

基于 Petri 网的工作流模型性能分析方法*

李建强 范玉顺

清华大学自动化系, 北京 100084

摘要: 除了有限的仿真支持以得到参考数据以外, 目前对于工作流模型性能的理论上的严格分析几乎是空白。在深入分析了经营过程工作流模型特点的基础上, 本文提出一种将工作流的 Petri 网模型分解为事件图的算法, 并给出了利用事件图对单个业务实例的处理过程进行性能分析的基础上获得整个工作流模型性能指标的方法。

关键字: Petri 网, 性能分析, 工作流技术

一、引言

从研究中我们认识到工作流模型主要以过程模型为中心, 并将企业模型的信息视图、资源视图、组织视图和功能视图映射进来以完成对企业经营业务过程的完整描述。目前有很多方法都可以用来进行工作流模型的定义和描述, 如: WFMC (工作流管理联盟) 定义的工作流描述语言、EPCM 模型、PERT 图及 Petri 网等。尽管各种工作流建模方法形式上不尽相同, 但它们都能提供对逻辑顺序结构, 资源使用, 路由控制信息等流程定义所必须的要素的描述。由于 PERT 图或其它各种形式的有向网络图具有简单、直观且易于在已有的项目管理软件工具上扩展实现等特点, 所以在实用系统的工作流建模工具中采用的比较多。而对于 Petri 网而言, 尽管其对经营过程的图形化的表达的可读性方面存在不足, 但由于其严格的数学基础和规范化的语义^[1], 对于工作流的建模与分析研究人员来讲有很大的诱惑力。特别是当考虑的经营过程较为复杂, 如存在并发、冲突等情形时, 用 Petri 网方法进行工作流模型分析有明显的优势。因此, 在实际应用中可以采用直观易读的有向网络图进行工作流建模, 而在模型分析时, 通过一种具有语义不变性的映射方法将有向网络图转换为相同语义结构的 Petri 网, 进而利用丰富 Petri 网理论进行模型分析。

由于 Petri 网的复杂性, 很难有一种通用的方法可以进行所有系统模型的分析, 目前, 大部分研究人员都只对于描述某一特定系统的某一种特殊类型的 Petri 网提出了分析或综合的方法。根据现在绝大部分的工作流建模工具所建立的工作流模型的语义特性^[2]和经营过程中具体业务实例在企业的业务处理流程中路由选择只依赖于业务实例本身的属性的特点, 我们可以得出绝大部分工作流模型都具有自由选择语义的结论 (见第二节模型映射)。因此, 工作流模型可以映射为自由选择 Petri 网。另外, 每个工作流过程模型描述某一类业务的处理过程, 而这一类的业务会包含多种业务实例。对于每一种业务实例, 业务过程模型中都应该有一种业务处理路由与其相对应, 我们可以把这种业务处理路由称为实例过程逻辑。在此启发下, 我们把描述工作流过程模型的选择 Petri 网分解为对应于每一种业务实例过程逻辑的事件图。而对于事件图的性能分析方法已经很成熟, 我们可以把它们引入到工作流的性能分析中来。本文的第二节提出了一种工作流模型到自由选择 Petri 网的映射方法; 第三节详细讨论了将具有自由选择特性的工作流网分解为分别对应于每种具体业务实例处理过程的事件图的方法; 第四节提出了利用事件图对单个业务实例过程逻辑的性能分析结果得出整个工作流模型的性能指标的工作原理。

二、模型映射

这里只对我们将要用到的 Petri 网^[1]和工作流网^[2]的一些概念或定理作简单介绍:

* 中国博士后基金资助

定义 1: Petri 网 $PN=(P,T,F)$ 称为自由选择 Petri 网, 如果任意变迁 t_1 和 t_2 满足 $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ 则 $\bullet t_1 = \bullet t_2$ 。

定义 2: Petri 网 $PN=(P,T,F)$ 称为事件图, 如果任意库所 p 只有一个输入变迁和一个输出变迁。

定义 3: 一个 Petri 网 PN 被称为 workflow 网, 当且仅当它满足下面的两个条件:

- (1) PN 有两个特殊的库所: i 和 o 。库所 i 是一个起始库所, 即 $\bullet i = \Phi$; 库所 o 是一个终止库所, 即 $o \bullet = \Phi$ 。
- (2) 如果在 PN 中加入一个新的变迁 t^* , 使 t^* 连接库所 o 与 i , 即 $\bullet t^* = \{o\}, t^* \bullet = \{i\}$, 这时所得到的 PN 是强连接的。

