

## 一种 workflow 模型的性能分析方法<sup>1</sup>

李建强 范玉顺

清华大学自动化系, 北京 100084

**摘要:** 本文在讨论了 workflow 模型分析的相关问题后, 基于 workflow 网<sup>[1]</sup>定义, 给出了包含资源、时间等信息的多维 workflow 网的概念, 并讨论了从基于 workflow 联盟(WfMC)过程定义<sup>[2]</sup>的一种有向网络图模型<sup>[3]</sup>到自由选择多维 workflow 网的模型映射。为了处理 workflow 模型中的循环结构, 将具有无环自由选择 workflow 网到一组变迁组件网(*T*-component)<sup>[19]</sup>的分解算法在本文中进行了扩展, 以处理 workflow 网中存在的循环结构。进而在模型结构进行分析的基础上, 根据 workflow 模型的实际业务输入速率可以计算得到 workflow 网中每个变迁的输入速率。进而, 结合开环 Jackson 排队网络理论, 详细讨论了对 workflow 模型的进行系统静态性能分析及基于系统资源状态的连续时间马尔科夫链 (CTMC) 的系统动态性能分析的方法。并以一个实例验证了本文提出的模型性能分析方法的有效性。

**关键字:** Petri 网, 性能分析, 工作流, 系统配置

### A Method of Workflow Model Performance Analysis

Li Jianqiang Fan Yushun

Dept. of Automation, Tsinghua University, Beijing 100084

**Abstract:** Today's workflow techniques almost did nothing in theoretical analysis of workflow model's performance except gave some limited simulation support to get a few referenced indicators. In this paper, we point out some relevant problems in the workflow analysis, and then extend the WorkFlow nets (WF-nets)<sup>[1]</sup> with resource and time information as the Multi-Dimension WorkFlow nets (MDWF-net). Then a model mapping method from a workflow model built by a graphic process modeling language<sup>[3]</sup> to the MDWF-net is given. According to the characteristics of workflow model for business process, the algorithm to decompose a free-choice and acyclic Petri Net (PN) into a set of *T*-components is extended to a MDWF-net containing iteration structures. Then based on model structure analysis and the theory of open Jackson Network, the methods of how to analyze the performance and the resource configuration of the corresponding workflow model is discussed. The usability of the research results presented in this paper is illustrated by an example.

**Keyword:** Petri nets, Performance analysis, Workflow techniques, System configuration

#### 一、引言

实施 workflow 管理的主要目的是通过合理的调用和分配与活动有关的信息和资源来实现企业业务处理过程的高效、自动化执行<sup>[5]</sup>。workflow 管理系统很好地实现了将企业业务处理过程中活动、信息及资源的调度策略集成到业务的自动协调操作中。但正因为如此, 组成业务处理过程的各活动之间协调逻辑关系、活动执行的资源分配策略及相关时间约束等信息都体现在业务过程的 workflow 模型描述中。因此建立逻辑正确、性能良好及能够可靠执行的 workflow 模型是成功实施 workflow 管理的关键。

反映企业经营过程的 workflow 模型的正确性、有效性、和高性能是模型分析和优化的主要内容, 所以我们把 workflow 模型分析分成三个部分: 模型证实、模型验证、模型性能分析。其

<sup>1</sup> 国家高新技术研究与发展(863)项目(编号:2001AA415020)和国家自然科学基金项目(编号: 60274046)资助。

中模型证实从一般模型分析的角度来讲仅是证实模型完成其功能的有效性,而在模型性能分析中则是需要确认模型是否与要描述的真实世界中系统相一致。模型验证则主要指分析 workflow 模型的正确性(不存在结构上或行为上的死锁或相关视图逻辑的一致性)的静态分析。而模型性能分析一般指通过仿真或严格的理论分析获得和系统性能相关的量化指标(业务平均处理时间、资源平均利用率等)来评估建立的经营过程工作模型是否满足目标需求。显然,关于 workflow 模型的三方面分析需要按一定的顺序进行,即:模型证实→模型验证→模型性能分析。也就是说性能分析是在保证模型正确性的基础上进行的。关于模型证实及模型验证已在文献[3,6,7,8]有很深入的研究。而对 workflow 模型的性能分析,目前除了有限的仿真支持以得到参考数据以外,理论上的严格分析还几乎是空白。

由于 workflow 模型的复杂性,对 workflow 模型的性能分析很难涵盖其描述的所有方面。所以在本文所讨论的性能分析中,主要针对于对模型性能起关键作用的资源和过程两个视图。目前有很多简单、直观且易于在已有的项目管理软件工具上扩展实现的有向网络图可以用来进行 workflow 模型的定义和描述,如: EPCM 模型、PERT 图及活动树等,但因为它们缺乏严格的数学基础和规范化的语义,很难直接对其描述的业务流程 workflow 模型进行验证或性能分析,特别对于复杂经营过程,如存在并发、冲突等情形时。由于 Petri 网具有图形化表达的形式化语义、基于状态的流程描述方式及丰富的模型分析方法<sup>[1]</sup>等特点,它对 workflow 的建模与分析研究人员来讲有很大的诱惑力[1,6-8,11-15]。为了兼顾模型的可读性及可分析性两方面特点,本文提供了一种具有语义不变性的映射方法将一种基于 workflow 联盟过程定义有向网络图模型转换为相同语义结构的高级 Petri 网,进而阐述了如何运用 Petri 网理论并结合业务过程的特点进行相应的性能分析。

本文第二节对文献[2]中提出的基于 WfMC 的过程定义的有向网络图模型和 workflow 网等相关概念进行介绍,并给出了多维 workflow 网的定义。第三节讨论了从这种有向网络图模型到多维 workflow 网的模型映射问题;第四节给出了一种将具有自由选择特性的 workflow 网分解为一组分别对应于每种具体业务实例处理过程的自由选择子网的算法,并对算法的有效性及其复杂性进行了讨论;第五节讲述了如何利用分解得到的自由选择子网集合及 Jackson 排队网络理论对 workflow 模型进行静态性能分析,并给出利用基于资源状态的 CTMC 得到动态性能指标的方法;最后,第六节给出了一个应用实例。

## 二、基本概念

文献[3]给出了一种基于 workflow 管理联盟过程定义的图形化建模语言,其利用两种对象实体进行过程建模(如图 1 所示):节点和有方向的连接弧。其中节点分为两种:任务节点和选择/汇合节点。任务节点用一个方框表示,代表为了实现某种目标或功能而需要做的工作(包括人工型和自动应用),它根据实际的需要细分为:活动、子过程、块、和空活动,而为了方便起见,在模型映射过程中我们将任务节点统称为活动。选择/汇合节点(逻辑节点)用一个圆圈表示,用来表达“或分支”和“或连接”的逻辑结构。连接弧分为有条件连接弧、无条件连接弧和数据连接弧。有条件和无条件连接弧用来描述活动执行时相互间的约束关系,而数据连接弧表示活动间数据的传输方向。由于在这里我们只使用其前后逻辑控制约束功能,所以各种连接弧在这里不再区分。需要指出的是,文献[3]在定义这种图形化建模语言时,不允许循环结构(内嵌在块中)的出现,为了增加其应用范围,本文允许循环结构在 workflow 模型中的显性表达。



图 1. 有向网络图形式的模型定义实体单元

由于这里对工作模型进行性能分析,所以我们需要对工作流的由向网络图模型作相关信息的补充描述。我们增加定义活动的执行时间,为了更具一般性,我们假设每个活动的执行时间服从负指数分布。而对于活动相关资源信息的描述则体现支持活动执行的资源分配中。由于现在的绝大部分 workflow 管理系统都不提供对基于多资源协调的任务执行方式的支持<sup>[1,8]</sup>,在这里我们假设每个活动的执行只需要一种资源的参与,而每个资源个体可以参与不同活动的执行。图 2 是我们补充定义相关信息后的活动描述。活动 A1 的执行需要资源  $R_j$  的参与,其执行时间服从均值为 5 的负指数分布。

