给标记赋值,就好比给它们涂上了不同

的颜色。这种扩展的 Petri 网称之为着色 Petri 网 (Colored Petri Net, CPN)。如果在工作流网的基础上加入颜色扩展, 这种工作流网定义为着色工作流网 (Colored WF-net)。

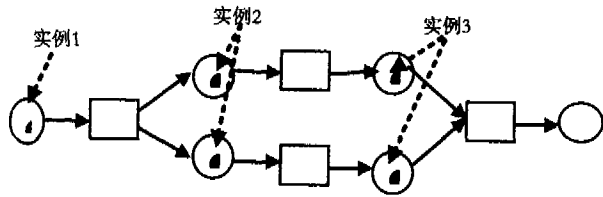


图1 使用一个或多个标记来说明实例

定义3 着色 Petri 网 (CPN)^[13]: 着色 Petri 网是多元组 $(\Sigma, P, T, A, C, G, E, I)$, 满足:

- (1) Σ 称之为 color sets, 是一个由非空的颜色集构成的有限集合, 它的每个元素 (即一个颜色集) 代表了一种类型;
- (2) P 是一个由库所构成的有限集;
- (3) T 是一个由变迁构成的有限集;
- (4) A 是一个由连接库所到变迁或变迁到库所的有向弧构成的有限集, 它满足这样的特性: $P \cap T = P \cap A = T \cap A = \phi$;
- (5) C 称之为颜色函数, 它是 P 到 Σ 的映射;
- (6) G 称之为守卫函数, 它将变迁 $t (t \in T)$ 映射成这样一个表达式: 表达式中涉及的所有的变量的类型都必须是属于 Σ 的, 而且该表达式的值是布尔值;
- (7) E 称之为弧表达式函数, 它将弧 $a (a \in A)$ 映射成这样的表达式: 表达式中涉及的所有的变量的类型都必须是属于 Σ 的, 而且该表达式的值是一个多重集, 这个集合的元素属于与弧 a 相连的库所的颜色集 $C(p(a))$;
- (8) I 称之为初始化函数, 它将库所 $p (p \in P)$ 映射成这样一个表达式: 该表达式是一个多重集, 这个集合的元素属于库所 p 的颜色集 $C(p)$ 。

定义4 着色工作流网 (Colored WF-net): 一个 CPN = $(\Sigma, P, T, A, C, G, E, I)$ 称作着色工作流网, 当且仅当:

- (1) 存在一个输入库所 $i \in P$, 使得 $i = \phi$; 存在一个输出库所 $o \in P$, 使得 $o = \phi$;
- (2) 每一个节点 $x \in P \cup T$ 都位于从 i 到 o 的一条路径上。

对于一个复杂的或较大规模的业务过程, 其相应的 Petri 网描述也较为复杂, Petri 网中的节点数量过多, 这使得对业务过程的分析较为困难。在此, 引入子网 (子过程) 的概念, 即构造层次化的 Petri 网。我们可以把一个过程, 特别是复杂的过程, 表示为由库所、变迁、弧和子过程构成的过程, 通过 (子) 过程的应用, 我们可以自顶而下或自底而上地构造 Petri 网。自顶而下的方法, 从最高层次开始, 过程不断地被分解为子过程, 直到最低层只包含变迁和库所。反复的分解得到层次化的描述。自底而上的方法恰好相反。它从最低层开始, 详细地描述基本组件, 这些元素 (子过程) 被组合成更大的过程, 不断组合, 最终得到过程的全部描述。

3 工作流过程模型设计

过程模型的结构采用 Petri 网的描述方法, 一个工作流过程可以看成是一个由两类节点——库所 (用圆圈表示) 和变迁 (用矩形框表示), 以及库所与变迁之间连接的有向弧 (用有向箭头表示) 组成的有向图。过程模型定义了过程在工作流管理系统执行中所需要的信息, 包括过程的描述信息 (如 ID, 版本、作者、创建日期等)、构成过程的活动、可能被调用的应用程序及相关数据和参数、与工作流相关的数据以及数据的类型等信息。过程模型采用 XML 作为过程模型设计结果的存储格式, 也称之为工作流文件, 可以实现过程定义在不同系统进行数据转换和数据传输。下面对过程模型进行详细设计, 如图4所示。