定义 4: 在 workflow 网 PN 中加入一个新的变迁 t , 使 t 连接库所 o 与 i , 即 $\bullet t = \{o\}, t \bullet = \{i\}$ 称此 PN 为扩展 workflow 网。

定义 5: 对于 Petri 网 $PN=(P,T,F)$ 和 $PN_1=(P_1,T_1,F_1)$, PN_1 是 PN 的子网当且仅当 $S_1 \subseteq S, T_1 \subseteq T$ 和 $F_1 = F \cap ((S_1 \times T_1) \cup (T_1 \times S_1))$ 。

定义 6: $PN_1=(P_1,T_1,F_1)$ 为 $PN=(P,T,F)$ 的子网, $PN_1=(P_1,T_1,F_1)$ 为变迁构成网当且仅当 $P_1 = T_1 \bullet \cup \bullet T_1$ ($T_1 \bullet$ 和 $\bullet T_1$ 是指在 PN 中)。

定义 7: $PN_1=(P_1,T_1,F_1)$ 被称为 $PN=(P,T,F)$ 的 T -组件网当且仅当 PN_1 为 PN 的变迁构成网且 $\forall p \in P_1: |\bullet p \cap T_1| \leq 1 \wedge |p \bullet \cap T_1| \leq 1$ 。

定理 1: workflow 网 PN 具有完整性的充分必要条件是扩展 workflow 网具有有界性和活性^[2]。

下面我们提供一种映射方法将 workflow 联盟提供的过程描述实体^[3]分别转化为 workflow 网 (Petri 网): 每个活动 A (选择/汇合节点 D) 映射为一个库所 Ap (Dp), 并且针对于每一个输出为非同步结构活动的路径产生一个以 Ap (Dp) 为输入库所的变迁 At (Dt), 同时变迁 At (Dt) 的输出库所是所有活动 A 的输出活动 (当输出为有同步结构的的活动 S 时, 见后)。对于有同步结构的的活动 S, 同样用一个库所 Sa 和其对应, 但产生两个变迁 Si 和 So : Si 为 Sa 的输入变迁, So 为 Sa 的输出变迁。对于同步 S 的输入: 当为任务时, 每个活动 SA 映射为库所 SAP , SAP 的输出变迁为 Si ; 当为选择/汇合节点 D 时, 我们创建库所 DSP 作为 Dt 的输出库所和 Si 的输入库所。最后, 在起始库所和终止库所之间加入一个新的变迁 t , 使 $\bullet t = \{o\}, t \bullet = \{i\}$ 。显然, 得到的扩展 workflow 网模型肯定是自由选择 Petri 网。通过后面的分析可以说明这种映射获得的工作流网模型不改变原工作流模型的语义 (其实关键是对同步节点的映射)。

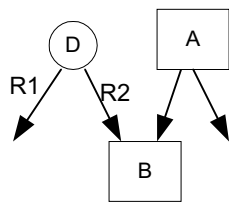


图 1. 形式上的非自由选择结构

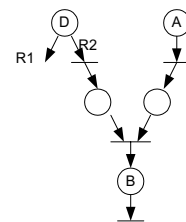


图 2. 非自由选择结构的 Petri 网映射

现在绝大多数 workflow 建模工具对于 workflow 活动之间的逻辑关系一般是通过事务在过程模型中的路由来体现的, 而每一具体事务的路由选择是由活动对事务处理的结果也就是事务本身的特性 (经过活动处理后的状态) 决定的。在此条件下, 对于有些表面形式上不具有自由选择特性的模型结构 (如图 1), 在转换为自由选择特性的结构后 (图 2), 虽然从形式上似乎改变了模型的逻辑关系, 但并没有改变过程模型语义: 选择/汇合节点 D 对输出路径 R1、R2 的选择与活动 A 是没有关系的, 因为它的选择是根据节点 D 中的事务特性决定的, 在这种情况下假如活动 A 没有执行, 它就会一直等下去, 直到活动 A 执行结束才开始执行活动 B。从这个角度说大部分 workflow 过程模型本身具有自由选择的特性。所以用前面