A1
[5]/ $R_j$

图 2. 活动执行相关信息补充定义示例

另外,这里还要对我们将要用到的 Petri 网<sup>[4,16]</sup>和 workflow 网<sup>[1]</sup>的一些概念或定理作简单介绍。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  $\bullet t_1 = \bullet t_2$ 。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:  $\varepsilon$  和  $\theta$ , 库所  $\varepsilon$  是一个起始库所,即  $\bullet \varepsilon = \Phi$ ; 库所  $\theta$  是一个终止库所,即  $\theta^\bullet = \Phi$ ;(2) 如果在  $PN$  中加入一个新的变迁  $t$ , 使  $t$  连接库所  $\theta$  与  $\varepsilon$ , 即  $\bullet t = \{\theta\}$ ,  $t^\bullet = \{\varepsilon\}$ , 这时所得到的扩展 workflow 网  $\overline{PN}$  是强连接的。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  $F_1 = F \cap ((S_1 \times T_1) \cup (T_1 \times S_1))$ ; 若  $PN_1$  为  $PN$  的子网,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)时,  $PN_1$  为  $PN$  的变迁构成网; 若  $PN_1$  为  $PN$  的变迁构成网且  $\forall p \in P_1: |\bullet p \cap T_1| \leq 1 \wedge |p^\bullet \cap T_1| \leq 1$ , 则  $PN_1$  被称为  $PN = (P, T, F)$  的变迁组件网 ( $T$ -component)。文献[1]中已给出 workflow 网  $PN$  具有完整性 (Soundness) 的充分必要条件是扩展 workflow 网具有安全性和活性。

很明显, workflow 网只完成了对业务过程逻辑层次的描述,而要进行模型性能分析,我们引入上面提到的资源、时间信息元素到 workflow 网中来,下面我们给出多维 workflow 网的定义。

**定义 1:** 多维 workflow 网为四元组 ( $WF$ -net,  $\alpha, R, M$ ), 其中:

$WF$ -net =  $(P, T, F)$  为 workflow 网;

$P = \{p_1, p_2, \dots, p_m\}$  为库所集合, 其中每个库所  $p$  表业务实例的状态;

$T = \{t_1, t_2, \dots, t_n\}$  为变迁集合, 其中每个变迁  $t$  表业务过程中的活动或事件;

$F$  为连接库所和变迁的有向弧, 表库所与变迁之间的控制依赖关系;

$\alpha$  为正实数  $\alpha(t)$  集合, 每个元素  $\alpha(t)$  表变迁  $t$  的触发时间期望值;

$R$  表资源  $R(t)$  集合, 每个元素  $R(t)$  表变迁  $t$  的触发所需资源;

$M$  为  $m$  维向量集合,  $M(p)$  表库所  $p$  中 Token 的数量;

类似地,我们可以给出扩展多维 workflow 网的定义。显然,有向网络图及这里给出的多维 workflow 网都可以用来描述相应业务处理过程。为了能够应用 Petri 网理论对现在大多数建模工具使用的有向网络图模型进行性能分析,我们首先需要讨论从有向网络图模型到多维 workflow 网之间的映射。

### 三、模型映射

下面我们提供一种映射方法将利用如前所述的图形化过程定义语言所建立的有向网络图映射为多维 workflow 网 (Petri 网): 每个选择/汇合节点  $D$  映射为库所  $p_D$ , 而对应于其任一个输出路径  $i$ , 创建变迁  $t'_D$  作为  $p_D$  的输出变迁; 每个活动  $A$  映射为一个变迁  $t_A$ , 如果活动  $A$  处于顺序结构中, 则创建库所  $p_A$  作为变迁  $t_A$  的输入库所, 而如果活动  $A$  为同步活动, 则对应于每一个输入路径  $j$ , 创建库所  $p'_A$  作为变迁  $t_A$  的输入库所; 对于结束活动  $E$ , 创建终止库所  $\theta$  作为变迁  $t_E$  的输出库所; 而有向网络图中的连接弧则直接映射为 workflow 网相应库所和变迁的连接弧。按以上规则将有向网络图中所有节点和连接弧进行转换, 即可完成从有向网络图到多维 workflow 网之间的拓扑结构映射。

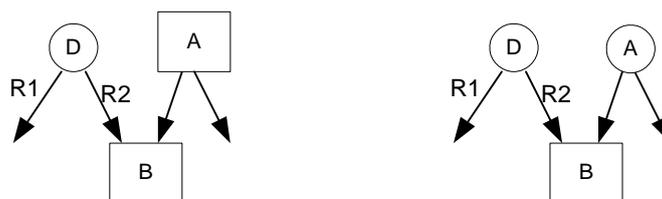


图3.非自由选择结构

由于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由 workflow 控制数据和 workflow 相关数据的协同工作共同来完成,所以现在绝大部分的 workflow 建模工具所建立的 workflow 模型都具有自由选择的语义特性<sup>[1]</sup>。而且,在有向网络图的非自由选择结构中,必然出现选择节点的输出路径中存在输入路径来源于不同选择输出路径的同步活动。图3所示的两个例子中,无论节点  $A$  是选择节点或与分支活动,如果选择节点  $D$  选择了  $R_1$  作为其输出路径,则活动  $B$  必死锁。所以我们假设经过模型验证正确的工作流模型都具有自由选择的特性,从而我们映射得到的 workflow 网模型在结构上一定是自由选择 Petri 网。

完成网络拓扑结构映射之后就要考虑其他模型相关信息的映射问题。在这里,对于每个活动的时间、资源等相关信息直接映射到相应活动映射得到的变迁上。而由于选择/汇合节点只用来表达一种路由逻辑关系,所以其映射得到的变迁时间或资源相关信息都设为零值。对于时间,它意味着该变迁的触发时间为零;而对于资源,零值表示该变迁的触发不需任何资源支持。

至此我们已经完成全部的模型映射过程。如前所述,由于需要性能分析的工作流模型首先要求模型必须是正确的,所以这里映射得到的多维 workflow 网必满足完整性<sup>[1]</sup>。

#### 四、模型分解

我们知道一个 workflow 模型用来描述某一类业务实例的处理过程,而针对于每一种业务实例,workflow 模型中都存在一种路由与其相对应。显然,这种路由在其 Petri 网模型中对应于一个自由选择子网(如果 workflow 网中不存在循环结构则为组件变迁网)。在此启发下,我们提供一种可以将描述一类业务实例的处理过程的多维 workflow 网(其中可能存在循环结构)分解为一组描述单个业务实例处理过程的自由选择子网的方法。

