

## 基于 PTCPN 的网构软件建模与分析\*

周航<sup>1,2+</sup>, 黄志球<sup>1</sup>, 张广泉<sup>4</sup>, 祝义<sup>1,5</sup>, 胡军<sup>1,3</sup>

<sup>1</sup>(南京航空航天大学 信息科学与技术学院, 江苏 南京 210006)

<sup>2</sup>(南京航空航天大学 民航学院, 江苏 南京 210006)

<sup>3</sup>(南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210093)

<sup>4</sup>(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

<sup>5</sup>(徐州师范大学 计算机科学与技术学院, 江苏 徐州 221116)

### Modeling and Analysis of Internetwork Based on PTCPN

ZHOU Hang<sup>1,2+</sup>, HUANG Zhi-Qiu<sup>1</sup>, ZHANG Guang-Quan<sup>4</sup>, ZHU Yi<sup>1,5</sup>, HU Jun<sup>1,3</sup>

<sup>1</sup>(College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

<sup>2</sup>(College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

<sup>3</sup>(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

<sup>4</sup>(School of Computer Science & Technology, Soochow University, Suzhou 215006, China)

<sup>5</sup>(School of Computer Science and Technology, Xuzhou Normal University, Xuzhou 221116, China)

+ Corresponding author: E-mail: jikang524@163.com

**Zhou H, Huang ZQ, Zhang GQ, Zhu Y, Hu J. Modeling and analysis of Internetwork based on PTCPN. *Journal of Software*, 2010,21(6):1254–1266. <http://www.jos.org.cn/1000-9825/3578.htm>**

**Abstract:** Time Petri net can analyze internetwork performance, but it cannot analyze internetwork changing price. To further satisfy the flexible modelling and the changing price analysis demands of internetworks, this paper defines a price time colored Petri net (PTCPN) which provides an extension in the changing price information and color information for the time Petri net. This paper redefines the semantic aspect of price time colored Petri net in terms of priced timed transition systems. A cumulate price state class is defined, and its soundness and completeness are discussed. An approach is proposed to formalize internetwork polymorphism and internetwork control structures based on the price time colored Petri net. Finally, the approach availability is verified with an example. The results show that it is feasible to applying price time colored Petri net to the formal modeling and the analysis of internetwork.

**Key words:** Internetwork; Petri net; price; timed transition system; polymorphism

---

\* Supported by the National Natural Science Foundation of China under Grant No.60873047 (国家自然科学基金); the Aeronautic Science Foundation of China under Grant No.2007ZD52043 (航空科学基金); the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No.20070287052 (高等学校博士学科点专项科研基金); the Jiangsu Provincial Natural Science Foundation of Jiangsu Province of China under Grant No.BK2008154 (江苏省自然科学基金); the Natural Science Foundation for Colleges and Universities of Jiangsu Province of China under Grant No.08KJB520010 (江苏省高校自然科学基金)

Received 2008-04-24; Revised 2008-08-07; Accepted 2009-01-14

**摘要:** 时间 Petri 网能够对网构软件的性能进行分析,但是无法对网构软件的可变代价进行分析.为了进一步满足对网构软件所具有的柔性建模以及可变代价分析的需求,提出一种代价时间有色 Petri 网模型(price time colored Petri net,简称 PTCPN).该模型对时间 Petri 网进行了可变代价和颜色信息的扩展,并使用代价时间变迁系统定义了 PTCPN 的语义;定义了 PTCPN 的累积代价状态类,并证明了累积代价状态类的合理性和完备性;给出了基于 PTCPN 的多 Agent 网构软件建模及分析方法.最后,通过实例分析说明应用 PTCPN 对网构软件进行形式化建模及分析是可行的.

**关键词:** 网构软件;Petri 网;代价;时间变迁系统;多态性

**中图法分类号:** TP311 **文献标识码:** A

由于网络环境的开放与动态性,以及用户使用方式的个性化要求,从而决定了这样一种 Software Web,它应能感知外部网络环境的动态变化,并随着这种变化按照功能指标、性能指标和可信性指标等进行静态调整和动态演化,以使系统具有尽可能高的用户信赖度.我们将具有这种新形态的软件称为网构软件(Internetware)<sup>[1]</sup>.在网构软件中,为了适应环境和用户需求的变化,协同的方式必须灵活多样,可动态调整.这不仅要求软件协同机制能够从软件实体中分离出来,而且应作为相对独立的机制加以研究与实现<sup>[2]</sup>.随着网构软件的发展,使得对网构软件的开发、维护及形式化分析方法研究面临新的课题.

国内外的研究人员对网构软件进行了深入的研究,开发了相应的工具.文献[2]针对开放协同软件模型服务实体与协同部分分离(结构特征)、基于第三方服务实体的协同聚合(开发特征)和协同模式的设计与演化适应环境变化(应变特征)的特征,提出了一种基于 Agent 技术将计算实体与软件协调分离的网构软件模型.文献[3]提出了一种以软件体系结构为中心的网构软件开发方法,该方法采用 ABC(architecture based component composition)方法框架,以提高系统开发的效率和质量.该方法主要解决了 3 个问题:(1) Internet 上无序软件资源的有效建模、组织和管理;(2) 网构软件自适应建模;(3) 支持网构软件特征的支持工具和运行平台.文献[4]针对网构软件实体间信任关系的量化、传递及合并过程进行了合理的抽象,并提出了一种基于自治实体的请求推荐模式来处理较为复杂的信任信息综合过程.模型重点强调对信任关系演化的支持,提出一种反馈学习的方法以支持推荐信任关系的形成与更新,同时考虑了减低恶意推荐信息对最终信任关系量化的影响.文献[5]研究了网构软件的可靠性,提出了一种网构软件的故障恢复方法.上述研究主要在网构软件的开发方法、故障恢复及信任度方面作了深入的探讨,但目前对网构软件的形式化建模和分析验证方面还缺乏相应的研究.随着网构软件采用 Agent 技术,多 Agent 环境决定了计划及运行是分布并行的、计划制定是非唯一的且运行是不确定的.因此,计划、运行的形式化必须能够描述这些分布并行特征和交叉假设问题<sup>[6]</sup>.同时,由于网构软件中的软件实体可以由不同的开发商提供,服务时需要用户支付相应的费用,而用户对价格又是比较敏感的,并且价格也是评价软件质量的重要指标之一,因此需要对网构软件的价格进行建模和分析,以满足用户对网构软件服务的代价要求和网构软件的质量评价需求.如果对网构软件的功能性和代价分开建模将会导致其中单一的模型都无法满足功能和代价两方面的验证分析需求,而重复建模又会导致分析时间及成本大大增加.因此,网构软件研究中一个重要的问题就是如何对网构软件进行统一建模分析,以保证网构软件功能性的同时满足对代价分析的需求.