提供的模型映射方法得到的扩展 workflow 网（具有“自由选择”特性）和原来 workflow 模型具有相同的语义。

三、模型分解

在文献[4]中已经证明任意具有活性和安全性的自由选择 Petri 网都被强连接的事件图所覆盖，并且给出一种由任一变迁生成包含此变迁的 T —组件网的算法。很明显，此 T —组件网即为某种业务实例的处理过程。在此基础上，我们提供一种可以有效的将不存在循环结构且具有完整性的处理一类业务实例的自由选择 workflow 网模型分解为处理单个业务实例的事件图子网的方法。我们这里要求完整性是因为如果在起始库所中有一个托肯，则完整性必保证活性和安全性的要求。而假如模型中存在循环结构则获得的事件图子网中必有一些不包括起始节点和终止节点，则这些事件图肯定不能对应于业务实例处理过程。对于在实际中的循环处理过程我们可以用一个循环块来表达，以避免循环结构的出现。在给出分解算法之前，先对其中要用到的两个重要概念进行定义。

定义 7: $PN_1 = (P_1, T_1, F_1)$ 为 $PN = (P, T, F)$ 的子网， $path = \{t_0, p_1, t_1 \cdots p_m, t_m\}$ 为变迁路径当且仅当：

- $(t_{j-1}, p_j) \in F \wedge (p_j, t_j) \in F (1 \leq j \leq m)$
- $path$ 为简单路径（在 $path$ 中每个元素只出现一次）
- $t_0 \in T_1 \wedge t_m \in T_1$
- $p_j \notin P_1 (1 \leq j \leq m)$ 且 $t_j \notin T_1 (1 \leq j < m)$

虽然这里的变迁路径与文献[4]中的 *Nice path* 的定义略有不同，但它们具有相同的语义结构，并且一一对应。对称的，我们给出库所路径的定义：

定义 8: $PN_1 = (P_1, T_1, F_1)$ 为 $PN = (P, T, F)$ 的子网， $path = \{p_0, t_1, p_1 \cdots t_m, p_m\}$ 为库所路径当且仅当：

- $(p_{j-1}, t_j) \in F \wedge (t_j, p_j) \in F (1 \leq j \leq m)$
- $path$ 为简单路径（在 $path$ 中每个元素只出现一次）
- $p_0 \in P_1 \wedge p_m \in P_1$
- $t_j \notin T_1 (1 \leq j \leq m)$ 且 $p_j \notin P_1 (1 \leq j < m)$

以下我们给出从自由选择扩展 workflow 网 \hat{N} 到 T —组件网集合 \mathfrak{R} 的分解算法：

步骤 1: 在库所集合 $P = \{p \mid |p| \geq 2\}$ 中任取库所 p ，对应于其后集 $T = (t_1, t_2, \cdots t_n) = p^\bullet$ ，得到子网集合 $\mathfrak{R} = \{\hat{N}_i \mid \hat{N}_i = (\hat{P}_i, \hat{T}_i, \hat{F}_i)$ 为包含变迁 $t \in T$ 的基本回路}；

步骤 2: 对于 $\forall \hat{N}_i \in \mathfrak{R}$ 库所集合 $\mathfrak{K} = \{p' \mid |p'| \geq 2$ （在 \hat{N} 中）并且 $\forall \hat{N}_i \in \mathfrak{R} : p' \notin \hat{P}_i\}$ 非空，则对于每一个 p' 的后续变迁集合 $\{t \mid t \in p'^{\bullet} \wedge t \notin \hat{N}_i\}$ 中的每一个变迁 t ，在集合 \mathfrak{R} 中添加包含变迁 $t \in T$ 基本回路子网 $\hat{N}_i = (\hat{P}_i, \hat{T}_i, \hat{F}_i)$ ；

步骤 3: 对于 \mathfrak{R} 中所有新添加的 \hat{N}_i 重复步骤 2，直到遍历每个 \hat{N}_i 的非空 \mathfrak{K} 中的每一个库所，得到新的子网集合 $\mathfrak{R} = \{\hat{N}_i \mid \hat{N}_i = (\hat{P}_i, \hat{T}_i, \hat{F}_i)$ 为基本回路}；

步骤 4: 对于 \mathfrak{R} 每一个子网 \hat{N}_i ，如果变迁 $t' \in \hat{T}_i$ 在 \hat{N} 中所对应的集合 $S = \{p \mid p \in t'^{\bullet} \wedge p \notin \hat{P}_i\}$ 为非空，则任选 $p_1 \in t'^{\bullet} \setminus \hat{P}_i$ ，存在从 $t' = t_0$ 经过 p_1 返回到 \hat{N}_i 的变迁路径 $path = \{t_0, p_1, t_1 \cdots p_m, t_m\}$ ，得到新的子网 $\hat{N}_i = (\hat{P}_i, \hat{T}_i, \hat{F}_i) = \hat{N}_i \cup path$ ；