在给出分解算法之前,先对其中要用到的几个重要概念<sup>[19]</sup>进行介绍。 $PN_1 = (P_1, T_1, F_1)$  为  $PN = (P, T, F)$  的子网,  $path = \{t_0, p_1, t_1 \dots p_m, t_m\}$  为变迁路径当且仅当: (1)  $(t_{j-1}, p_j), (p_j, t_j) \in F, 1 \leq j \leq m$  ( $path$  为有向路径); (2)  $path$  为简单路径(在  $path$  中每个元素只出现一次); (3)  $t_0, t_m \in T_1$ ; (4)  $p_j \notin P_1, 1 \leq j \leq m$ , 且  $t_j \notin T_1, 1 \leq j < m$ 。对称地,  $path = \{p_0, t_1, p_1 \dots t_m, p_m\}$  为库所路径当且仅当: (1)  $(p_{j-1}, t_j), (t_j, p_j) \in F, 1 \leq j \leq m$  ( $path$  为有向路径); (2)  $path$  为简单路径(在  $path$  中每个元素只出现一次); (3)  $p_0, p_m \in P_1$ ; (4)  $t_j \notin T_1, 1 \leq j \leq m$  且  $p_j \notin P_1, 1 \leq j < m$ 。 $PN$  中库所  $p$  称为选择库所当且仅当  $|p^\bullet| \geq 2$ 。 $PN$  的子网  $PN_1$  为简单路径或回路(不存在分支结构),且  $p \in P_1$  为选择库所,则  $p$  的选择阶数定义为从库所  $p_i$  到  $p$  的路径中选择库所的个数( $p_i$  指简单路径中的第一个库所或 workflow 网中简单回路中的初始库所  $\epsilon$ )。另外,为了表述方便,我们补充一个新的概念。

**定义 2:**  $R$  为一简单路径(回路)集合,选择库所  $p$  为恰当边界库所当且仅当对  $R$  中的任意简单回路(路径)来说,  $p$  不属于它或者  $p$  在其中选择阶数最小。

由上文我们知道,通过模型映射得到的多维 workflow 网在结构上必满足完整性,故其必具有安全性和活性。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 网都被强连接的变迁组件网所覆盖,并且给出一种由任一变迁生成包含此变迁的变迁组件网的算法。文献[19]给出了一种自由选择 workflow 网到一组变迁组件网的分解算法,然而此种算法只适合于不存在循环结构的工作流网模型。下面我们对文献[19]中的分解算法进行扩展。以使其能够应用到具有循环结构显性表达的工作流网模型  $PN$ 。另外,需要说明的是这里要讨论的模型分解指模型结构分解,所以与每个变迁触发相关的时间、资源等相关信息都直接随变迁带入

到分解得到的子网中来。

### 分解算法 1:

**步骤 1:** 通过在起始库所 $\varepsilon$ 和终止库所 $\theta$ 增加新的变迁 $t$ , 即  $\bullet t = \{\theta\}$ ,  $t \bullet = \{\varepsilon\}$ , 以构建扩展工作流网;

**步骤 2:** 对应于任意选择库所 $p^1 \in P_C = \{p \mid |p^\bullet(\text{在PN中})| \geq 2\}$ , 创建任一经过 $p^1$ 和变迁 $t$ 的简单回路子网 $PN_1 = (P_1, T_1, F_1)$ 。另外,  $P_S = \phi$ ,  $R_1 = \{PN_1\}$ , 其中 $P_S$ 为一选择库所集合,  $R_1$ 为一子网集合。

**步骤 3:** 重复以下步骤直到对应于 $\forall PN_j = (P_j, T_j, F_j) \in R_i$ , 集合 $\Psi_j = \{p \mid p \in P_j \wedge |p^\bullet(\text{在PN中})| \geq 2 \wedge p \notin P_S\}$ 为空。最后得到的子网集合为 $R = R_1 = \{PN_1, PN_2, \dots, PN_n\}$ 。

**3.1:**  $R_2 = R_1$ ,  $PL = \phi$ ;

**3.2:** 对每一简单回路子网 $PN_j = (P_j, T_j, F_j) \in R_2$ , 如果 $\Psi_j = \{p \mid p \in P_j \wedge |p^\bullet(\text{在PN中})| \geq 2 \wedge p \notin P_S\}$ 非空, 将其中具有最小选择阶数的选择库所 $p \in \Psi_j$ 并入到 $PL$ 中, 即 $PL = PL \cup \{p\}$ ;

**3.3:** 对每一库所 $p_k \in PL$ , 如果 $p_k$ 来源于 $PN_k \in R_2$ 的 $\Psi_k$ , 同时 $p_k$ 属于产生于 $PN_m \in R_i$ 的 $\Psi_m$  ( $m \neq k$ )且 $p_k$ 的选择阶数在 $\Psi_m$ 中不是最小的, 则 $PL = PL \setminus \{p_k\}$ ;

**3.4:**  $P_S = P_S \cup PL$ 且对于每一库所 $p_l \in PL$ 进行如下操作:

对于每一 $PN_j = (P_j, T_j, F_j) \in R_i$ , 如果 $p_l \in P_j$ , 则必有变迁集合 $\eta = \{t \mid t \in p_l^\bullet(\text{在PN中}) \wedge t \notin T_j\}$ 为非空, 则对 $\eta$ 中每一变迁 $t$ , 必存在库所路径 $path_1 = \{p_l, t_1, \dots, p_e\}$ , 其中只有 $p_l$ 和 $p_e$ 属于 $PN_j$ 。如果 $p_e$ 存在于从 $\varepsilon$ 到 $p_l$ 的一条有向路径中, 则将 $path_1$ 与 $PN_j$ 合并为 $PN'_j$ , 并用 $PN'_j$ 替代 $R_1$ 中原来的 $PN_j$ ; 否则,  $p_e$ 必存在于从 $p_l$ 到 $\theta$ 的一条有向路径中, 利用 $path_1$ 替换 $PN_j$ 中从 $p_l$ 到 $p_e$ 的有向路径, 并将这个新的子网 $PN'_j$ 添加到 $R_1$ 中 (原来的 $PN_j$ 仍存在);

**步骤 4:** 对每一简单回路子网 $PN_k = \{P_k, T_k, F_k\} \in R$ , 重复以下操作直到遍历每一个具有非空集合 $\{p'' \mid p'' \in t''^\bullet(\text{在PN中}) \wedge p'' \notin P_k\}$ 的变迁 $t'' \in T_k$ : 对于每一个库所 $p_1 \in t''^\bullet \setminus P_k$ , 创建一从 $t'' = t_0$ 到 $t_m \in T_k$ 的变迁路径 $path = \{t_0, p_1, t_1, \dots, p_m, t_m\}$ , 并有 $PN_k = PN_k \cup path$ 。然而, 如果在 $PN_k$ 新并入的变迁路径 $path$ 中存在选择库所, 则所有这些 $PN_k$ 组成 $R'_1$ ,  $R = R \setminus R'_1$ , 且重复如下步骤直到对于 $\forall PN_j = (P_j, T_j, F_j) \in R'_1$ ,  $\Psi'_j = \{p' \mid |p'^\bullet(\text{在PN中})| \geq 2 \wedge p' \in P_j \wedge p' \notin P_S\}$ 为空, 并将最后得到的 $R'_1$ 并入到 $R$ 中。

**4.1:**  $R'_2 = R'_1$ ,  $PL' = \phi$ ;

**4.2:** 对于每一子网 $PN_j = (P_j, T_j, F_j) \in R'_2$ , 如果 $\Psi'_j = \{p' \mid p' \in P_j \wedge |p'^\bullet(\text{在PN中})| \geq 2 \wedge p' \notin P_S\}$ 非空, 每个 $p'$ 必属于从 $t_0$ 到 $t_m$ 的变迁路径 $path$ 。将在 $path$ 中具有最小选择阶数的选择库所 $p' \in \Psi'_j$ 并入到 $PL'$ 中, 即有:  $PL' = PL' \cup \{p'\}$ 。

**4.3:** 对每一库所 $p'_r \in PL'$ , 如果 $p'_r$ 来源于 $PN_r \in R'_1$ 的 $\Psi'_r$ , 同时 $p'_r$ 属于 $PN_s \in R'_2$ 的 $\Psi'_s$  ( $r \neq s$ )且 $p'_r$ 的选择阶数在 $\Psi'_s$ 中不是最小的, 则 $PL' = PL' \setminus \{p'_r\}$ ;

