

一种 Colored WF_logic Net 的工作流过程建模

王静馨 李艳梅 徐娟 (兰州交通大学 电子与信息工程学院 甘肃 兰州 730070)

摘要: 结合着色 Petri 网和 WL_net 相关理论, 提出有色工作流逻辑网(CWL_net)这一概念来实现工作流的过程建模。最后以保险索赔业务过程为例, 采用绘制可达树的方法分析了业务流程的合理性。利用 CWL_net 可以准确描述业务流程的工作流逻辑, 且这种逻辑结构可以区分工作流具体流程中不同变迁产生的任务完成信息, 避免了某些问题。

关键词: 工作流; 工作流过程建模; 着色 Petri 网; 有色工作流逻辑网

Workflow Process Modeling Based on Colored WF_logic Net

WANG Jing-Xin, LI Yan-Mei, XU Juan

(School of Electornics and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

Abstract: After introducing the theory of Colored Petri Net and WL_net, this paper proposes a colored workflow logic net to achieve the modeling of workflow. In a case study about handling insurance, reachability graph is employed to analyze the feasibility of the process. By using CWL_net, a business process flow could be modeled in a definite way, and in this workflow model the logic of process could distinguish the task finished information produced in different transitions.

Keywords: workflow; workflow process modeling; colored Petri net; colored workflow logic net

1 引言

工作流建模(也称流程建模)就是将现实世界中的业务过程抽象出来, 并用一种形式化的、计算机可处理的方式来表示, 这种形式化结果就称为工作流模型^[1]。工作流模型是目前工作流技术研究的热点, 过程模型是其核心。本文主要探讨工作流过程模型的建立。可支持过程管理的建模工具主要有: IDEF 系列方法、Petri 网模型、ECA、EPC 等^[2]。其中, 作为描述与分析离散事件动态系统的模型工具——Petri 网, 非常适于异步并发系统的建模。在运用 Petri 网进行工作流建模方面, 已经有很多国内外专家做了大量的研究。其中, 荷兰学者 Aalst 以 EN_系统为基础提出的工作流网(WF_net), 借用活性和有界性等概念对工作流网的安全性和健壮性等概念做出了定义。Aalst 的这一模型是基于 Petri 网的工作流模型中最有影响的一种。然而, 由于 WF_net 混淆了工作流逻辑的控制与任务, 没有区分逻辑、语义和管理的界限, 使 WF_net 中的托肯

含义发生转变, 从而致使网络中出现错误和冗余, 不能清晰地描述业务流程。

文献[3]中在 Petri 网的通用网论中同步论的基础上, 提出了同步器的概念, 同步器是工作流逻辑网(WL_net)中的库所, 用于建模过程的逻辑结构。同步器不但能将业务过程连接成一个整体, 给出完整的过程结构, 而且能够控制任务的执行, 确保任务间的动态同步, 它已成为 Petri 网建模方法中最有效的工具^[4]。但 WL_net 在实际建模中出现的问题或者错误, 也主要是由同步器的引入造成的, 即是由 WL_net 在对 P/T 系统的变迁规则上另加的两点约束造成的。

2 工作流逻辑网分析

2.1 同步网

工作流逻辑网和工作流语义网概念的提出, 将工作流的逻辑和语义划分开, 增强了业务流程的描述能力。而描述工作流逻辑网的同步网, 清晰地表达了业

务流程中任务之间的依赖关系,下面简要介绍同步网。

定义 1. 同步网^[3]: 加权有向图 $WN=(P,T;F,K,W)$ 称为同步网, 简称 Syn_net , 如果 $(P,T;F)$ 为有限有向网, 而且:

$$\forall t \in T: t \neq \emptyset \wedge t \neq \emptyset$$

$$\wedge \forall p \in P: p \neq \emptyset \wedge p \neq \emptyset \Rightarrow p = (T_1, T_2, (a_1, a_2))$$