目前,Z,CSP,CSS,有穷自动机等形式化工具是受到广泛接受的形式化工具,但是它们都是基于交叉语义假设,而 Petri 网有效地克服了交叉假设,适宜描述分布并行计算语义<sup>[7]</sup>.Petri 网作为一个经典的并发模型,已经在协议分析验证、软件验证、业务过程等很多领域得到应用<sup>[8]</sup>.研究人员还为 Petri 网扩展了颜色<sup>[9]</sup>和时间<sup>[10]</sup>等属性,使得 Petri 网的描述和分析能力不断增强.如文献[11]使用时间 Petri 网对 UML 活动图建模并进行实时分析.由于时间 Petri 网只描述了时间特性,因此,使用该模型能够分析网构软件的性能但不能分析网构软件的代价.在为 Petri 网扩展代价信息的研究过程中,文献[12]提出一种代价 Petri 网,为解决 Petri 网的代价分析问题作了初步尝试;文献[13]提出面向网用户应用的代价时间 Petri 网;文献[14]提出了价格时间 Petri 网并解决最小成本可达问题,上述 Petri 网的价格扩展主要针对固定服务价格的扩展.文献[14]虽然在解决最小成本可达问题时提出了累积价格状态类分析法分析,但还存在两点不足:(1) 对变迁实施时间区间分析不够充分,计算价格区间的

方法不够准确.该方法定义 14 中计算价格区间所使用的变迁实施时间区间定义为  $[\alpha_f, \min_{t_j \in enable(M)} \beta_j]$ .如图 1 所示,在初始条件下,变迁  $t_1$  和  $t_2$  都能使.如果变迁  $t_2$  先实施,那么变迁  $t_1$  的延迟时间区间根据文献[14]的定义为  $[1,10]$ ,而实际上,此时变迁  $t_1$  的延迟时间区间应该是  $[2,10]$ ,否则,变迁  $t_1$  有可能先实施,改变变迁序列.因此,文献[14]中计算价格区间的方法不够准确.(2) 由于文献[14]定义的价格是固定价格,无法满足对可变价格进行建模,降低了模型的表达能力.而随着基于时间竞争(time-based competition)<sup>[15]</sup>服务模式的形成,价格和时间开始联系起来(如企业实践表明,消费者愿意为更短的服务响应和处理时间付出更高的代价<sup>[16]</sup>).因此,价格是随时间动态变化的,同时,这种价格变化关系随着网构软件的发展已经反映在网构软件的价格上,如网络服务和电信服务的价格等很多价格都是随时间变化的.所以,使用现有的价格 Petri 网不能准确地对网构软件进行建模和分析.

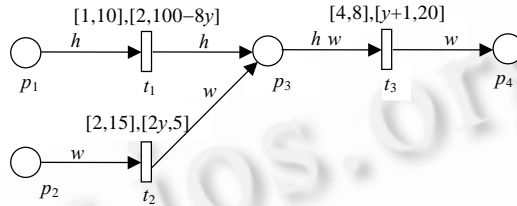


Fig.1 A price time colored Petri net  
图 1 一个代价时间有色 Petri 网

因此,本文在现有 Petri 网相关研究工作的基础上,根据网构软件所具有的柔性特点以及对可变代价的分析验证需求,为时间 Petri 网扩展可变代价和有色信息构成代价时间有色 Petri 网(price time colored Petri net,简称 PTCPN),使得 PTCPN 网能够描述网构软件的多态性、时间约束和可变代价等性质,满足网构软件功能性验证需求的同时,满足对非功能性属性如时间和可变代价的分析需求,有效地降低了形式化建模及验证的成本.

本文提出了基于有色网中的有色集对网构软件实体的多态性建模方法,进而满足了网构软件多态性的建模需求;给出了网构软件多 Agent 技术的 PTCPN 建模方法及相应代价计算方法;最后,通过实例说明基于 PTCPN 的网构软件建模和分析是可行的.

本文第 1 节讨论代价时间有色 Petri 网的语法及语义.第 2 节定义进一步分析状态类的时间区间,给出状态类时间区间的计算公式,并证明计算公式的正确性和完备性.在此基础上,提出 PTCPN 的累积代价状态类,并证明累积代价状态类的合理性和完备性.第 3 节描述网构软件的 PTCPN 建模方法.第 4 节通过实例分析,说明使用 PTCPN 对网构软件进行形式化建模和分析是可行的.第 5 节总结并提出下一步的研究工作.

### 1 代价时间有色 Petri 网的基本定义

20 世纪 80 年代后期,随着基于时间竞争(time-based competition)<sup>[15]</sup>服务模式的形成,价格和时间开始联系起来,而这种价格关系随着网构软件的发展必然会反映到网构软件的价格中去(如网构软件的实体会根据服务响应延迟或服务时间变化收取相应的费用).因此,网构软件的代价不是固定不变而是随时间动态变化的.本文将代价定义为一个与服务时间相关的代价函数,例如  $CP=CA-CB(time)$ ,其中,  $time \in Time, CP \geq 0$ ,表示代价,它随着延迟时间的增加会相应的降低,这里的  $CA, CB$  为常数,由软件实体的提供者直接给出;  $time \in Time$  表示时间变量的取值在相应的延迟时间内.基于时间变化的价格函数,可以由软件实体的提供者根据需要自行定义.

为了能够描述可变代价及多态性等信息,本文在相关 Petri 网研究的基础上,为时间 Petri 网扩展可变代价和有色信息构成 PTCPN,使 PTCPN 能够更准确地描述网构软件的多态性、时间约束和可变代价等性质.

#### 1.1 PTCPN 的语法

定义 1. 多重集  $ms$  被定义为  $\bigcup_{z \in Z} ms(z)z$ , 其中,  $Z$  是一个有限非空有色集,  $ms(z) \in N$  (自然数);  $Z_{MS}$  为  $Z$  集上所有多重集的合集.例如,设有色集  $Z = \{a, b, c\}$ , 它的两个多重集为  $ms_1 = \{a, 2b, c\}, ms_2 = \{2a, c\}$ .

定义 2. 设  $ms_1, ms_2$  属于  $Z_{MS}, ms_1, ms_2$  的运算规则如下:

$ms_1$  加  $ms_2$  表示为  $ms_1 + ms_2$ , 运算规则为  $\bigcup_{z \in Z} (ms_1(z) + ms_2(z))z$ ;  $ms_1$  减  $ms_2$  表示为  $ms_1 - ms_2$ , 运算规则为  $\bigcup_{z \in Z} (ms_1(z) - ms_2(z))z$ ;  $ms_1$  不等于  $ms_2$  表示为  $ms_1 \neq ms_2$ , 判定规则为  $\exists z \in Z: ms_1(z) \neq ms_2(z)$ ;  $ms_1$  小于等于  $ms_2$  表示为  $ms_1 \leq ms_2$ , 判定规则为  $\forall z \in Z: ms_1(z) \leq ms_2(z)$ ; 多重集  $ms_1$  的长度表示为  $|ms_1|$ , 计算规则为  $\sum_{z \in Z} |ms_1(z)|$ .