(1) 工作流过程组 (Workflow Processes): 在一个工作流过程组中可以定义多个过程。一个工作流过程组对应于一个 XML 文档。

(2) 工作流过程 (Workflow Process) 定义: 过程定义实体为过程其他实体提供上下文信息。过程定义包含了一个过程头 (Process Header) 信息, 其定义了过程的作者、创建日期、版本等信息。过程定义实体是一个过程容器, 为过程管理 (过程中创建日期、作者等) 或过程执行提供相关数据。

(3) 工作流过程活动 (Activity): 工作流过程活动对应于 Petri 网的变迁 (注意符号“活动”的重用, 这里的活动不表示变迁的实施)。活动有就绪和执行两种状态, 就绪的活动必须有相应的触发机制才能从就绪状态变成执行状态。

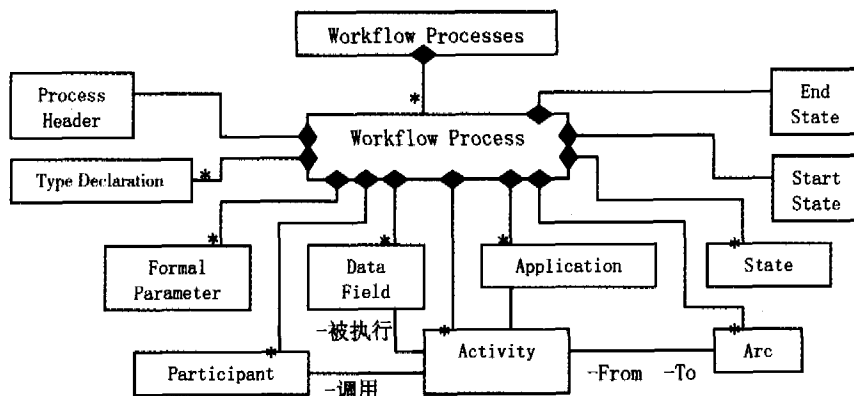


图4 过程模型实体

(4) 状态 (State) 信息: 对应于 Petri 网的状态信息, 即库所中标记的分布, 是工作流过程的条件信息。状态信息中的每个状态 (库所) 节点包含零个或若干个标记 (token), 每个标记表示一个工作流过程实例。活动只有满足可实施的条件才能实施。也就是, 只有每个输入库所都至少有一个同颜色 (ID

值) 的标记, 活动才变成就绪的。
 (5) 开始状态 (Start State) 节点: 对应于工作流网的输入库所 i , 是工作流过程的开始节点。进入开始状态节点的标记代表着一个工作流过程实例的开始。
 (6) 结束状态 (End State) 节点: 对应于工作流网的输出

库所,是工作流过程的结束节点。进入结束状态节点的标记代表着一个工作流过程实例的结束。

(7) 有向弧(Arc)信息:对应于 Petri 网的有向弧,是工作流过程的转移信息。Petri 网的节点之间通过有向弧连接起来,包括两种弧:从库所(状态节点)到变迁(活动节点)的有向弧和从变迁(活动节点)到库所(状态节点)的有向弧。每条弧都有两个基本的属性:源节点、目标节点。一般情况下,有向弧只是说明节点间的转移顺序,至于有向弧的转出信息和转入信息则在其源节点和目的节点中进行说明。

(8) 工作流参与者(Participant)声明:描述了在过程定义中各个活动的执行者所需要的资源,资源可以是人或物。特定的资源可以分配给特殊的活动,并且作为活动的一个属性来进行说明。