$$\wedge \left| \{p \mid p = \emptyset\} \right| = 1$$

$$\wedge F^+ (F^{-1})^+ = \emptyset$$

$$\wedge \forall P: P = \emptyset \vee P = \emptyset \Rightarrow$$

$$k(p) = 1 \wedge \forall t \in p: W(t, p) = 1 \wedge t \in p: W(p, t) = 1$$

其中, $p=(T_1, T_2, (a_1, a_2))$ 表示 p 为同步器, 且 $T_1 = pT_2 = p, K(p) = a_1 \times a_2$, 只要 $p \neq \emptyset \wedge p \neq \emptyset, p$ 必须是同步器。 p 的结构如图 1 所示, $T_1 = \{t_{11}, \dots, t_{1m_1}\}, T_2 = \{t_{21}, \dots, t_{2m_2}\}, 1 \leq a_1 \leq T_1, 1 \leq a_2 \leq T_2$, 且 $\forall t \in p: W(t, p) = a_2 \wedge \forall t \in p: W(p, t) = a_1$, 表示从 T_1 中选中 a_1 个任务执行, 执行后可授权 T_2 中 a_2 个任务执行。

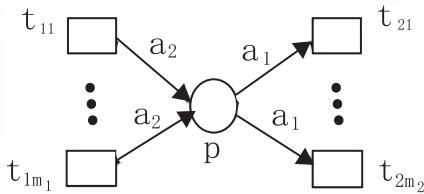


图 1 同步器

2.2 workflow 逻辑

在 Syn_net 的基础上, 文献[3]中定义了 workflow 逻辑 $WF_logic = (P, T; F, K, W, M_0)$, 其变迁规则是对 P/T 系统的变迁规则加上以下两点约束:

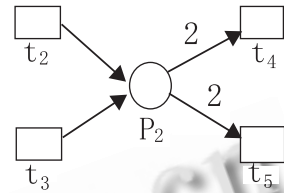
定义 2. workflow 逻辑网的变迁规则

- (1) 每个变迁至多只能发生一次。
- (2) 同步器 $p=(T_1, T_2, (a_1, a_2))$ 只有在 $M(p) = a_1 \times a_2$ 时才能同步授权给 T_2 中变迁发生权。

2.3 分析

如图 2 所示, 这是保险索赔的 workflow 逻辑图中的一部分, p_2 是 $\{t_2, t_3\}$ 和 $\{t_4, t_5\}$ 的同步器。根据 WF_logic 附加的第二条规则, p_2 只有在 $M(p_2) = a_1 \times a_2 = 2 \times 2 = 4$ 时, 才能同步授权给变迁 t_4 和 t_5 发生权。此外又有说明, 当变迁 t_4 和 t_5 从 p_2 中获得发生权以后并不必同步发生, 即 t_4 和 t_5 并不是必须同时执行, 只要并行执行就可。这里就出现问题了: 若 t_4 和 t_5 中任意一个获得

发生权并执行后, 它将会消耗掉 p_2 中的 2 个托肯, 这时 p_2 中的托肯数变成 2 (小于 4), 根据变迁规则(2)可知, $\{t_4, t_5\}$ 中剩下的一个变迁便失去了发生权, 这与保险索赔的 workflow 逻辑图的设计初衷是完全不同的。



t_2 —检查险种; t_3 —检查索赔依据; t_4 —发拒赔信; t_5 —赔付。

图 2 保险索赔的 WF_logic 中的同步器

p_2 中的 4 个托肯中, 其中 t_2 在 p_2 中产生的 $2(a_2)$ 个托肯代表 t_2 中任务完成信息的 2 个拷贝, 即同一 B_form 的两份副本(我们记为 B_1), t_4 和 t_5 各一份。 t_4 和 t_5 均需等待从 t_3 来的另一份。同理, t_3 在 p_2 中产生的 $2(a_1)$ 个托肯, 我们记为 B_2 。也就是说, p_2 中的 4 个托肯是 $\{B_1, B_1, B_2, B_2\}$ 。那么 t_4 和 t_5 中任意一个获得发生权后消耗的 2 个托肯有可能是 $\{B_1, B_1\}$ 、 $\{B_2, B_2\}$ 、 $\{B_1, B_2\}$ 三种情况之一。依前言, t_4 和 t_5 分别需要 B_1 和 B_2 这两个副本才可获得发生权。然先获得发生权的变迁由于 B_1 、 B_2 含义不同在 p_2 中并未表现出来, 而不能分辨自己将要选择的 2 个托肯, 从而使剩下的一个变迁有 $2/3$ 的几率是不能发生的。