定义 3. 代价时间有色网(PTCPN)是一个 13 元组,  $PTCPN = (\Sigma, P, T, A, ND, C, G, E, \alpha, \beta, \mu, \nu, IN)$ , 本文定义的风格与文献[9]保持一致. 定义如下:

- $\Sigma$  是类型(type)的非空有限集, 称为颜色集;
- $P$  是有限库所集;
- $T$  是有限变迁集且  $P \cap T = \emptyset$ ;
- $A$  是有限弧集且  $P \cap T = P \cap A = T \cap A = \emptyset$ ;
- $ND$  是节点(node)函数,  $ND: A \rightarrow P \times T \cup T \times P$ ;
- $C$  是着色函数,  $C: P \rightarrow \Sigma$ ;
- $G$  是一个变迁守护函数, 变迁  $t$  守护函数的作用是控制变迁  $t$  的发生权限, 只有在守护函数满足时, 变迁  $t$  才能够实施, 定义为

$$G: T \rightarrow Expression; \forall t \in T: [Type(G(t)) = Bool \wedge Type(Var(G(t))) \subseteq \Sigma],$$

其中,  $Type(G(t)) = Bool$  表示  $G(t)$  的类型为  $Bool$  型,  $Var(G(t))$  表示  $G(t)$  中的所有变量取值属于有色集;

- $E$  是  $A$  上的弧函数, 表示变迁实施时有色集的运算函数, 定义为

$$E: A \rightarrow Expression, \forall a \in A: [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma],$$

$C(p(a))_{MS}$  表示弧连接库所的多重颜色全集, 参见定义 1;

- $\alpha$  表示变迁实施前的最小延迟时间, 定义为  $T \rightarrow R^+$ ;
- $\beta$  表示变迁实施前的最长延迟时间, 定义为  $T \rightarrow (R^+ \cup \{\infty\})$  且  $\alpha \leq \beta$ ;
- $\mu$  是在时间  $[\alpha, \beta]$  区间内变迁使能的代价函数, 定义为

$$\mu: T \rightarrow Expression; \forall t \in T: [Type(\mu(y)) = R \wedge y \subseteq [\alpha, \beta]],$$

表示当变迁使能时, 在时间延迟区间  $[\alpha, \beta]$  内, 由于使能变迁延迟  $y$  个时间单位所消耗的价格. 如图 1 所示: 变迁  $t_2$  的使能价格函数为  $2y$ ,  $y$  的取值范围为  $[2, 15]$ ; 而  $t_3$  的使能价格函数为  $y+1$ ,  $y$  的取值范围为  $[4, 8]$ ;

- $\nu$  是在时间  $[\alpha, \beta]$  区间内变迁使能及实施的代价函数, 定义为

$$\nu: T \rightarrow Expression; \forall t \in T: [Type(\nu(y)) = R \wedge y \subseteq [\alpha, \beta]],$$

表示当变迁在时间区间  $[\alpha, \beta]$  内实施时, 变迁在延迟  $y$  个时间单位后执行的实施价格(这里, 使能及实施时间代价函数是由服务提供者的定价策略来决定的);

- $IN$  是初始化函数,

$$IN: P \rightarrow Expression; \forall p \in P: [Type(IN(p)) = C(p)_{MS} \wedge Var(IN(p)) = \emptyset]$$

为每个库所赋上相应颜色值生成初始标识  $M_0$ .

图 1 表示了一个代价时间有色 Petri 网,  $[1, 10], [2, 100-8y]$  表示变迁  $t_1$  的延迟时间区间  $[\alpha, \beta]$  为  $[1, 10]$ , 使能的代价函数  $\mu$  为常值 2, 实施代价函数  $\nu$  为  $100-8y$  (该函数反映了实施代价随着响应时间的延迟而降低). 变迁  $t_1$  在初始条件下经过 2 个时间单位的延迟后, 实施的总代价为  $2+(100-8y)=86$ , 总代价包括两部分: 1) 变迁  $t_1$  实施延迟 2 个时间单位消耗的使能代价是固定值为 2; 2) 变迁  $t_1$  实施一次消耗的的实施代价为  $100-8y=84$ , 并且随着延迟时间的增加实施价格降低. 从例子可以看出, 在时间延迟下, PTCPN 能够描述变迁的可变使能代价和可变实施代价.

定义 4. PTCPN 的一个标识  $M$  由向量  $(M(p_1), M(p_2), \dots, M(p_n))$  表示, 其中,  $\forall p \in P: M(p) \in C(p)_{MS}$ .

定义 5. PTCPN 中, 变迁  $t$  在标识  $M$  下使能的条件为: 1)  $\forall v \in Var(t): G(t)(b(v)) = true$ , 表示在绑定  $b$  下, 变迁  $t$  的守护函数  $G(t)$  的值为真. 其中,  $b(v) \in C(p), p \in \{p \in P | \exists a \in A: ND(a) = (p, t)\}$ ; 2)  $\forall p \in P(t): E(p, t) \leq M(p)$ , 则称变迁  $t$  在标识

$M$ 下是使能的,记为  $M[t]$ .在标识  $M$ 下所有使能变迁的集合记为  $enable(M)$ .

定义 6. 若变迁  $t$  在标识  $M$ 下是使能的,且变迁  $t$ 的延迟时间  $d(t)$ 满足  $\alpha(t) \leq d(t) \leq \beta(t)$ ,则变迁  $t$ 可以实施,实施的结果为  $\forall P \in \{p \in P \mid \exists a \in A: ND(a) = (p, t) \vee ND(a) = (t, p)\}: M' = M(p) - E(p, t) + E(t, p)$ ,记为  $M[t]M'$ .如果变迁  $t$ 的实施导致变迁  $t_j$ 使能,记为  $t_j \in newly\_en(M, t)$ .

1.2 PTCPN的语义

在时间 Petri 网(TPN)中,一个状态  $s$ 是一个二元组  $s = (M, d)$ ,其中: $M$ 为标识;时间函数  $d: t \rightarrow R^+, d(t)$ 表示变迁  $t$ 在最近一次使能以后延迟的时间.如果变迁  $t$ 在标识  $M$ 下不能使能,则令  $d(t) = 0; d_0$ 是初始状态时的空值,对所有使能的变迁  $t, d_0(t) = 0; d+x$ 表示所有使能变迁延迟的时间都增加了  $x$ 个时间单位.变迁  $t$ 经过延时  $x$ ( $x$ 是非负实数)能从状态  $s(M, d)$ 触发的条件,当且仅当:(1) 存在  $M[t]$ ;(2)  $\alpha(t) \leq d(t) + x \leq \beta(t)$ .

定义 7. 一个代价时间变迁系统  $PTTS^{[17]}$ 是一个二元组  $((S, s_0, \rightarrow), Cost)$ ,其中: $(S, s_0, \rightarrow)$ 是一个时间变迁系统; $Cost$ 是一个从  $\rightarrow$ 映射到  $R^+$ 的代价函数,并且代价可以累加.

例如:一个代价为  $Cost(s \xrightarrow{k} s')$ 的变迁  $s \xrightarrow{k} s'$ 记作  $s \xrightarrow{k, ep_k} s'$ ,其中,  $ep_k = Cost(s \xrightarrow{k} s')$ .当  $s \xrightarrow{k} s', s' \xrightarrow{k'} s''$ 时,总的代价  $ep_{k,k'} = ep_k + ep_{k'}$ .