4.1: 如果 $path$ 中库所集合 $\Pi = \{p' \mid |p'| \geq 2$ （在 \hat{N} 中）并且 $\forall \hat{N}_i \in \mathfrak{R} : p' \notin \hat{P}_i\}$ 非空，则对于每一个 p' 的变迁集合 $\{\bar{t} \mid \bar{t} \in p'^{\bullet} \wedge \bar{t} \notin \hat{T}_i\}$ 中的每一个变迁 \bar{t} ，存在经过 p' 和 \bar{t} 返回到 $path$ 中库所 p'' 的库所路径 $path1$ ，在集合 \mathfrak{R} 中添加子网

$\hat{N}_i = \hat{N}_i \cup path2$ ($path2$ 由 $path1$ 替换 $path$ 中 p' 和 p'' 间的路径获得);

4.2: 对变迁路径 $path1$ 重复步骤 4.1, 直到遍历每个 \hat{N}_i 的非空 Π 中的每一个库所;

步骤 5: 重复步骤 4, 直到 \mathfrak{R} 中每一个子网 \hat{N}_i 中每一个变迁所对应的 S 都为空;

文献[4]已经对步骤 4 中变迁路径 $path$ 对于库所 p_1 的存在性进行了证明。实际上, 根据满足完整性的自由选择扩展工作流网的特点 (每个闭环回路都经过起始库所 i 和终止库所 o), 由 t' 出发的任一路径都必能返回到 \hat{N}_i 且返回节点必为变迁, 从而变迁路径 $path$ 必存在。而且对于步骤 4.1, 可以证明从 p' 出发的路径 $path1$ 的返回节点必为 $path$ 中的库所, 因为假如不在 $path$ 中, 则 $path1$ 的返回到 \hat{N}_i 的节点或者为库所、或者为不同于 t_m 的变迁, 而这两种情况都会破坏扩展工作流网的完整性。而假如在 $path$ 中而返回节点非库所, 则同样与扩展工作流网的完整性相矛盾。从而可以得出库所路径 $path1$ 必存在。

容易看出, 步骤 1—3 得到的集合 \mathfrak{R} 是由经过库所 p 的简单回路中的所有分支库所衍生出来的所有基本回路的集合。每一个基本回路又经过步骤 4、5 的重复实施, 并入其中所有并发变迁产生的变迁路径而得到变迁构成网, 其有效性已在文献[4]中得到证明。而在新并入的变迁路径中也会存在分支选择库所, 为了保证得到所有的 T —组件网, 步骤 4.1、4.2 将由此衍生出的所有会扩展成变迁构成网的子网都并入到集合 \mathfrak{R} 中, 并利用步骤 4、5 将其扩展为变迁构成网。而对于每一个变迁构成网中的每一个库所只在步骤 2、3 或 4 中只处理过一次, 或者出现在基本回路中, 或者出现在变迁路径中。而在这两种情况下每个库所都只有一个输入一个输出弧, 所以可以肯定最后得到的子网为事件图, 从而满足 T —组件网的条件。

由以上的分析我们可以得出结论: 对于自由选择扩展工作流网 $PN=(P,T,F)$ 必可应用以上的分解算法获得 T —组件网的集合 \mathfrak{R} , 并且 \mathfrak{R} 中包含了 PN 中可以分解出来的所有 T —组件网。显然, 每个 T —组件网对应于一种业务实例的处理过程, 或者说是业务实例在工作流模型中的一种路由。

四、模型性能评估

我们假设要分析的工作流模型共可以处理 n 种业务实例 $\{R_1, R_2, \dots, R_n\}$, 每种业务实例 R_i 在此工作流模型处理的所有业务中所占的比例也是一定的。对此, 有向量 $P=\{p_1, p_2, \dots, p_n\}$ 满足 $p_1 + p_2 + \dots + p_n = 1$, 其中 p_i 为对应 R_i 在业务总数中所占的百分比。下面我们在扩展工作流网的分解得到的 T —组件网集合 $\mathfrak{R}=\{\hat{N}_1, \hat{N}_2, \dots, \hat{N}_n\}$ (其中 \hat{N}_i 为对应于业务实例 R_i 的处理过程事件图) 的基础上分别对工作流模型从时间、成本、资源利用率三个角度进行性能分析。