**4.4:**  $P_S = P_S \cup PL'$ , 同时对每一库所 $p'_l \in PL'$ 进行如下操作:

对每一子网 $PN_j = (P_j, T_j, F_j) \in R'_2$ , 如果 $p'_l \in P_j$ , 则变迁集合 $\eta' = \{t' \mid t' \in p'_l^\bullet(\text{在PN中}) \wedge t' \notin T_j\}$ 非空。则对每一变迁 $t' \in \eta'$ , 必存在库所路径 $path'_1 = \{p'_l, t'_1, \dots, p'_e\}$ , 其中只有 $p'_l$ 和 $p'_e$ 属于从 $t_0$ 到 $t_m$ 的变迁路径。如果 $p'_e$ 存在于有向路径 $\{t_0, p_1, t_1, \dots, p'_l\}$ 中, 则合并 $path'_1$ 和 $PN_j$ 为 $PN'_j$ , 并用 $PN'_j$ 替代 $R'_1$ 中原来的 $PN_j$ ; 否则, 用 $path'_1$ 替换变迁路径 $path \in PN_j$ 中从 $p'_l$ 到 $p'_e$ 的部分, 以得到新的子网 $PN'_j$ 并将它添加到 $R'_1$ 中。

**步骤 5:** 删除 $R$ 中每一个子网中的变迁 $t$ 及相应的连接弧 $(\theta, t)$ 和 $(t, \varepsilon)$ 。

这里给出的分解算法1主要是对文献[19]中的算法进行扩展以应用于具有循环结构的工作流模型。而扩展工作流网是作为模型分解的中介而提出的。为了使算法更具有可读性,引入步骤1和5(分别通过增加和删除附加变迁 $t$ )用来构建和消除扩展网。由于 $PN$ 具有完整性,步骤1获得的扩展工作流网必具有活性及有界性。步骤2用来构建经过变迁 $t$ 及任一选择库所的简单回路子网 $PN_1$ ,显然其必包含起始库所和终止库所。

在步骤3的每一轮循环中,子步骤3.2和3.3针对与当前的 $R_2$ 选出 $P_C \setminus P_S$ 中的恰当临界库所。对于每一个包含恰当临界库所的子网 $PN_j$ 中,必存在一个非空的恰当临界库所后续变迁(相对于 $PN$ )集合 $\eta$ 。而对于其中的每一个元素(变迁) $t$ ,必存在起始于 $p_i$ (经过 $t$ )终止于 $PN_j$ 中另一库所 $p_e$ 的库所路径(工作流网的完整性保证了其存在性)。如果 $p_e$ 属于从 $\varepsilon$ 到 $p_i$ 的有向路径中, $path_1$ 必为工作流网中循环结构的一部分,所以将 $path_1$ 合并到原始的 $PN_j$ 中。否则, $p_e$ 在从 $p_i$ 到 $\theta$ 的有向路径中,即 $path_1$ 不属于循环结构的一部分,而是选择结构的一部分。此时,需要用 $path_1$ 替换原始 $PN_j$ 中从 $p_i$ 到 $p_e$ 的部分,从而获得并添加一个新的子网到 $R_1$ 中。应用步骤3可以获得从 $PN_1$ 产生出来的所有子网(每一个都可以最后扩充为对应于一种业务实例处理过程的自由选择子网)。

对于步骤3结束后获得的每一个简单回路子网,重复应用步骤4可以获得相应的自由选择子网,其有效性已在文献[4]给以证明。然而在步骤4中新并入的变迁路径中会存在不属于 $P_S$ 的选择库所,为了保证最后得到所有的自由选择子网,对于新并入的变迁路径应用和步骤3相类似的步骤4.1-4.4,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 $R$ 中,进而再应用步骤4获得其相应自由选择子网。最后,经过步骤5获得的集合 $R = \{PN_1, PN_2, \dots, PN_n\}$ 中的每一个子网对应于一种业务实例在 $PN$ 中的路由。

显然,最坏的情况是分解得到的自由选择子网的数量随此多维工作流网中选择库所的数量成指数增长。然而,虽然我们的算法是非多项式时间算法,但其仍具有很好的实际应用前景。一方面,因为此分解算法中的步骤3和步骤4的重复迭代次数和最多为 $|P_C|$ ,其中 $P_C$ 为模型中选择库所的集合,所以此分解算法可以在多项式步骤内完成。另一方面,由于在步骤3中得到的任一简单回路子网都可以立即对其应用步骤4-5得到相应的自由选择子网集合,从这个意义上讲,此分解算法具有很好的并行性。在多处处理器环境下,这种特性可以在很大程度上提高计算速度。

## 五、模型性能分析

下面我们在多维工作流网分解得到的一组自由选择子网 $R$ 的基础上对多维工作流网描述的工作流模型进行相关性能分析,并在得出相应静态性能指标的基础上,结合系统资源状态CTMC实现模型的动态性能分析。

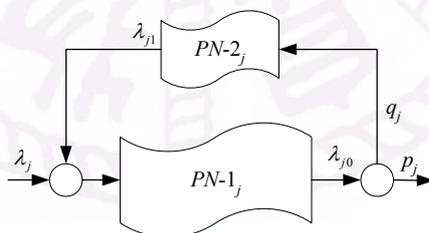


图4. 一个包含 $PN-1_j$ 和 $PN-2_j$ 两个构件的循环结构示例

由于对应于一种业务实例的路由的每个自由选择子网 $PN_i$ 中可能包含循环结构,对于其中属于一个或多个循环结构的变迁来说,完成每个这种业务实例可能要触发多次。为了获得这部分信息,我们引入了相对迭代系数的概念,我们说每一个变迁 $t \in PN_i$ 都有一个相对于自由选择子网 $PN_i$ 的迭代系数 $\Omega_i(t)$ 和一个相对于循环结构(Iterative Control Structure ICS)  $ICS_j \in PN_i^k$ 的迭代系数 $\delta_j(t)$ ,其中 $\Omega_i(t)$ 用来表示为完成到达业务实例的处理,变迁 $t$ 的平均触

发次数,  $\delta_i(t)$ 是作为计算 $\Omega_i(t)$ 的中介而提出的。

下面我们给出计算变迁 $t$ 相对于 $PN_i$ 中每ICS  $ICS_j$ 的迭代系数 $\delta_j(t)$ 的计算方法, 进而最终得到相对于 $PN_i$ 的迭代系数 $\Omega_i(t)$ 。对于不属于ICS  $ICS_j$ 的变迁 $t$ 来说, 它的相对于 $ICS_j$ 的迭代系数为 1, 即 $\delta_j(t)=1$ 。否则,  $t$ 相对于 $ICS_j$ 的迭代系数有它在 $ICS_j$  中的位置决定。

假设自由选择子网 $PN_i$ 包含图 4 所示的循环结构 $ICS_j$ , 其由 $PN-1_j$  和 $PN-2_j$ 两个子网组成。此循环结构的外部业务实例的到达速率为 $\lambda_j$ , 而子网 $PN-1_j$ 和 $PN-2_j$ 的业务实例到达速率分别为 $\lambda_{j0}$ 和 $\lambda_{j1}$ 。每个经过子网 $PN-1_j$ 处理完成后的业务实例路由出此循环结构的概率为 $p_j$  (其进入子网的概率必为 $q_j=1-p_j$ )。显然, 有 $\lambda_{j0}=\lambda_{j1}+\lambda_j$ 及 $\lambda_{j1}=\lambda_{j0}q_j$ , 两个方程联立求解得 $\lambda_{j1}=q_j\lambda_j/p_j$ ,  $\lambda_{j0}=\lambda_j/p_j$ 。则子网 $PN-1_j$  中的变迁相对于 $ICS_j$ 的迭代系数为  $1/p_j$ , 而 $PN-2_j$  中的变迁相对于 $ICS_j$ 的迭代系数为 $q_j/p_j$ 。