定义 8. 为了讨论 PTCPN 网的语义,这里引入价格时间变迁系统的概念<sup>[17]</sup>.一个 PTCPN 网的语义可以定义为一个代价时间变迁系统  $PTTS_{PTCPN} = ((S, s_0, \rightarrow), Cost)$ ,其中: $s_0 = (M_0, d_0)$ 初始状态; $Cost$ 是一个从  $\rightarrow$ 映射到  $R^+$ 的代价函数,表示变迁的代价.  $\rightarrow \in S \times (R^+ \cup T) \times R^+ \times S$ 为变迁关系,包括离散的和连续的变迁关系:

(1) 连续的变迁关系

$$\forall x \in R^+, (M, d) \xrightarrow{x, ep_x} (M, d+x) \text{ iff } \begin{cases} \forall t \in enable(M): (d(t) + x) \leq \beta(t) \\ ep_x = Cost((M, d) \xrightarrow{x} (M, d+x)) = \sum_{\forall t \in enable(M)} \mu_t(d(t) + x) - \sum_{\forall t \in enable(M)} \mu_t(d(t)) \end{cases}$$

(2) 离散的变迁关系

$$\forall t \in T, (M, d) \xrightarrow{t, ep_t} (M', d') \text{ iff } \begin{cases} t \in enable(M): \alpha(t) \leq d(t) \leq \beta(t) \\ \forall t' \in T, d'(t') = \begin{cases} 0, & \text{if } t' \in newly\_en(M, t) \\ d(t'), & \text{otherwise} \end{cases} \\ ep_t = Cost((M, d) \xrightarrow{t} (M', d')) = v_t(d(t)) \end{cases}$$

该语义模型表达了 PTCPN 具有 Merlin 模型<sup>[18]</sup>的如下性质:(1) 所有变迁使能的延迟时间不会超过该变迁的时间上界;(2) 新使能变迁的实施时间间隔由该变迁的最早和最迟实施时间决定;(3) 变迁的实施不考虑时间消耗.该语义模型为离散和连续的变迁关系分别联系了一个成本函数,这保证了:(1) 消耗的使能成本为所有使能变迁的使能成本之和;(2) 任何变迁每实施一次所消耗的成本等于在延迟时间  $d(t)$ 下该变迁相应的实施价格.

一个 PTCPN 网的行为被定义为一个时间变迁序列  $\delta = (t_1, d_1, ep_1)(t_2, d_2, ep_2) \dots (t_k, d_k, ep_k)$ ,其中,  $(t_k, d_k, ep_k)$ 表示变迁  $t_k$ 在延迟  $d_k$ 个时间单位后实施.  $\sum_{i=1}^k d_i$ 为  $\delta$ 实施后总的延迟时间,  $\sum_{i=1}^k ep_i$ 为  $\delta$ 实施后总的代价.如果有一个时间变迁序列  $\delta$ 能从标识  $M_0$ 导出标识  $M_i$ ,则称标识  $M_i$ 是可达标识.

例如图 1 中,变迁序列  $(t_1, 2, 86)(t_2, 4, 17)(t_3, 4, 25)$ 实施后到达终止状态时,总延迟时间为 10,总代价为 128;同样地,变迁序列  $(t_2, 2, 9)(t_1, 3, 62)(t_3, 4, 25)$ 实施后到达终止标识时,总延迟时间为 9,总代价为 96.这两个变迁序列实施后总代价不同的原因是变迁  $t_1$ 和  $t_2$ 的实施延迟的时间不同.

定义 9. 一个 PTCPN 网可接受的时间语言为它所有的时间变迁序列的集合.

定义 10. 一个  $A$ 类 Petri 网可接受的所有语言都能够被一个  $B$ 类 Petri 网所接受,则称  $A$ 类 Petri 网是  $B$ 类 Petri 网的子类<sup>[19]</sup>.

定理 1. 时间 Petri 网(TPN)是代价时间有色 Petri 网(PTCPN)的子类.

证明:设  $\psi$ 是一个  $TPN = (P, T, B, F, \alpha, \beta, m_0)$ ,  $\psi$ 的时间变迁系统为  $TTS_{TPN} = (S, s_0, \rightarrow)$ ,  $\psi$ 能接受的时间语言为  $L; \psi'$ 是一个  $PTCPN = (\Sigma, P, T, A, ND, C, G, E, \alpha, \beta, \mu, v, IN)$ ,  $\psi'$ 的代价时间变迁系统为  $PTTS_{PTCPN} = ((S', s'_0, \rightarrow'), Cost)$ ,  $\psi'$ 能够接

受的代价时间语言为  $L'$ . 在  $\psi$  中对资源没有分类只用  $token$  进行表示, 而根据定义 1、定义 2 可得, 有色集只是对  $token$  分类, 则在  $\psi$  中的任意一个标识  $M$  都可以在  $\psi'$  中找到相对应的分类标识  $M'$  表示; 而变迁系统的状态  $S=(M, d), S'=(M', d)$ , 所以  $S$  可以转换为  $S'$ . 同时, 在  $\psi$  中  $B$  定义为  $T \times P$  上的向后关联函数,  $F$  定义为  $P \times T$  上的向前关联函数. 根据定义 3 可得,  $ND$  是节点函数  $ND: A \rightarrow P \times T \cup T \times P$ , 且  $E$  是  $A$  上的弧函数. 显然,  $B, F$  可以等价转换为  $A, ND$  和  $E$ , 则  $TTS_{TPN}$  中的变迁关系  $\rightarrow$  可以等价转换为  $PTTS_{PTCPN}$  中的变迁关系  $\rightarrow'$ . 因此, 对于时间变迁系统为  $TTS_{TPN}=(S, S_0, \rightarrow)$ , 加入了相应的价格函数就可以得到  $PTTS_{PTCPN}=(S', S'_0, \rightarrow'), Cost$ ; 并且在  $L$  中任意一个时间变迁序列  $\delta=(t_1, d_1)(t_2, d_2) \dots (t_k, d_k)$ , 根据定义 8 可得, 在  $L'$  中唯一对应一个代价时间变迁序列  $\delta'=(t_1, d_1, ep_1)(t_2, d_2, ep_2) \dots (t_k, d_k, ep_k)$ . 所以,  $\psi'$  能接受时间语言  $L$ . 根据定义 10 可得, TPN 是 PTCPN 的子类.

## 2 累积代价状态类

状态类方法是 Petri 网分析中解决状态空间爆炸的主要方法之一. 在时间 Petri 网的状态类方法研究方面, 文献[18]提出了能够保持线性时间性质的时间 Petri 网状态类方法. 文献[20]提出了可以保持分支时间性质的强原子状态类方法. 在此基础上, 文献[14]提出一种扩展了累积成本的计价状态类方法, 该文献的定义 14 中计算价格区间所使用的变迁实施时间区间定义为  $[\alpha_j, \min_{t_j \in enable(M)} \beta_j]$ . 如图 1 所示, 在初始条件下, 变迁  $t_1$  和  $t_2$  都使能, 如果变迁  $t_2$  先实施, 那么变迁  $t_1$  的延迟时间区间根据文献[14]定义为  $[1, 10]$ , 而实际上, 此时变迁  $t_1$  的延迟时间区间应该是  $[2, 10]$ , 否则变迁  $t_1$  有可能先实施, 改变变迁序列. 因此, 文献[14]中计算价格区间的方法不够准确. 同时, 目前状态类分析法中的实施域  $D$  是通过时间变量  $\theta$  定义的, 由于  $\theta$  是相对当前状态下的实施时间, 而变迁  $t$  可能在标识  $M$  和  $M'$  下持续使能, 即  $t \in enable(M, M')$  (文献[14]定义 12), 因此,  $\theta$  并不一定是变迁  $t$  的绝对实施时间 (这里的绝对实施时间是相对于变迁  $t$  成为新使能的时刻来说的). 显然, 为每个状态定义一个不同时间代价函数是不合实际的, 一般都是以开始使能为时间起点定义一个统一时间代价函数. 如图 1 所示, 如果请求服务的资源 (有色集) 为  $2h$ , 变迁  $t_1$  在延迟 2 个时间单位后执行, 此时消耗的实施价格为  $100-2 \times 8=84$ ; 在变迁  $t_1$  执行过一次的标识状态下, 变迁  $t_1$  仍然持续使能, 如果再延迟 1 时间单位执行变迁  $t_1$ , 由于  $\theta$  是相对当前标识的时间, 所以此时消耗的实施价格为  $100-1 \times 8=92$ . 而事实上, 第 2 个请求资源总共延迟了 3 个时间单位, 实际消耗的实施价格应该为  $100-3 \times 8=76$ . 所以, 不能使用  $\theta$  计算变迁  $t$  在延迟后的代价, 也就无法计算累积代价.