4.1 时间性能

为了进行时间上的性能分析, 需要定义每个活动执行所需的执行时间。我们对应于包含有 m 个活动的工作流模型定义 $T_a=\{t_1, t_2, \dots, t_m\}$, 其中 t_i 为活动 i 执行时间的均值。这样, 每个 t_i 对应于在扩展工作流网由活动映射出来的变迁的发射时间, 而对应于由汇合/同步节点映射产生的变迁和起始库所和终止库所之间补充的变迁的发射时间则都为零。这里所以对每个活动执行时间取均值, 一方面是为了分析的方便; 另一方面是因为我们这里分析工作流模型的时间性能上下界的方法, 其结论的分布函数线性无关性已经在文献[5]中进行了证明。另外, 我们假设工作流模型系统在保证资源的使用不发生死锁的情况下最多可以同

时处理的事务数为 m ，则对应于每种业务实例的事件图 \hat{N}_i 中同时处理的事务数为 $m * p_i$ 。

我们知道， $\mathfrak{R} = \{\hat{N}_1, \hat{N}_2, \dots, \hat{N}_n\}$ 中的每一个事件图对应于一种业务实例的处理过程，要完成业务实例 R_i ， \hat{N}_i 中每一个变迁都要执行一次，这样很容易得到每一个业务实例的平均处理时间。并且 \hat{N}_i 中的每一个基本回路都必须经过起始库所和终止库所，所以每一个基本闭环回路中所含有的托肯数都为 $m * p_i$ 。利用文献[4]中的结论，很容易可以得到整个工作流程模型的时间性能上下界：

$$\text{最大输出事务率 } pr = \min_{\hat{N}_i} (\lambda_i / p_i) \quad \text{其中: } \lambda_i = \min_{\gamma} (1 / C(\gamma)) \quad (1)$$

$$\text{最小输出事务率 } pr = \min_{\hat{N}_i} (\pi_i / p_i) \quad \text{其中: } \pi_i = 1 / \mu(\hat{N}_i) \quad (2)$$

(1) 式中的 γ 为属于 \hat{N}_i 中的基本回路， $C(\gamma) = \mu(\gamma) / (m * p_i)$ 称为基本回路 γ 的循环时间，其中 $\mu(\gamma)$ 为基本回路 γ 中所有变迁执行时间的和， $M(\gamma)$ 为基本回路 γ 中所有库所重托肯数的和。(2) 式中的 $\mu(\hat{N}_i)$ 为 \hat{N}_i 中所有变迁执行时间的和。其中对于基本回路 γ 的计算，可以通过计算 \hat{N}_i 的 p -不变量来获得。

4.2 资源利用率

为了计算一个共有 k 个活动、 l 种资源的工作流模型的资源利用率，假设正常工作情况下在一个固定时间 T 内此业务流程所处理的业务总数为 s 。并且，为了清楚地描述一个事件图（对应一种业务实例的处理过程）包含那些活动，每个活动的执行需要那些资源的参与，构造活动—事件图矩阵 $B_{k \times n}$ ：

$$b_{ij} = \begin{cases} 1 & \text{如果 } j \text{ 事件图包含活动 } i \\ 0 & \text{如果 } j \text{ 事件图不包含活动 } i \end{cases}$$

和活动—资源矩阵 $R_{l \times k}$ ：

$$r_{ij} = \begin{cases} 1 & \text{如果活动 } j \text{ 的执行需要资源 } i \\ 0 & \text{如果活动 } j \text{ 的执行不需要资源 } i \end{cases}$$

通过计算 $B * P^T * s$ 得到 $N^T = \{n_1, n_2, \dots, n_k\}^T$ ，其中 N^T 中的 n_i 为第 i 个活动在这一段时间的执行次数。另外，定义 $TN_a = \{n_1 * t_1, n_2 * t_2, \dots, n_k * t_k\}$ ，则计算 $N_r^T = R * TN_a$ ， $N_r^T = \{tn'_1, tn'_2, \dots, tn'_l\}^T$ 中的每一项 tn'_i 为在这一固定时间 T 内每个资源处于工作状态的时间总和。进而可以利用公式：资源利用率=资源工作时间/资源存在时间，很容易得到每个资源的利用率。