若变迁属于 $PN_i$ 中的 $y$ 个循环结构, 而且其相对于每个循环结构 $ICS-x$ 的迭代系数 $\delta_x(t)$ 都已经通过上面的计算方法得到, 则其相对于自由子网 $PN_i$ 的迭代系数为 $\Omega_i(t)=\prod_{x=1}^y \delta_x(t)$ 。显然,

如果 $t \in PN_i$  不属于其中的任何一个循环结构, 它的 $PN_i$  相对迭代系数必为 1, 也就是说完成与此自由选择子网相对应得业务实例的处理过程需要此变迁恰好触发 1 次。

我们假设需要分析的多维 workflow 网模型 $PN$ 包含有 $k$ 个变迁、 $l$ 种资源 $\{Y_1, Y_2, \dots, Y_l\}$  (其中 $Y_i$ 为第 $i$ 种资源的数量), 其描述了 $n$ 种业务实例 $\{I_1, I_2, \dots, I_n\}$  ( $I_i$ 表第 $i$ 种业务实例) 的处理过程。而每种业务实例在总业务实例总量中占有一定的百分比, 既有 $\beta = \{\beta_1, \beta_2, \dots, \beta_n\}$ 满足 $\beta_1+\beta_2+\dots+\beta_n=1$ , 其中 $\beta_i$ 为 $I_i$ 所占的百分比。每种业务实例 $I_i$ 的处理过程包含的变迁及其逻辑关系由相应的自由选择子网 $PN_i$ 表示, 为了清楚地描述一个自由选择子网 $PN_i$  (对应一种业务实例的处理过程) 包含哪些变迁及为完成业务实例处理需要相应变迁的触发次数, 我们构造变迁—自由选择子网矩阵 $B_{n \times k}$ :

$$b_{ij} = \begin{cases} \Omega_i(t_j) & \text{如果 } PN_i \text{ 包含变迁 } t_j \\ 0 & \text{否则} \end{cases}$$

另外, 我们假定此多维 workflow 网所描述的工作流模型的业务实例到达过程为 Poisson 过程且其总速率为 $\lambda$ 。则每种业务实例 $I_i$ 的到达速率为 $\lambda_i = \lambda * \beta_i$ ,  $\lambda_i = \{\lambda_{i1}, \lambda_{i2}, \dots, \lambda_{in}\}$ 。为了完成业务实例 $I_i$ 的处理过程, 对应其在多维 workflow 网路由的自由选择子网 $PN_i$ 中的每一个变迁平均触发 $\Omega_i(t)$ 次。而自由选择子网 $PN_i$ 的业务实例输入速率为 $\lambda_i = \lambda * \beta_i$ , 则对其中的每一个变迁来说, 其业务实例 $I_i$ 的输入速率也为 $\lambda_i$ 。由此可以通过计算 $\lambda_i = \lambda_i * B_{n \times k}$ 得到每个变迁的业务实例总输入速率, 其中 $\lambda_i$ 的第 $i$ 项 $\lambda_{ii}$ 为变迁 $t_i$ 的业务输入速率。同时, 对应于 $k$ 个变迁定义 $\mu = \{\mu_1, \mu_2, \dots, \mu_k\}$ , 其中 $\mu_i$ 为变迁 $t_i$ 的触发延迟均值的倒数, 即 $\mu_i = 1/\alpha(t_i)$ 。另外, 为了描述方便, 这里的时间延迟以某单位时间来衡量, 而业务实例输入速率的单位为“个/单位时间”。

由于时间的性能指标和资源的使用是息息相关的, 所以我们这里把关于时间与资源负载的性能分析放到一起来讨论。在讨论模型性能分析之前, 为了清楚的描述每个变迁的触发需要哪种资源的参与, 同时也是为了计算的方便我们构造变迁—资源矩阵 $C_{l \times k}$ :

$$c_{ij} = \begin{cases} 1 & \text{如果 变迁 } t_j \text{ 的触发需要资源 } Y_i \text{ 的参与} \\ 0 & \text{否则} \end{cases}$$

我们知道, 在分解得到的 $R = \{PN_1, PN_2, \dots, PN_n\}$ 中每一个自由选择子网对应于一种业务实例的处理过程。要完成业务实例 $I_i$ ,  $PN_i$ 中每一个变迁都要平均执行 $\Omega_i(t)$ 次, 这样很容易得到单个业务实例 $I_i$ 的最短平均完成时间为 $Max\{\phi(\gamma)\}$ , 其中 $\gamma$ 为属于 $PN_i$ 中的简单回路,  $\phi(\gamma)$

为简单回路 $\gamma_i$ 中所有变迁 $\Omega_i(t)$ 倍触发延迟时间之和。而对于简单回路 $\gamma_i$ 的计算, 可以通过计算 $PN_i$ 的 $P$ -不变量来获得。

我们这里可以把每一个自由选择子网看成一个简单的Jackson排队网络模型, 其中每个库所可以看成是一个服务等待队列, 而每个变迁表为执行服务服务中心, 其触发所需要的资源 $R_i$ 为相应的服务台, 因此资源 $R_i$ 的个数 $Y_i$ 即为其中平行服务台的个数。而对于同步变迁, 可以通过时间约束或可调度优化使其输入库所中业务实例的相互等待时间尽可能的小。这里我们假设其相互等待时间为0。这样每个服务中心均为单一服务等待队列, 单一服务中心模型即M/M/C队列。

对于由自由选择子网 $PN_i$ 映射出来的Jackson排队网络来说, 初始服务等待队列的顾客(业务实例)到达概率为 $\lambda_{ii}$ , 其他服务等待队列的顾客到达速率均为0。而业务实例在第 $i$ 个变迁(服务中心)以服务规则FIFO接受服务, 其服务时间是均值为 $1/\mu_i$ 的负指数分布, 到达时间和服务时间彼此独立。另外, 若顾客(业务实例)在服务中心 $i$ (变迁 $t_i$ )接受服务后, 如果变迁 $t_j$ 为变迁 $t_i$ 的后序变迁, 则其相应转移概率 $p_{ij}$ 为1, 否则为0, 而顾客在终止库所的输入变迁触发结束后离开系统的概率为1, 而在其他变迁触发结束后离开系统的概率为0。很容易看出, 在这种特殊Jackson网络中, 每个服务中心(变迁)的总顾客输入率可以通过计算 $\lambda_i = \lambda_i * B_{n \times k}$ 得到, 其中 $\lambda_i$ 中的第 $i$ 项 $\lambda_{ii}$ 为变迁 $t_i$ 的业务实例输入速率。

在工作流的实际配置过程中, 每个资源可能参与几个活动的执行。这样就造成在相应的Jackson排队网络中的某些服务台(资源)属于不同的服务中心。前边已经计算得到每个服务中心的顾客到达速率, 由此很容易得到参与 $m$ 个变迁(服务中心)触发的资源 $R_i$ (服务台)的顾客到达速率为 $\omega_i = \sum_m \lambda_{ii}$ , 即为 $\omega = C_{l \times k} * (\lambda_i * B_{n \times k})^T$ 的第 $i$ 项。