基于以上分析, 我们知道 workflow 逻辑网在实际建模过程中存在一定的问题, 这些问题的根源是同步器中托肯具有不同的含义。在同步器 p 中, T_1 中的任意一个变迁 t 在 p 中所产生的 a_2 个托肯, 代表变迁 t 中任务完成信息的 a_2 个拷贝; T_2 中任意一个变迁要发生, 都需获得 T_1 中 a_1 个不同任务的完成信息。即 T_1 中不同变迁产生的托肯含义是不一样的, 但是同步器采用的是“同步授权”, 这种托肯含义的不同并未在其中表现出来。从这个意义上说, 同步授权时不现实的。而且, 依 Petri 网的定义, 网中只有库所是存放资源的。那么“授权”后, T_3 中的变迁没有发生, p 中的托肯不可能在这些变迁的后继库所里, T_2 中的变迁又不存放任何资源, 它只是一个消耗资源和产生新资源的处理过程。由此得出“授权”产生了一种介于库所和变迁之间的中间态。这种变迁不发生, 却“占有”

库所的状态在 Petri 网中是不允许存在的。同时, 这种情况也使我们不可能构造出网络的可达图, 从而给网络安全性和健全性的分析带来了巨大的困难。一种解决方法是通过扩展 Petri 网, 给每个托肯都提供一个值以便来区分托肯。给托肯赋值, 就好比给它们涂上了不同的颜色。这种扩展的 Petri 网为着色 Petri 网 (Colored Petri Net, CPN)^[5]。

3 着色 workflow 逻辑网

基于上述分析, 本文在着色 Petri 网和 workflow 逻辑网概念的基础之上, 引入有色 workflow 逻辑网 (CWL_net) 这一概念来解决由于变迁所产生的托肯含义不同, 而同步网却不能表现出这种不同, 致使在实际业务流程建模过程中而产生的一些问题。为了便于理解, 先简要介绍一下 CPN 的概念。

定义 3. 着色 Petri 网: $CPN=(P, T, A, C, G, E, I)$ 称为着色 Petri 网(符号和术语参考文献[5]), 且满足:

(1) P 是一个由颜色集构成的非空有限集合, 称为颜色集(color set), 它的每个元素, 即一个颜色集代表了一种类型; 类型集决定了数据值及能够在网表达式中使用的操作和函数。类型可使用抽象数据类型方法加以定义。类型中必须包含 Case_form, 为托肯所属的数据类型;

(2) P 是由库所构成的有限集;

(3) T 是由变迁构成的有限集; P 和 T 满足: $P \cap T = \emptyset$; $\forall t \in T: t \neq \emptyset$; $\forall p \in P: p \neq \emptyset \wedge p \neq \emptyset \Rightarrow p = (T_1, T_2; (a_1, a_2))$, $\{p | p = \emptyset\} = \pm 1$;

(4) A 是由连接库所到变迁或变迁到库所的有向弧构成的有限集, 且 $P \cap T = P \cap A = T \cap A = \emptyset$, 表明 CPN 中没有有向环;

(5) C 是颜色函数: $(P \times T) \rightarrow SS$, SS 为 P 的有限子集, 它是 P 到 SS 的映射;

(6) G 是守卫函数, 它将变迁 $t(T)$ 映射成这样一个表达式: $\forall t \in T: [Type(G(t)) = BType(Var(G(t))) \subseteq]$, 其中, $B = [true, false]$, 即表达式中涉及的所有变量的类型都必须是属于 SS 的, 而且该表达式的值是布尔值;