4.3 成本

工作流程模型中成本的统计是性能分析的重要内容之一，它一般从按价值类型统计和按资源类型统计两个方面来进行。其中按价值类型统计将业务过程中的活动分为增值活动、商业增值活动和非增值活动三类，而按资源类型统计把每个活动的成本由劳动力成本、设备成本、原材料成本和其他可计算成本构成四部分。下面我们以价值成本为例说明成本的统计方法。

我们假设每一个活动处理不同业务实例的时所需要的以上各种类型的成本是相同的(当每个活动处理不同的业务实例的成本不同时,采用合适的矩阵运算同样可以得到相应的成本统计),即有 $C_T = \{c_{i1}, c_{i2}, \dots, c_{ik}\}$, 其中 c_{ii} 为活动 i 的执行成本。对于按价值类型的成本统计, c_{ii} 将根据每个活动的特点属于某一种价值类型。为此,构造活动—价值类型矩阵 $C_{3 \times k}$:

$$c_{ij} = \begin{cases} 1 & \text{如果活动 } j \text{ 属于 } i \text{ 行的增值类型} \\ 0 & \text{如果活动 } j \text{ 不属于 } i \text{ 行的增值类型} \end{cases}$$

利用前面得到的 $N^T = \{n_1, n_2, \dots, n_k\}^T$, 有 $CN_T = \{c_{i1} * n_1, c_{i2} * n_2, \dots, c_{ik} * n_k\}$, 进而计算 $C * CN_T$ 得到 $\hat{C}^T_{3 \times 1} = \{\hat{c}_1, \hat{c}_2, \hat{c}_3\}^T$, 其中的 \hat{c}_i 即为相应价值类型的总过程成本。对于按资源类型的统计可以把参与活动的资源分为四种类型,然后对应每种类型分别建立相应的矩阵进行成本计算。当然,也可以利用活动—事件图矩阵对每种业务实例 R_i 的成本构成进行统计。

五、结论

workflow模型的分析是 workflow技术中重要的研究课题之一,本文提出一种利用 Petri 网进行 workflow模型性能分析的方法。首先根据企业业务处理过程的所具有的特性以及大部分过程模型定义工具所建立的 workflow模型都有自由选择性的特点,将 workflow过程模型映射为自由选择扩展 workflow网。然后,将自由选择扩展 workflow网分解为一组事件图,每个事件图对应于一种业务实例的处理过程。在对分解得到的这组事件图进行性能分析的基础上得到整个 workflow模型时间性能、成本及资源利用率的性能指标。从而实现了将 Petri 网理论中利用事件图进行系统性能分析方法引入到 workflow技术领域中来,以进行 workflow模型的性能分析。

参考文献

- [1] Tadao Murata. Petri Nets: Properties, Analysis and Applications *Proceeding of the IEEE*, Vol.77.No.4, April 1989
- [2] W.M.P.van der Aslst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers* 8(1):21-66, (1998)
- [3] Workflow Management Coalition. *Interface 1: Process Definition Interchange, Process Model*. Document Number WfMC TC-1016-P(1998)
- [4] E.Best. Structure theory of Petri nets: the free choice hiatus. In W.Brauer, W.Reisig, and G.Rozenberg, editors, *Advances in Petri nets 1986 Part I: Petri Nets, central models and their properties*, Vol.254 of *Lecture Notes in Computer Science*, pp.168-206. Springer-Verlag, Berlin, 1987
- [5] Javier Campos, Giovanni Chiola, Jose M.Colom, Manuel Silva, Properties and Performance: Bounds for Timed Marked Graphs. *IEEE Transactions on Circuits and Systems* Vol.39, No.5, pp.386-401, May 1992

Petri Net Based Workflow Model Performance Analysis Method

Li Jianqiang Fan Yushun

Dept.of Automation, Tsinghua University, Beijing 100084

Abstract: According to the performance of workflow model, today's workflow techniques almost did nothing in theoretical analysis except gave some limited simulation support to get a few referenced indicators. In this paper, according to the characteristics of workflow model for

business process, a algorithm for decomposing free choice Petri_nets based workflow model into a set of marked graphs is proposed, and then the methods of how to take advantage of the set of marked graph to get the performance indicators of the workflow model is discussed.

Keyword: Petri nets, Performance analysis, Workflow techniques

作者简介: 李建强 男 博士研究生; 范玉顺 男 教授 博士生导师