由于变迁触发延迟时间的不同造成每个资源 $R_i$ 在参与不同变迁触发时所消耗时间是不同的。参与 $m$ 个变迁触发的资源 $R_i$ 对不同变迁中的业务实例的处理时间(即变迁触发的延迟时间)均服从负指数分布, 但由于其顾客来源于不同的多个变迁, 显然其服务时间不再服从负指数分布, 但通过计算可以得到资源 $R_i$ 处理其到达业务(顾客)的时间均值为 $1 / \sum_m (\mu_i * \lambda_{ii} / \sum_m \lambda_{ii})$ , 即资源 $R_i$ 作为服务台对其顾客的服务速率为 $\eta_i = \sum_m (\mu_i * \lambda_{ii} / \sum_m \lambda_{ii})$ 。

此时对于资源 $R_i$ 的M/M/ $Y_i$ 模型来讲, 如果计算得其服务速率 $\eta_i$ 大于顾客平均到达速率 $\omega_i$ , 则资源 $R_i$ 在系统中的平均利用率为 $\rho_i = \omega_i / (Y_i * \eta_i)$ 。从而可以利用相应的排队网络理论给出该资源 $R_i$ 的平均等待队长 $L_{qi}$ 、业务实例在该资源处的等待时间 $w_i$ 及每个业务实例从初始化到最终完成的平均停留时间等性能指标的解析解。而如果 $\eta_i < \omega_i$ , 则说明此系统是不稳定的, 不存在稳态解。此种情况下, 资源配置显然不能满足系统的基本要求, 从而需要从新进行资源配置(改变资源分配策略或增加资源个数)。

显然, 前面提到的工作流模型性能分析得到的性能指标都是在系统资源状态 $\{Y_1, Y_2, \dots, Y_l\}$ 固定的情况下得到的, 即它只是系统的静态性能指标, 而随着系统的运行, 系统的资源状态必然会发生变化(资源个体的损坏或修复导致可使用资源数量发生变化), 在这种条件下, 可以将系统的资源状态的概率分布考虑到系统的性能分析中来。对于每种资源来讲, 如果我们假设其损坏的速率为 $\lambda$ , 其修复的速率为 $\mu$ , 则可以建立此种资源的Markov状态转移率矩阵 $Q$ , 其中每个状态为相应种类资源的数量个数, 从而利用公式:

$$\begin{aligned} \pi Q &= 0 \\ \sum_i \pi_i &= 1 \end{aligned}$$

得到相应CTMC的稳态分布。而对应于每一个状态, 利用前面的计算方法的到系统的静态性能指标, 综合考虑每种资源状态(当前可用资源个体的数量)的概率分布及相应的系统性

能指标进而得到系统的动态平均性能指标(详见第六节实例)。

另外需要指出的是,虽然第四节给出的从自由选择扩展多维 workflow 网到一组自由选择子网的分解算法为非多项式时间算法,但最坏的情况是它只随此多维 workflow 网中选择库所(并不是所有库所)的数量成指数增长。虽然利用随机 Petri 网理论<sup>[18]</sup>也可以实现 workflow 模型的性能分析,但其不能避开状态空间随系统中的(所有)库所数量成指数增长的问题。而本文提出的按照业务实例处理过程进行模型分解并结合 Jackson 网络理论进行的模型性能分析则可从很大程度上减少状态空间的生长速度(只是随系统中选择库所的数量成指数增长),从而大大提高模型的性能分析速度。

### 六、实例

在如图 5(a)所示有向网络图形式的工作流模型中,开始活动、结束活动及逻辑节点的时间、资源需求等均为零。我们假设此系统的业务实例到达速率为  $\lambda=0.2$ , 而  $\beta=\{\beta_1,\beta_2\}=\{0.5,0.5\}$ , 且业务过程执行的资源  $\{R_1,R_2,R_3,R_4\}$  状态(数量)向量为  $\{1,2,1,1\}$ 。下面我们以此 workflow 模型为例来验证前面给出的性能分析方法的有效性。

图 5(b)为通过模型映射得到的相应自由选择多维 workflow 模型。其中只给出了时间信息的描述(每个变迁的资源需求由其相应活动直接映射得到)。

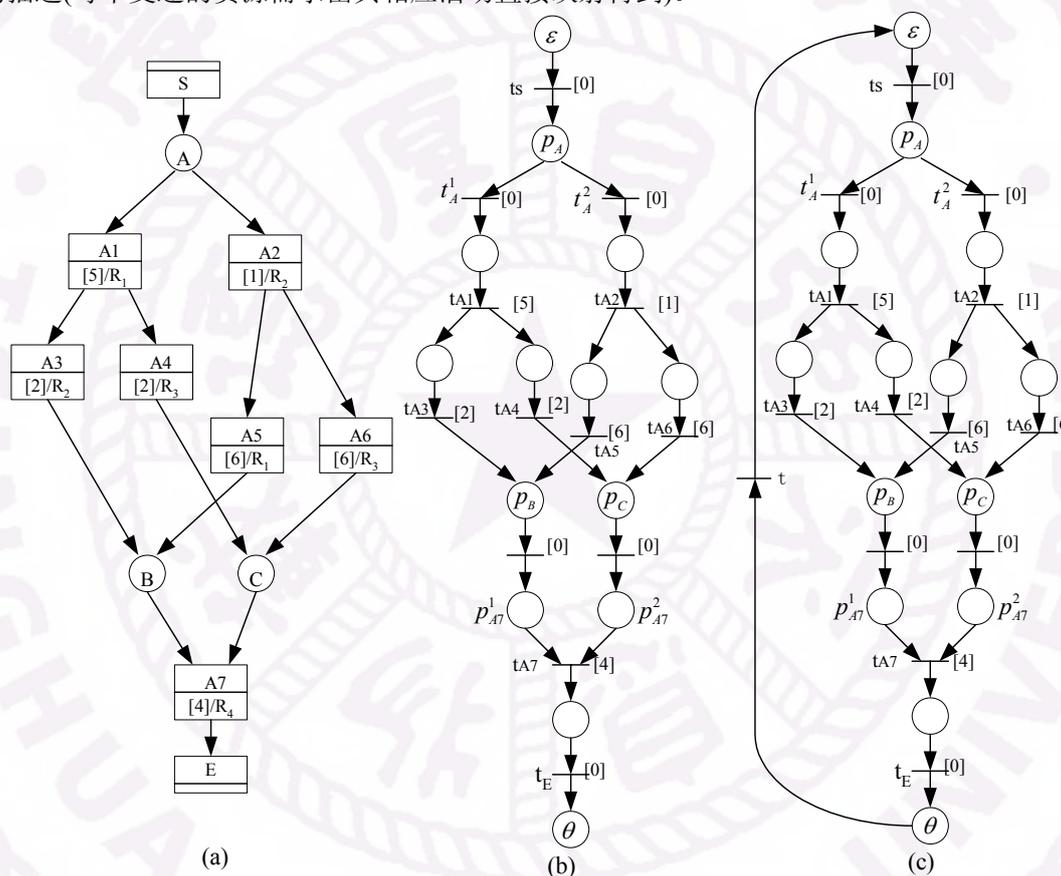


图 5.模型映射实例

图 5(c)和 图 6 则是对图 5(b)中的扩展 workflow 网进行分解的过程示例。其中,图 5(c)中的扩展多维 workflow 网是应用步骤 1 的结果。选取  $p^1=p_A, t=t_A^1$ , 对图 5(c)应用分解步骤 2 得到图 6(a)。图 6(a) 和 6(b) 组成了应用步骤 3 后得到的简单回路子网集合  $R$ 。对  $R$  应用步骤 4 得到图 6(c-d)中两个扩展自由选择子网。最后,步骤 5 将其中的附加变迁去掉之后得到图 6(e-f)中的两个自由选择子网。因此,我们知道此 workflow 模型描述了两种业务实例  $I_1$  和  $I_2$  的处