因此, 本文提出了状态类延迟时间区间的计算方法并证明了该方法的正确性, 并在此基础上给出了一种累积代价状态类方法, 用于可变代价的累积计算. 累积代价计算方法的主要思想是: 根据状态类的时间约束、使能代价函数和实施代价函数计算出相应变迁在延迟区间内实施的代价区间; 然后, 通过给出的公式计算出从初始状态到达该状态类的累积代价区间.

**定义 11.** 一个代价区间  $\varphi=[a, b]$ , 其中,  $a, b \in R^+$  且  $a \leq b$ . 设  $\varphi_1=[a_1, b_1], \varphi_2=[a_2, b_2], Const$  为常数且  $Const \in R^+$ , 则  $\varphi_1 + \varphi_2 = [a_1 + a_2, b_1 + b_2], \varphi_1 + Const = [a_1 + Const, b_1 + Const], Const \cdot \varphi_1 = [Const \cdot a_1, Const \cdot b_1]$ .

**定义 12.** 当  $M[t_i]M'$  时, 若  $t_j$  在标识  $M$  和  $M'$  都使能, 则  $t_j \in enable(M)$  且  $t_j \in enable(M')$ , 记为  $t_j \in enable(M, M')$ .

**定义 13.** 当  $M[t_i]M'$  时, 在状态类  $(M', D')$  下, 所有新使能的变迁集所能接受的最大延迟为  $\eta'$ ; 在标识  $M$  和  $M'$  都使能的变迁集所能接受的最大延迟为  $\omega'$ , 其定义如下:

$$\eta' = \min_{t_k \in newly\_en(M', t_i)} (\beta_{t_k}), \omega' = \min_{t_k \in enable(M, M')} (\beta_{t_k}),$$

其中,  $D'$  是状态类的实施域<sup>[15]</sup>. 实施域定义中的不等式有两种类型:

$$\begin{cases} \forall t_j \in enable(M'): \alpha_j \leq \theta_j \leq \beta_j \\ \forall t_m, t_n \in enable(M') \wedge m \neq n: -\gamma_{mn} \leq \theta_m - \theta_n \leq \gamma_{mn} \end{cases}$$

这里,  $\theta_j$  是使能变迁  $t_j$  在当前状态类下的实施时间,  $-\gamma_{mn}$  是变迁  $t_m$  和  $t_n$  实施时间差的最小可能值,  $\gamma_{mn}$  是变迁  $t_m$  和  $t_n$  实施时间差的最大可能值. 因此, 一个状态类包括了两个变迁实施之间的所有可达状态. 一个状态类中的所有状态有着同样的标识  $M'$ , 并且通过一组不等式  $D'$  定义了标识  $M'$  下使能变迁  $t_j$  实施时间  $\theta_j$  的约束关系. 由于  $\theta_j$  是相对当前状态下的实施时间, 不能使用  $\theta_j$  计算变迁  $t_j$  在延迟后的实施代价. 下面给出了变迁  $t_j$  作为新使能到实施时绝对时间区间的定义.

**定义 14.** 当  $M[t_i]M'$ , 且  $y_i$  为变迁  $t_i$  的延迟区间时, 在状态类  $(M', D')$  下, 一个变迁  $t_j$  可实施的延迟区间为  $y_j$ , 其定义如下:

- (1) 当  $t_j \in \text{newly\_en}(M, t_i)$  时,  $y_j = [\alpha_{t_j}, \min(\eta', \omega' - \min(y_i))]$ ;
- (2) 当  $t_j \in \text{newly\_en}(M, M')$  时,  $y_j = [\max(\min(y_i), \alpha_{t_j}), \min(\eta' + \min(y_i), \omega')]$ .

**定理 2.** 当  $M[t_i]M'$  时, 在状态类  $(M', D')$  下, 一个变迁  $t_j$  可实施的延迟区间肯定在区间  $y_j$  内.

**证明:** 首先证明当  $t_j \in \text{newly\_en}(M, t_i)$  时,  $y_j = [\alpha_{t_j}, \min(\eta', \omega' - \min(y_i))]$ . 采用反证法, 先设新使能的变迁  $t_i$  延迟实施的时间为  $x$ . 假设  $x < \alpha_{t_j}$ , 因为定义 6 规定  $x \geq \alpha_{t_j}$ , 所以假设与定义 6 矛盾, 假设不成立. 假设  $x > \min(\eta', \omega' - \min(y_i))$ , 该假设存在两种可能:

$x > \eta'$ . 根据定义 13 可得  $\eta' = \min_{t_k \in \text{newly\_en}(M', t_i)}(\beta_k)$ , 则  $\eta'$  为所有新使能的变迁集所能接受的最大延迟. 如果  $x > \eta'$ , 根据 PCTPN 的强制触发性可得, 必定存在新使能的变迁  $t_k (k \neq j)$  先实施, 即  $M'[t_k]$ , 而定义 14 的定义为在标识  $M'$  实施的变迁应为  $t_j$ , 即  $M'[t_j]$ . 显然, 假设  $x > \eta'$  时与定义 14 矛盾, 假设不成立.

$x > \omega' - \min(y_i)$ , 根据定义 13 可得  $\omega' = \min_{t_k \in \text{enable}(M, M')}(\beta_k)$ , 即  $\omega'$  为在标识  $M$  和  $M'$  都使能的变迁集所能接受的最大延迟,  $\min(y_i)$  表示在标识  $M$  状态下变迁  $t_i$  实施的最小延迟时间. 如果  $x > \omega' - \min(y_i)$ , 根据 PCTPN 的强制触发性可得, 必定存在一个在标识  $M$  和  $M'$  都使能的变迁  $t_k (k \neq j)$  先实施, 即  $M'[t_k]$ , 而定义 14 的定义为在标识  $M'$  实施的变迁应为  $t_j$ , 即  $M'[t_j]$ . 显然, 假设  $x > \omega' - \min(y_i)$  与定义 14 矛盾, 假设不成立. 所以, 假设  $x > \min(\eta', \omega' - \min(y_i))$  不成立, 则当  $t_j \in \text{newly\_en}(M, t_i)$  时, 变迁  $t_j$  的延迟区间  $y_j = [\alpha_{t_j}, \min(\eta', \omega' - \min(y_i))]$  内. 同理可证, 即当  $t_j \in \text{newly\_en}(M, M')$  时, 变迁  $t_i$  的延迟区间  $y_j = [\max(\min(y_i), \alpha_{t_j}), \min(\eta' + \min(y_i), \omega')]$  内. 定理 2 成立.