(9) 工作流应用程序(Application)声明:描述了工作流服务需要调用的 IT 应用程序或者接口。工作流服务调用这些应用程序或者接口来支持对相关活动进行处理,或者是自动执行这些活动。这些应用程序以及接口在相关活动的应用程序属性中定义。被调用的应用程序可以是工业工具软件、特殊部门(企业)服务,或者在工作流管理系统中实现的本地过程(方法)。工作流外部应用程序声明反映了工作流机与外部应用程序之间的接口,包括需要传递的任何参数(Formal Parameter)。

(10) 工作流相关数据(Workflow Relevant Data):是指那些在过程运行期间,每个过程实例都需要用到或者创建的数据(Data Field)。活动或者过程在执行时必须得到这些数据,这些数据也可以用来在活动间传递持久信息或者中间结果。活动、被调用的程序都可能涉及到工作流相关数据。

(11) 数据类型(Data Type):过程模型中给出了一些标准的数据类型(integer, float, string, date/time 等),这些数据类型与工作流相关数据、工作流应用程序参数相关。需要这些数据类型来构造工作流相关数据和应用程序参数的定义和表达式。

(12) 类型声明(Type Declaration):开发者在特殊情况

下,可能需要使用复杂的数据类型。过程模型通过类型声明定义开发者自己的数据类型。这些类型声明中定义的数据类型可以被工作流相关数据和工作流应用程序引用。

4 工作流建模工具的实现

工作流建模工具开发于 Windows 环境下,使用 Java 1.4.2 作为程序设计语言,使用 IBM 开源项目 Eclipse 3.0.1 作为开发工具。本工作流建模工具的设计目标是提供一个功能实用、操作简单、界面友好、结构清晰的工作流过程定义工具,系统主界面包括:

(1) 菜单栏:提供建模工具所需命令操作,包括文件操作(新建、打开、关闭、保存工作流文件等操作)、编辑(撤销、复制、剪切、粘贴、删除等操作)、绘图(选择对象、绘制活动、绘制状态、绘制连接线等操作)、查看(显示/隐藏工具栏、显示/隐藏绘图工具栏等操作)、工具(工作流定义分析)、帮助五类子菜单。

(2) 工具栏:包括常规工具栏和绘图工具栏。常规工具栏提供了如文件的新建、打开、保存等命令;绘图工具栏提供了以 Petri 网图形结构为基础的图形元素,方便用户绘图操作。

(3) 工作区:该区域以树型结构的方式显示各种带有层次结构的非可视对象内容,并提供新建、删除、修改等操作。支持 XML 文件数据的显示。

(4) 绘图区:工作流业务过程绘制的区域,每个业务过程都在此区域内以 Petri Net 图形结构完整地表示出来,让用户一目了然。

(5) 信息提示区:该区域主要提示显示相关操作辅助信息,协助用户对操作理解。

下面给出系统主界面的主要类图,如图 6 所示,工作流文件,即过程模型定义结果是建模工具的核心部分,过程模型中的所有对象类都从一个基类继承,该基类实现一些它们所有共有的成员变量和成员方法,派生类根据自身的特殊情况,通过重载和增加方法来实现自身的特有属性。

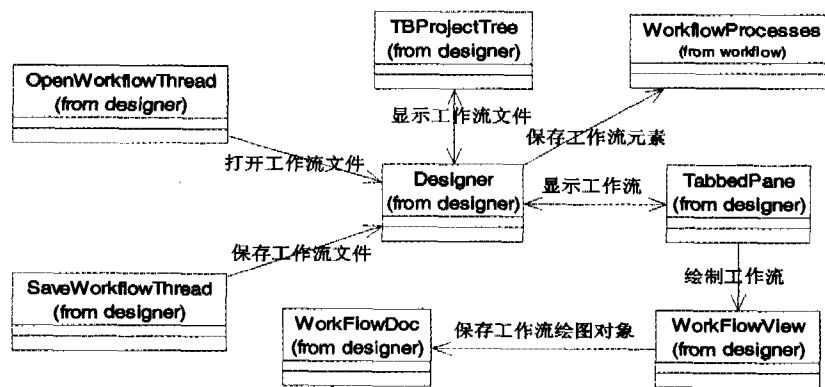


图 6 系统主界面的主要类图

总结 本文提出了一种扩展的 Petri 网——着色工作流网,并对其进行了层次化的扩展,使 workflow 设计人员更容易设计出易懂、易维护的业务过程模型,有利于不同企业和系统的流程复用。接着本文在基于着色工作流网的工作流模型的基础上,对工作流模型进行了详细设计,实现了一个基于着色工作流网的工作流建模工具。着重考虑了工作流模型的设计,但是对一个企业,特别是规模较大的企业来说,企业业务过程还涉及到其他方面,如企业资源、企业组织人员等,如何把工作流模型与其他模型(功能模型、信息模型、资源模型、组织模