理过程, 其具体路由分别由图 6(e) 和 6(f) 给予描述。

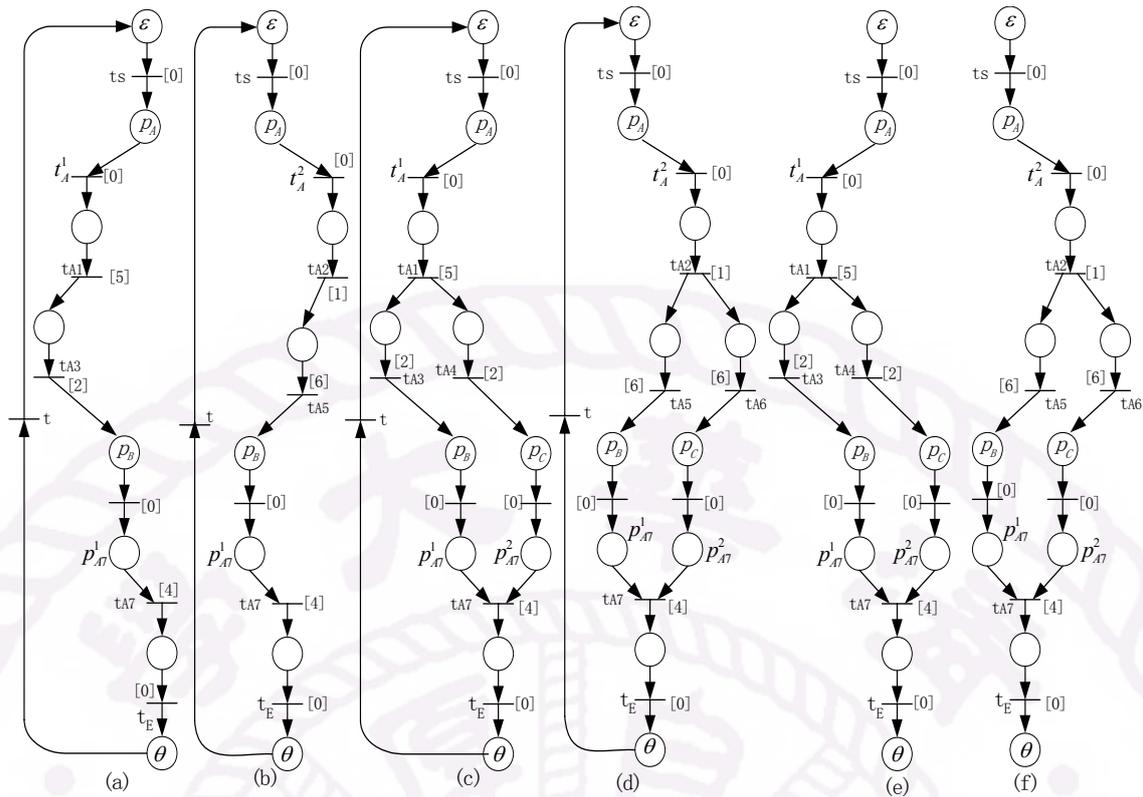


图 5. 模型分解示例

由此, 很容易构造出变迁-自由选择子网矩阵  $B_{2 \times 13}$ , 如下所示:

$$B_{2 \times 13} = \begin{bmatrix} t_s & t_A^1 & t_A^2 & t_{A1} & t_{A2} & t_{A3} & t_{A4} & t_{A5} & t_{A6} & t_B & t_C & t_{A7} & t_E \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} I_1 \\ I_2 \end{matrix}$$

由总业务实例到达速率  $\lambda$  和各业务实例在总业务实例中所占的百分比  $\beta$ , 很容易得到  $\lambda_i = \{0.1, 0.1\}$ , 即  $I_1$  和  $I_2$  的到达速率均为 0.1。

下面讨论其性能分析。对应于多维工作流网中的 13 个变迁可以得到  $\mu = \{\mu_1, \mu_2, \dots, \mu_{13}\} = \{0, 0, 0, 0.2, 1, 0.5, 0.5, 0.167, 0.167, 0, 0, 0.25, 0\}$ , 其中从  $\mu_1$  至  $\mu_{13}$  分别表示变迁  $t_s, t_A^1, t_A^2, t_{A1}, t_{A2}, t_{A3}, t_{A4}, t_{A5}, t_{A6}, t_B, t_C, t_{A7}, t_E$  的触发延迟均值的倒数, 即  $\mu_i = 1/\alpha(t_i)$ 。由图 6(c) 和图 6(d) 很容易看出, 业务实例  $I_1$  和  $I_2$  的最短完成时间都服从均值分别为  $Max\{11, 11\} = 11$  的负指数分布。由  $\lambda_i = \{0.1, 0.1\}$  及  $B_{2 \times 13}$ , 有  $\lambda_i * B_{2 \times 13} = [0.2 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.2 \ 0.2 \ 0.2]$ 。另外, 其变迁—资源矩阵  $C$  为:

$$C_{4 \times 13} = \begin{bmatrix} t_s & t_A^1 & t_A^2 & t_{A1} & t_{A2} & t_{A3} & t_{A4} & t_{A5} & t_{A6} & t_B & t_C & t_{A7} & t_E \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{matrix}$$

从而得到  $\omega = C_{l \times k} * (\lambda_i * B_{n \times k})^T = [0.2 \quad 0.2 \quad 0.2 \quad 0.2]^T$ , 其中的每一项为相应资源的业务实例到达速率。而我们知道资源  $R_1$  参与变迁  $t_{A1}$ 、 $t_{A5}$  的触发, 而变迁  $t_{A1}$ 、 $t_{A5}$  的触发延迟均值分别为 5 和 6, 所以其平均服务速率为  $\eta_i = \sum_m (\mu_i * \lambda_{ii} / \sum_m \lambda_{ii}) = 0.183$ , 同理可得  $R_2$ 、 $R_3$  及  $R_4$  的平均服务速率分别为 0.75、0.333 和 0.25。此时很容易看出, 由于资源  $R_1$  的服务负载  $\rho_1 = \omega_1 / \eta_1 = 0.2 / 0.183 = 1.093 > 1$ , 造成系统不稳定, 其配置显然不满足要求。而如果我们增加其数量至 2,  $\rho_1 = \omega_1 / \eta_1 = 0.2 / (2 * 0.183) = 0.546 < 1$ , 此时资源  $R_1$  的配置满足系统稳定性的要求。同样可以验证其他资源的配置均满足系统稳定性的要求。在资源状态  $\{2, 2, 1, 1\}$  下, 利用排队网络理论可以得到业务实例在每种资源服务的平均性能指标如表 1 所示。

性能指标	资源 $R_1$	资源 $R_2$	资源 $R_3$	资源 $R_4$
资源负载 $\rho$	0.546	0.133	0.601	0.8
空闲概率 $P_0$	0.294	0.765	0.399	0.2
服务队长 $L_s$	1.554	0.271	1.504	4
等待队长 $L_q$	0.464	0.005	0.903	3.2
平均等待时间 $W_q$	2.32	0.025	4.519	4
平均逗留时间 $W_s$	7.77	1.355	7.519	5

进而, 可以得到业务实例  $I_1$  和  $I_2$  在系统中的平均逗留时间分别为 20.289 和 14.125。可以看出, 虽然它们的最短完成时间服从相同的分布, 但由于资源配置的影响, 它们在系统中的平均逗留时间的差别是很大的。另外资源  $R_4$  为系统潜在瓶颈, 系统的最大吞吐率为 0.25。而资源  $R_2$  的利用率则偏低。

显然, 以上只是得到系统的静态性能指标, 下面以资源  $R_2$  为例看一下系统动态性能分析。我们假设其损坏速率为  $\lambda = 0.5$ , 而其修复速率为  $\mu = 0.3$ , 则其状态转移关系如图 7 所示:

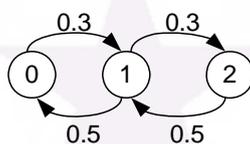


图 7. 资源  $R_2$  的状态转移图

则利用如上所述的 CTMC 的计算公式, 很容易得到  $\pi_0 = 0.51, \pi_1 = 0.306, \pi_2 = 0.184$ 。由此可以得出在每个单位时间内的平均可用资源数量为  $0.306 + 2 * 0.184 = 0.674$ 。在此情况下, 可以计算得到相应资源利用率为  $\rho = 0.396$ 。显然, 其值比静态指标  $\rho = 0.133$  更真实的反应系统的性能及资源配置合理性。

## 七、结论

由于 workflow 技术缺乏坚实的理论基础及其模型的复杂性, 现存的工作流系统对于业务过程模型分析一般只提供有限的仿真支持。运用合适的数学工具对工作流模型进行理论上的验证与分析在工作流技术的研究中越来越受到重视。

由于 Petri 网具有图形化表达的形式化语义、基于状态的流程描述方式及丰富的模型分析方法<sup>[11]</sup>等特点, 已经有很多将 Petri 网技术引入到 workflow 领域中来的尝试性工作<sup>[1, 12-15]</sup>。但其中很多工作<sup>[1, 12, 14, 15]</sup>都是应用 Petri 网对工作流中相关控制流和数据流进行描述以适应由于业务规则或过程的动态性而造成的工作流柔性建模的需要。文献<sup>[13]</sup>提出利用随机 Petri 网 (SPN) 根据系统的状态演化过程进行 workflow 模型的性能分析。但由于系统的状态数量随模型的规模和复杂性的增加而成指数性增长, 当对一个包含十几个或几十个活动的工作流模型进行实际系统性能评价几乎不可能。文献<sup>[18]</sup>介绍了很多 SPN 模型分解与压缩技术, 根据模

型的结构或层次性等特点将系统状态空间进行分割,然后分别对相应子模型进行压缩,以减少需要考察的状态数量,从而分别求解化简子模型以得到整个系统的性能指标。但实际中很多网络结构是难分解和等效压缩的,这种事实造成这些方法具有很大的局限性,而且其不能避开状态空间指数增长问题。

本文在给出了多维工作流网的定义后,根据业务处理过程本身所具有的特性以及大部分过程模型定义工具所建立的工作流模型都有自由选择性的特点,将工作流过程模型映射为自由选择多维工作流网。在此基础上,提出了一种将自由选择工作流网分解为一组自由选择子网的算法,分解得到的每个自由选择子网对应于一种业务实例的处理过程。从而,利用自由选择子网的特点并结合 Jackson 排队网络理论得到整个工作流模型时间性能及资源利用率的静态性能指标及基于 CTMC 的工作流系统的动态性能计算方法。

显然,利用模型结构信息(而不是基于系统的状态演化过程)来进行工作流模型性能分析的方法则可以从很大程度上降低模型性能分析的复杂度(从按所有库所数量指数增长到只随选择库所指数增长)。另外,本文给出的从自由选择 Petri 网到一组自由选择子网的分解算法也是对 Petri 网理论本身的一种丰富。

这里在对工作流模型的性能分析中只考虑到对其最有影响的过程视图和资源视图,但由于企业经营过程的复杂性,一个具有高效性和可靠性的工作流系统还需要定制与过程视图、资源视图相匹配的合理的组织、功能及信息视图。如何在考虑到多过程、多视图相互匹配(尽可能多的信息)的基础上,对工作流模型或业务过程进行集成化分析是我们未来研究的方向。

#### 参考文献

- [1] W.M.P.van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers* 8(1):21-66, (1998).
- [2] Workflow Management Coalition. *Interface 1: Process Definition Interchange, Process Model*. Document Number WfMC TC-1016-P(1998).
- [3] Wasim Sadiq and Maria E. Orlowska. Analyzing Process Models Using Graph Reduction Techniques, *Information Systems*, Vol.25, No.2, pp.117-134, 2000.
- [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] 范玉顺主编. 工作流管理技术基础—实现企业业务过程重组、过程管理与业务过程自动化的核心技术. 北京: 清华大学出版社, 施普林格出版社, 2001年3月.
- [6] W.M.P.van der Aalst. Verification of workflow nets, in *Proceedings of 18<sup>th</sup> international Conference, ICATPN'97*; Toulouse, France; 23-27 Jun 1997, P.Azema and G.Balbo, Eds., *Lecture notes in Computer science, Application and theory of Petri nets 1997*, vol.1248, Springer-Verlag, pp.407-426, 1997.
- [7] 李建强, 范玉顺, 基于 Petri 网模型化简的工作流模型验证, 《信息与控制》将刊登在 Vol.6, 2001
- [8] W.M.P. van der Aalst, Arthur H.M. ter Hofstede. Verification of Workflow Task Structures: A Petri-Net-Based Approach, *Information systems* Vol.25.NO.1, pp.43-69 2000.
- [9] 范玉顺, 吴澄, “工作流管理技术研究及产品现状及发展趋势”, 《计算机集成制造系统 CIMS》, Vol.6, No.1, Jan., 2000, pp.1-7.
- [10] 罗海滨, 范玉顺, 吴澄, “工作流技术综述”, 《软件学报》, 2000年第7期, Vol.11, No.7, pp.899-907.
- [11] W.M.P.van der Aalst. Chapter 10: Three Good Reasons for Using a Petri-net-based Workflow Management System. In T. Wakayama et al. Editor, *Information and Process Integration in Enterprise: Rethinking documents*, The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Norwell pp.161-182 (1998).
- [12] C.A.Ellis and G.J.Nutt, Modeling and enactment of workflow systems, in *Proceedings of the 14<sup>th</sup>*

- International Conference Chicago, Illinois, USA, June 1993, M.A.Marsan,ED., Lecture notes in Computer Science, Application and Theory of Petri Nets 1993 vol.691, Springer-Verlag, pp.1-16,1993.
- [13] A.Ferscha, Qualitative and Quantitative Analysis of Business Workflows using Generalized Stochastic Petri Nets, in Proceeding of CON'94: Workflow management challenges, Paradigms and Products, Linz, Austria, October 19-21, 1994, G.Chroust, A.Benczur(Eds.), pp.222-234, Oldenbourg Verlag, 1994.
- [14] N.R.Adam, V.Atluri, and W.K.Huang, Modeling and Analysis of Workflow Using Petri Nets Journal of Intelligent Information Systems: Special Issue on Workflow and Process Management, M.Rusinkiewicz and S.H.Abdelsalam, Eds., vol.10, no.2, pp.1-29,1998.
- [15] E.Badouel and J.Oliver, Reconfigurable Nets, a Class of High Level Petri Nets Supporting Dynamic Changes within Workflow Systems, Publication Interne IRISA PI 1163,1998.
- [16] Murata. Petri Nets:Properties,Analysis and Applications *Proceeding of the IEEE*, Vol.77.No.4,Appril 1989
- [17] Y. Dallery, Z.Liu, and D.Towsley, "Equivalence, reversibility and symmetry properties in fork/join queueing networks with blocking" *Tech. Rep. MASI 90-32*, Univ.Paris 6,June 1990.
- [18] 林闯, 随机 Petri 网和系统性能评价, 北京: 清华大学出版社 2002 年 1 月.
- [19] 李建强, 范玉顺, " workflow模型时间有界性验证与分析研究", 《计算机集成制造系统 CIMS》, Vol.8, No.10, 2002, pp.770-775.