**定义 15.** 在状态类  $(M, D)$  下, 一个变迁  $t_i$  的实施区间为  $y_i$ , 且变迁  $t_i$  执行的代价由两部分组成: 一是从使能到执行时消耗的使能代价  $\mu_{t_i}(y_i)$ , 二是变迁  $t_i$  在相应时间执行时的实施代价  $v_{t_i}(y_i)$ . 因此, 一个变迁  $t_i$  实施的代价区间  $TP_{t_i}$  定义为

$$TP_{t_i} = [\min(\mu_{t_i}(y_i) + v_{t_i}(y_i)), \max(\mu_{t_i}(y_i) + v_{t_i}(y_i))].$$

**定义 16.** PTCPN 网的累积代价状态类  $SPC$  是一个三元组  $(M, D, SP)$ , 其中,  $M$  是一个标识,  $D$  是状态类的实施域,  $SP = [a, b]$  是状态类的累积代价区间.

**定义 17.** PTCPN 的累积代价状态类图  $SPCG$  是一个变迁系统  $TSE_{SPCG} = (SPC, spc_0, \rightarrow)$ , 其中:  $SPC$  为状态类集合;  $spc_0$  是初时状态, 为  $(M_0, D_0, [0, 0])$ ;  $\rightarrow \in SPC \times T \times SPC$ , 表示状态间变迁关系, 定义如下:

$$(M, D, SP) \xrightarrow{t_i} (M', D', SP') \text{ iff } \begin{cases} t_i \in \text{enable}(M) \\ M[t_i]M' \\ D' = \text{next}(D, t_i) \\ SP' = SP + TP_{t_i} \end{cases}$$

这里,  $\text{next}(D, t_i)$  为变迁  $t_i$  实施后的下个区域, 详细定义见文献[19]. 为了计算累积代价, 本文通过一个变迁系统定义了累积代价状态类  $SPC$  间的变迁关系, 以及相应变迁关系的累积代价计算方法. 在状态  $(M, D, SP)$  下, 变迁  $t_i$  实施后到达状态  $(M', D', SP')$ , 此时, 累积代价  $SP'$  等于前一个状态类  $SPC$  的累积代价  $SP$  加上执行变迁  $t_i$  所消耗的代价  $TP_{t_i}$ . 当出现循环迭代时, 由于循环次数事先无法确定, 因此可以根据不同的代价评估需求对循环次数取一次、最大次或平均次进行计算.

例如如图 1 中, 在初始标识下,  $SP_0 = [0, 0]$  且变迁  $t_1$  和  $t_2$  都使能. 如果变迁  $t_2$  先实施, 根据定义 14 可得, 变迁  $t_2$  实施的延迟区间为  $y_1 = [2, 10]$ , 否则, 变迁  $t_1$  将实施; 根据定义 15 可得

$$TP_1 = [\min(2y_1 + 5), \max(2y_1 + 5)] = [9, 25],$$

根据定义 17 得

$$SP_1 = SP_0 + TP_1 = [9, 25].$$

由于  $t_1 \in \text{enable}(M_0, M_1)$ , 根据定义 14 可得, 变迁  $t_1$  实施的延迟区间  $y_2 = [2, 10]$ , 则

$$TP_2 = [\min(100 - 8y_2 + 2), \max(100 - 8y_2 + 2)] = [22, 86],$$

$$SP_2=SP_1+TP_{t_2}=[9,25]+[22,86]=[31,111],$$

由于  $t_3 \in \text{newly\_en}(M_1, t_1)$ , 根据定义 14 可得, 变迁  $t_3$  实施的延迟区间  $y_3=[4,8]$ , 则

$$TP_{t_3}=[\min(y_3+1+20), \max(y_3+1+20)]=[25,29],$$

根据定义 17 得  $SP_3=SP_2+TP_{t_3}=[31,111]+[25,29]=[56,140]$ .

**定理 3.** 定义 17 是合理的, 即任意一个由  $(M_0, D_0, [0,0])$  可达的状态类  $(M_n, D_n, SP_n)$  的  $SP_n$  一定是一个代价区间.

**证明:** 使用数学归纳法, 设从  $(M_0, D_0, [0,0])$  到  $(M_n, D_n, SP_n)$  的一个变迁序列为  $t_0 t_1 \dots t_n$ . 当  $n=0$  时,  $(M_n, D_n, SP_n) = (M_0, D_0, [0,0])$ , 其代价为  $[0,0]$ . 根据定义 11 可得,  $[0,0]$  是一个代价区间, 定理成立; 假设在  $n \leq k$  的情况下定理成立, 那么  $SP_k$  是代价区间; 当  $n=k+1$  时, 根据定义 17 可得  $SP_{k+1} = SP_k + TP_{t_{k+1}}$ , 并且由定义 15 可得  $TP_{t_{k+1}}$  是一个代价区间. 又因为  $SP_k$  是一个代价区间, 根据定义 11 可得,  $SP_{k+1} + TP_{t_{k+1}}$  是一个代价区间, 即  $SP_{k+1}$  是一个代价区间. 可以证明定理 3 成立.

**定理 4.** 定义 17 是完备的, 即从  $(M_0, D_0, [0,0])$  到达  $(M_n, D_n, SP_n)$  的任意一个变迁序列的总实施代价必定在区间  $SP_n$  内.

**证明:** 使用数学归纳法, 设从  $(M_0, D_0, [0,0])$  到  $(M_n, D_n, SP_n)$  的一个变迁序列  $t_0 t_1 \dots t_n$  实施的总代价为  $SUM_n$ . 当  $n=0$  时, 总代价  $SUM_0=0$ , 根据定义 16 可得代价区间为  $SP_0=[0,0]$ . 显然,  $SUM_0$  在区间  $SP_0$  内, 定理成立; 假设当  $n \leq k$  时定理成立, 即  $SUM_k$  在区间  $SP_k$  内, 当  $n=k+1$  时, 根据定义 15 可得  $TP_{t_{k+1}}$  为变迁  $t_{k+1}$  实施所消耗的代价区间, 那么变迁  $t_{k+1}$  实际消耗的代价  $ep_{k+1}$  必定在区间  $TP_{t_{k+1}}$  内. 又因为  $SUM_{k+1} = SUM_k + ep_{k+1}$ , 所以  $SUM_{k+1}$  肯定在代价区间  $SP_k + TP_{t_{k+1}}$  内. 由定义 17 可得  $SP_{k+1} = SP_k + TP_{t_{k+1}}$ , 则  $SUM_{k+1}$  肯定在代价区间  $SP_{k+1}$  内, 定理 4 成立.

### 3 基于 PTCPN 的网构软件建模

#### 3.1 计算单元的 PTCPN 建模

网构软件中的计算单元是实现软件功能的基本单位, 它们能够通过接口进行应用间的交互. 因此, 本文把计算单元表示为一个带有输入和输出库所的 PTCPN 模型, 通过输入库所  $p_{in}$  及输出库所  $p_{out}$  所描述计算单元的接口信息, 通过变迁  $t_r$  描述计算单元的功能, 其定义如下:

**定义 18.** 计算单元  $R$  是一个 PTCPN,  $R=(\Sigma, P, T, A, ND, C, G, E, \alpha, \beta, \mu, \nu, IN)$ , 其中:  $\Sigma = \{\text{类型的集合}\}$ ;  $P = \{p_{in}, p_{out}\}$ ;  $T = \{t_r\}$ ;  $A = \{a_{in}, a_{out}\}$ ;  $ND = \{ND(a_{in}) = (p_{in}, t_r), ND(a_{out}) = (t_r, p_{out})\}$ ;  $C = \{C(p_{in}) = \text{输入类型}, C(p_{out}) = \text{输出类型}\}$ ;  $E = \{E(a_{in}) = \text{基于输入类型的弧函数}, E(a_{out}) = \text{输出弧函数}\}$ ;  $G$  为  $R$  相应的守卫函数;  $\alpha$  为变迁  $t_r$  最小延迟时间;  $\beta$  为变迁  $t_r$  最大延迟时间;  $\mu$  表示变迁  $t_r$  是使能代价函数;  $\nu$  表示变迁  $t_r$  的实施代价函数;  $IN$  为初始函数. 如图 2 所示.