(7) E 是弧表达式函数, 它将弧 $a(A)$ 映射成这样的表达式: $\forall a \in A: [Type(E(a)) = C(p(a))ms \wedge Type(Var(E(a))) \subseteq]$, 即表达式中涉及的所有变量的类型都必须是属于 SS 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 称为初始化函数, 它将库所 $p(P)$ 映射成这样一个表达式: $\forall p \in P: [Type(I(p)) = C(p)ms]$, 即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. 有色 workflow 网(CWF_net): 一个 $CPN=(P, T, A, C, G, E, I)$ 是有色 workflow 网, 当且仅当:

(1) 存在 p 的两个子集 IN 和 OUT , 即 $IN, OUT \subseteq p$, 且 $[IN = \pm 1, OUT = -1 \text{ 且 } OUT = 1]$, $\forall i \in IN, i = \emptyset$, $\forall o \in OUT, o = \emptyset$;

(2) $\forall x \in P \cup T \wedge x \in IN \wedge x \notin OUT$, x 至少在一条从 $i \in IN$ 到 $o \in OUT$ 的路径上。

定义 5. 变迁的使能

当变迁 t 满足如下关系: $M(p) \geq E(p, t)$ 且 $G(t) = true$, 我们就说变迁 t 使能。

变迁 t 在标识 M 下有发生权, 记为 $M[t >]$ 。

定义 6. 变迁的触发

获得使能的变迁完成触发后, 产生新的颜色集并且使得后续变迁获得使能, 则标识 M 变成其后继 M' , 系统推进到一个新的状态, 即 $M[t > M']$:

$\forall p \in P: M'(p) = M(p) - E(p, t) + E(t, p)$, 其中, $E(p, t)$ 表示消耗的托肯, $E(t, p)$ 表示新生的托肯。

定义 7. $< \bullet$ 称为 CWF_net 中变迁的后继关系, 如果 $< \bullet = \{(t, t') | t, t' \in T \wedge \exists p \in P: t \in p \wedge t' \in p\}$ 。

文献[3]中描述了变迁后继的传递性会在网络中引入多余的托肯。因而, 在 workflow 逻辑中, $< \bullet$ 关系不应具有传递性。

定义 8. 有色 workflow 逻辑网(CWL_net): 一个 $CWF_net=(P, T, A, C, G, E, I)$ 是有色 workflow 逻辑网, 当且仅当:

(1) 对于 CWF_net 中的 $< \bullet$ 关系, $< \bullet \circ < \bullet = \emptyset$, 且 $(T, < \bullet)$ 为连通图;

(2) $\forall p \in P(p = \emptyset \Rightarrow M(p_0) = 1 \wedge C_0 = 1) \wedge (p \neq \emptyset \Rightarrow M(p_0) = 0 \wedge C_0 = 0)$ 。

其中, (1) 规定了 CWF_net 中的后继关系不存在传递性, (2) 规定了在初始状态下, 只有入口库所中含有一个托肯, 其他库所中均不含托肯。

4 业务实例及合理性分析

下面用本文中有色 workflow 逻辑网对保险索赔的业务流程来建模。图 3 是保险索赔的流程图。所有权重是 1 的在图中都被省略掉而没有标出。其中 t_1, t_2, t_3, t_4 ,

和 t_5 依次表示任务“接受赔付申请”、“检查险种”、“检查索赔依据”、“发拒赔信”和“赔付”。

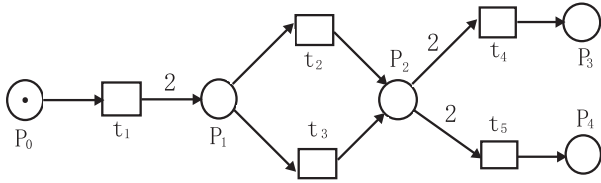


图3 保险索赔流程图

我们用 CWL_net 来对图 3 表示的流程建模。模型中所有库所中的托肯都为有色托肯。库所 p_0 中的托肯表示有索赔申请,可以看出,其经传递、复制及合并最终会流动到终点库所 p_3 或 p_4 中,即索赔受理只会产生两种结果:“拒赔”或“赔付”。这证明了模型的畅通性。