型等)进行有效集成,以支持企业更复杂的过程模型,是作者今后研究的重点。

参考文献

- 1 范玉顺主编. 工作流管理技术基础. 清华大学出版社, 施普林格出版社, 2001
- 2 金字, 周良, 丁秋林. 基于活动网络图的工作流建模工具的设计与实现. 航空计算技术, 2003, 33(1)
- 3 Sadiq W, Orłowska M. Analyzing process models using graph reduction techniques. Information Systems, 2000, 25(2): 117~134
- 4 Medina-Mora R, Winograd T, Flores R, Flores F. The Action Workflow Approach to Workflow Management Technology. In: Proc. of CSCW'92, Toronto, Canada, 1992. 281~288

5 Alonso G, Agrawal D, Abbadi A E, Kamath M, et al. Advanced Transaction Models in Workflow Contexts. In: Proc. of the Twelfth Intl Conf. on Data Engineering, 1996

6 Garcia-Molina H, Salem K. Sagas. In: Proc. 1987 SIGMOD Intl. Conf. on Management of Data, May 1987. 249~259

7 Endl R, Knolmayer G, Pfahrer M. Modeling Processes and Workflows by Business Rules. In: Proc. of the 1st European Workshop on Workflow and Process Management, Zurich, Switzerland, 1998. 47~56

8 WfMC. Workflow Management Coalition. Interface 1: Workflow Process Definition Interface-XML Process Definition Language. WfMC TC-1025, 2002

9 服务的商业流程执行语言. <http://dev2dev.bea.com/Web-services/BPEL4WS.jsp>

10 李红臣, 史美林. Petri 网在业务过程建模中的应用. 小型微型计算机系统, 2001, 22(1)

11 袁崇义. Petri 网原理与应用. 北京: 电子工业出版社, 2005

12 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

13 Jensen K. Coloured Petri Nets; Basic Concepts, Analysis Methods and Practical Use, Vol. 1: Basic Concepts, EATCS Monographs on Theoretical Computer Science, Springer, Berlin, 1992

(上接第 131 页)

矩阵 A 的元素值在 16 比特范围内, 我们选取 $P=17$ 。反量化过程可以表示为:

$$Y_r(i, j) = [Y_q(i, j)B(Q_M, i, j) \gg (8 - Q_E)]$$

在量化和反量化过程中, 矩阵 A 和 B 的元素可以表示为: $A(Q_M, i, j) = M(Q_M, r)$, $B(Q_M, i, j) = S(Q_M, t)$ 。其中, 当 $(i, j) = \{(0, 0), (0, 2), (2, 0), (2, 2)\}$ 时, $r=0, t=0$; 当 $(i, j) = \{(1, 1), (1, 3), (3, 1), (3, 3)\}$ 时, $r=1, t=1$; 当 $(i, j) = \{(0, 1), (0, 3), (2, 1), (2, 3)\}$ 时, $r=2, t=2$; 其他情况时, $r=3, t=2$ 。矩阵 M 和 S 的元素分别表示为:

$$M = \begin{bmatrix} 52429 & 7231 & 27537 & 13768 \\ 47663 & 6574 & 25304 & 12517 \\ 40330 & 5563 & 21182 & 10591 \\ 37449 & 5165 & 19669 & 9835 \\ 32768 & 4520 & 17211 & 8605 \\ 29127 & 4018 & 15298 & 7649 \end{bmatrix}$$

$$S = \begin{bmatrix} 5120 & 1412 & 2689 \\ 5632 & 1554 & 2958 \\ 6656 & 1836 & 3496 \\ 7168 & 1977 & 3765 \\ 8192 & 2260 & 4303 \\ 9216 & 2542 & 4840 \end{bmatrix}$$