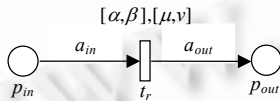


Fig.2 A PTCPN model of software entity  
图 2 软件实体 PTCPN 模型

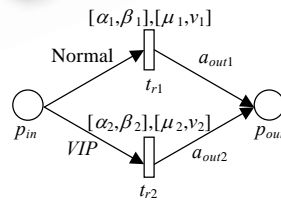


Fig.3 A PTCPN model of polymorphic software entity  
图 3 多态性的软件实体 PTCPN 模型

多态性是指网构软件系统的效果体现出相容的多目标性, 它使得网构软件系统在网络环境下具备了一定的柔性和满足个性化需求的能力. 多态性是网构软件具有区别于传统软件形态的独有的基本特征之一<sup>[1]</sup>, 多态性意味着软件实体具有多种形态的能力. 多态性软件实体运行时, 将根据用户的动态需求, 绑定并执行软件实体中的相应方法. 这种在运行时才能确定绑定类型的多态性方式称为动态绑定(dynamic binding). 为了更准确地对网构软件进行形式化分析, 就必须对网构软件的多态性进行形式化建模, 其 PTCPN 模型定义如下:

**定义 19.** 具有多态性的计算单元  $R=(\Sigma, P, T, A, ND, C, G, E, \alpha, \beta, \mu, \nu, IN)$ , 其中:  $\Sigma$  是  $\{\text{类型集} \cup \text{多态性集}\}$ ;  $P = \{p_{in},$



$p_{out}\}; T=\{t_{r1}, t_{r2}\}; A=\{a_{in1}, a_{out1}, a_{in2}, a_{out2}\}; C=\{C(p_{in})= \text{输入类型} \cup \text{多态类型}, C(p_{out})= \text{输出类型}\}; ND=\{ND(a_{in1})=(p_{in}, t_{r1}), ND(a_{out1})=(t_{r1}, p_{out1}), ND(a_{in2})=(p_{in2}, t_{r2}), ND(a_{out2})=(t_{r2}, p_{out2})\}; E=\{E(a_{in1})= \text{基于输入和多态类型的输入函数}, E(a_{in2})= \text{基于输入和多态类型的输入函数}, E(a_{out1})= \text{相应的输出函数}, E(a_{out2})= \text{相应的输出函数}\}; G \text{ 表示变迁 } t_{r1} \text{ 和 } t_{r2} \text{ 相应的守卫函数}; \alpha_1, \alpha_2 \text{ 分别表示变迁 } t_{r1} \text{ 和 } t_{r2} \text{ 的最早开始时间}; \beta_1, \beta_2 \text{ 分别表示变迁 } t_{r1} \text{ 和 } t_{r2} \text{ 的最迟开始时间}; \mu_1, \mu_2 \text{ 分别表示变迁 } t_{r1} \text{ 和 } t_{r2} \text{ 的使能代价函数}; \nu_1, \nu_2 \text{ 分别表示变迁 } t_{r1} \text{ 和 } t_{r2} \text{ 的实施代价函数}; IN \text{ 为初始函数}.$

如图 3 所示,它表示了计算单元  $R$  具有两个多态性的服务  $t_{r1}$  和  $t_{r2}$ ,分别为普通用户(normal)和贵宾用户(VIP)提供服务,但服务的内容、响应时间及价格都不同.在建模过程中,通过有色集(normal, VIP)代表不同用户类型的请求.当输入库所  $p_{in}$  中输入的数据类型是 normal 时,代表 normal 用户请求服务,计算单元  $R$  执行个性化功能  $t_{r1}$ ;当输入库所  $p_{in}$  中输入的数据类型是 VIP 时,代表 VIP 用户请求服务,计算单元  $R$  执行个性化功能  $t_{r2}$ .这里给出的例子是对网构软件中基本单元软件实体的多态性建模.由于网构软件可以由一个或多个基本单元实体组合构成,因此可以通过对构成中每个基本单元实体的多态性进行建模来实现对整个网构软件的多态性建模,并通过有色集构成的多重集对模型的行为进行描述和控制.如,可以通过多重集  $\{VIP, Float\}$  实现先执行计算单元  $R$  的个性化服务  $t_{r2}$ ,再执行下一个计算单元的浮点计算服务.通过例子可以看出,使用有色集可以对网构软件的多态性动态行为进行描述,进一步增加了网构软件建模的准确性,为准确地分析和验证网构软件打下了基础.

### 3.2 多Agent网构软件的PTCPN建模

为了能够满足开放网络环境对方法调用多样化的需求,文献[2]提出了基于 Agent 技术的协同程序设计方法.为了能够对此类网构软件进行形式化分析,本文根据 Agent 的特点给出了相应 Agent 的 PTCPN 建模方法.

#### (1) 软件协同 Agent (software coordination agent, 简称 SCA)

软件协同 Agent 负责任务的分配,并定义了 Agent 之间交互的规则.软件协同 Agent 是一个三元组  $SCA=(ICP, LP, Con)$ ,如图 4 所示.其中:ICP(intelligent coordination place)为软件协同 Agent 的智能协同库所,负责协同任务的分配;LP(location place)为位置库所,负责感知外部环境变化的信息,并存储各个计算 Agent 提供的服务信息(如名字、地址和接口信息等);Con(connecting)为软件协同 Agent 的接口库所,负责建立 Agent 间的消息传递通道.

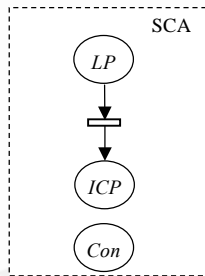


Fig.4 A PTCPN model of software coordination agent

图 4 软件协同 Agent 的 PTCPN 模型

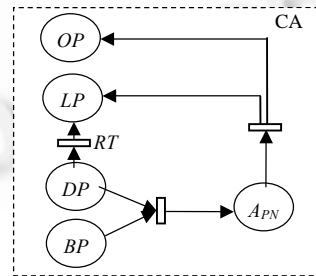


Fig.5 A PTCPN model of computer agent

图 5 计算 Agent 的 PTCPN 模型

#### (2) 计算 Agent (computer agent 简称 CA)

计算 Agent 是一个提供计算功能的 Agent 实体,由接口和内部实现构成.计算 Agent 以 BDI(belief-desire-intention)模型为基础,是一个六元组  $\langle BP, DP, A_{PN}, LP, OP, RT \rangle$ ,其中:BP(belief place)为信念库所,表示计算 Agent 具有的信念;DP(desire place)为愿望库所,表示计算 Agent 具有的愿望; $A_{PN}$ 是一个计算实体的 PTCPN 网,表示计算 Agent 完成计算功能所具有的操作行为集合.这里,  $A_{PN}$  中的  $G$  描述了 Agent 行为的规则;  $[\alpha, \beta], [\mu, \nu]$  分别描述了 Agent 中各个计算单元的时间及代价约束以便于对代价进行计算和分析.  $A_{PN}$  的一个变迁序列  $\delta$  表示  $A_{PN}$  的一个行为序列.如图 5 所示,信念(BP)和愿望(DP)决定 Agent 的行为( $A_{PN}$ );LP(location place)为位置库所,负责存储计算 Agent 提供的服务信息(如名字、地址和接口信息等);OP(output place)是输出库所,负责计算结果的输出;

RT(restriction transition)是限制变迁,表示在无法满足任务需求或等待超时等限制情况下自动结束,避免长期占用计算 Agent 资源.