模型的可达性分析方法主要有:可达树/图法(reachability tree/diagram)、矩阵方程法(matrix equation)、约简或分解法(reduction or decomposition)。由于 CWL_net 中托肯是有颜色的,而且我们是针对 p_0 中有托肯这一初始状态的,因此考虑用可达树证明其可达性。根据传统 P/T 系统可达树的构建方法,结合 CWL_net 的定义和其变迁的使能及触发规则,即可实现 CWL_net 可达树的构造算法。图 4 是利用该算法生成的图 3 中业务流程的可达树。

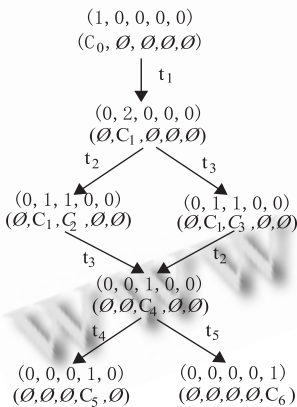


图4 保险索赔流程的可达树

上面得到的可达树中,每个状态包括各个库所中托肯的个数集和颜色集。其中, C_0 是初始状态时 p_0 中托肯的颜色集, $C_1=\{C(t_1)\}$, $C_2=\{C(t_2)\}$, $C_3=\{C(t_3)\}$, $C_4=\{C(t_2), C(t_3)\}$, $C_5=\{C(t_4)\}$, $C_6=\{C(t_5)\}$ 。图 3 中的每一个变迁都在可达树中出现了,因此 CWL_net 中不存在死变迁;图 4 中的可达树中,每一个标识都

出现过一次,因此不存在死标识。图 4 中死节点只有两种状态: $(0, 0, 0, 1, 0)$, $(\emptyset, \emptyset, \emptyset, C_5, \emptyset)$ 和 $(0, 0, 0, 0, 1)$, $(\emptyset, \emptyset, \emptyset, \emptyset, C_6)$, 分别代表 p_3 中有一个托肯,其他库所中均没有托肯; p_4 中有一个托肯,其他库所中均没有托肯。因此,流程中只有“拒赔”或“赔付”两种可能的结果,且流程中不含冗余信息。

经以上分析,证明用 CWL_net 建立的保险索赔业务流程模型是合理的。

5 结论

本文在研究了基于 Petri 网的工作流建模理论的基础上,针对同步器中托肯含义由于未加严格限定,可能导致“同步授权”的不可实现性等一系列问题,通过对 WL_net 的分析和 CPN 的扩展,提出了有色工作流逻辑网的概念,并在此基础上提出了一种工作流过程建模方法。该建模方法拥有严格的形式化数学定义,并可以利用成熟的 Petri 网性能分析方法对构造成的业务流程进行合理性分析。有色工作流网将业务流程中任务之间的依赖关系清晰地划分开,使模型中逻辑结构与任务实现相分离,流程结构的设计不必依赖任务的实现方式。本文着重考虑了对工作流模型的设计,但对一个规模较大的企业来说,企业业务流还涉及到其他方面,如企业资源和组织人员等,如何把工作流模型与其他模型(功能模型、信息模型、资源模型等)进行有效集成,从而可支持企业更复杂的过程模型,值得进一步研究。

参考文献

- 1 C'ordoba A, Villadangos J, Astrain J J, et al. A B2B Replication Service.roc of the 13th Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2005. 337 - 343.
- 2 汤双权,可音,余青,等.工作流过程模型研究.计算机技术与发展,2007,17(12):44 - 52.
- 3 袁崇义. Petri 网原理和应用.北京:电子工业出版社,2005.
- 4 王斌,章云,王晓红.基于 Petri 网的工作流模式建模及应用.计算机工程与应用,2008,44(13):238 - 241.
- 5 Jensen K. Coloured Petri Nets-Basic ConcePts, Analysis Methods and Practical Use(2nd Ed). Springer, New York, 1997.