4 试验结果

我们通过修改 H. 264 参考软件 JM7. 3 来执行新的变换和量化方法。我们对几个 QCIF 和 CIF 格式的视频序列进行了测试。实验条件为 QP 等于 16, 20, 28 和 32, 两个参考帧, 运动向量的搜索范围为 32, 用 IPP...P 帧格式编码, 编码频率为 30Hz。图 2 给出了变换(5, 2)和变换(2, 1)的率失真曲线比较, 表 2 给出了平均 PSNR 增益和平均比特率节省。

表 2 变换(5, 2)和变换(2, 1)的率失真性能比较

序列		平均 PSNR 增益 (dB)	平均比特率节省 (%)
QCIF	News	+0.063	+0.93
	FlowerGarden	+0.16	+1.56
	Foreman	+0.03	+0.76
	Silent	+0.085	+1.28
CIF	News	+0.036	+0.67
	Mobile	+0.03	+0.41
	Foreman	+0.033	+0.74
	Tempete	+0.058	+0.97
	Flowergarden	+0.06	+0.8
	Container	+0.002	+0.1
Average		+0.056	+0.82

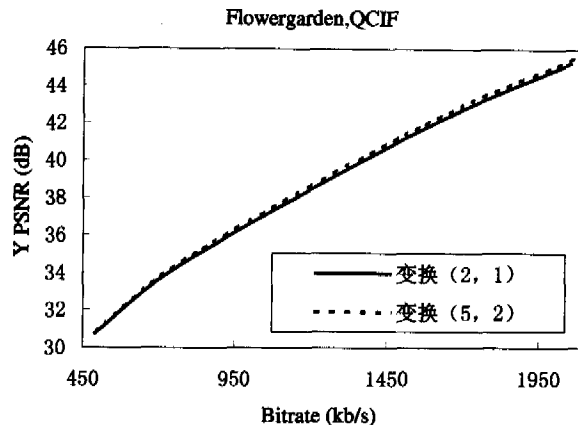


图 2 率失真曲线比较

从图 2 和表 2 中我们可以看到, 变换(5, 2)的率失真性能优于变换(2, 1), 其平均 PSNR 增益为 0.056dB。变换(5, 2)的缺点是在正变换和反变换的列变换或行变换的计算需要多 2 次加法和移位, 但采用该变换带来的性能增益足以弥补计算量的增加, 因为加法和移位操作的计算非常简单。

结论 文中提出一种新的 4×4 整数变换矩阵, 该整数变换矩阵非常接近 DCT 变换, 并且无需乘法运算, 只需要 16 位的加法和移位操作, 因此大大降低了计算的复杂度。试验结果表明: 该变换在率失真性能上优于变换(2, 1), 其平均 PSNR 增益为 0.056 dB。

参考文献

1 Malvar H S, Hallapuro A, Karczewicz M, et al. Low-complexity transform and quantization in H. 264/AVC [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 598~603

2 Wiegand T, Sullivan G J, Bjontegaard G, et al. Overview of the H. 264 / AVC video coding standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 1~19

3 Rao K R, Yip P. Discrete Cosine Transform: Algorithms, Advantages, Applications. Boston: Academic Press, 1990

4 Cham W. Development of integer cosine transforms by the principle of dyadic symmetry [J]. Proceedings of IEE Communications, Speech and Vision, 1989, 136: 276~282

5 Pei Soo-Chang, Ding Jian-Jiun. The integer transforms analogous to discrete trigonometric transforms [J]. IEEE Trans on Signal Processing, 2000, 48(12): 3345~3364

6 Wu F S, Cham W K. A comparison of error behaviour in the implementation of the DCT and the ICT [C]. IEEE Region 10 Conference on Computer and Communication Systems, Tencon, 1990