### 4 实例验证

根据第 3 节的定义及建模方法,可以将一个信息检索网构软件转换成一个 PTCPN.图 6 是一个基于多 Agent 的信息检索网构软件的 PTCPN 模型,该信息检索软件由一个软件协同 Agent(SCA)和 3 个计算 Agent(CA<sub>1</sub>,CA<sub>2</sub>,CA<sub>3</sub>)构成.软件协同 Agent(SCA)主要负责用户请求的输入以及分类、检索 Agent 软件的注册和任务的分配;CA<sub>1</sub> 负责用户检索请求信息的分类,将不同的检索请求信息发送给相应的信息检索 Agent 进行检索;CA<sub>2</sub>,CA<sub>3</sub> 是信息检索 Agent,它们负责根据请求检索的信息在不同的信息数据库进行检索.CA<sub>2</sub> 的检索功能由变迁  $t_{r2}$  表示, $t_{r2}$  执行的延迟时间[3,4], $t_{r2}$  使能所消耗的代价为  $2y+1$ , $t_{r2}$  实施消耗的代价随着延迟时间的增加而降低为  $30-3y$ .检索 Agent CA<sub>2</sub> 没有对不同用户分类进行服务,而检索 Agent CA<sub>3</sub> 分别对会员用户和普通用户进行服务,变迁  $t_{r3},t_{r4}$  分别表示多态性检索 Agent CA<sub>3</sub> 对会员用户和普通用户的两个个性化功能.分类和检索 Agent CA<sub>1</sub>,CA<sub>2</sub> 和 CA<sub>3</sub> 通过协同 Agent(SCA)中的接口库所(Con<sub>1</sub>,Con<sub>2</sub>)进行交互.各个变迁的延迟时间约束及使能代价和实施代价函数见表 1,在这个模型中为了简洁,没有考虑相应控制及连接的延迟时间约束和实施代价.当外部环境的变化导致增加或删除计算 Agent 构件形成新的网构软件.为了保证新构成的网构软件质量,就必须对该网构软件重新进行验证和分析.通过信息检索网构软件的 PTCPN 模型可以看出,当新检索 Agent 软件 CA<sub>4</sub> 加入到该软件中时,可以将 CA<sub>4</sub> 模型中的 DP,BP,LP 和 OP 分别与 SCA 的 ICP,Con<sub>1</sub>,LP 和 Con<sub>2</sub> 相连接,就可以得到新软件的 PTCPN 模型.根据该 PTCPN 模型,可以对新构成的网构软件重新进行验证和分析.

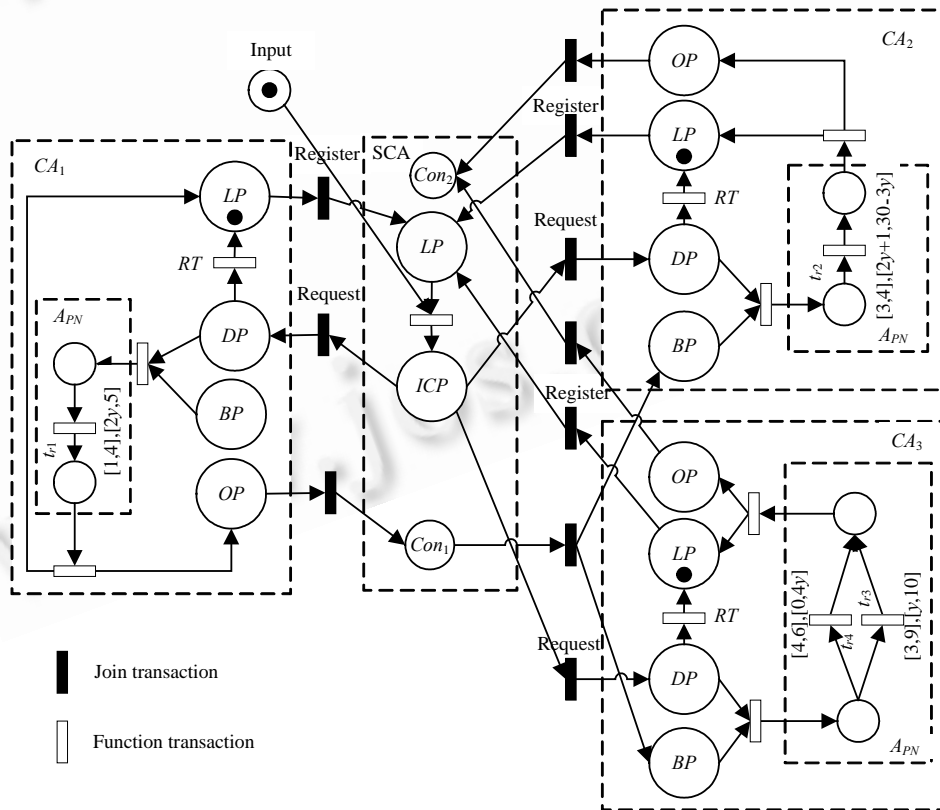


Fig.6 A PTCPN model of a Internetware  
图 6 一个网构软件的 PTCPN 模型

Table 1 Description of transitions

表 1 变迁的含义

| Transition | Explanation       | Time constraint | Enable price | Perform price |
|------------|-------------------|-----------------|--------------|---------------|
| $t_{r1}$   | Classify          | [1,4]           | 2y           | 5             |
| $t_{r2}$   | Search            | [3,4]           | 2y+1         | 30-3y         |
| $t_{r3}$   | Search for member | [3,9]           | y            | 10            |
| $t_{r4}$   | Search for normal | [4,6]           | 0            | 4y            |

(1) 静态分析.如果不考虑变迁上的时间约束和代价约束,那么图 6 就是一个只考虑 Agent 实体间逻辑关系的经典有色 Petri 网模型.从图 7 可以得出:在该网构软件中,3 个计算 Agent( $CA_1, CA_2, CA_3$ )通过注册变迁(register)向软件协同 Agent(SCA)进行注册;接着,SCA 根据任务需求通过请求变迁(request)向计算 Agent 提出计算要求;然后,计算 Agent  $CA_1$  根据库所 BP 的输入进行分类,在分类结束后,通过库所 LP 重新注册,并将计算结果传送给接口库所  $Con_1$ ;最后,计算 Agent  $CA_2, CA_3$  根据接口库所  $Con_1$  的输入进行并行检索,并将结果返回给软件协同 Agent(SCA).静态验证(如冲突、冲撞、死锁等)的具体方法见文献[22].

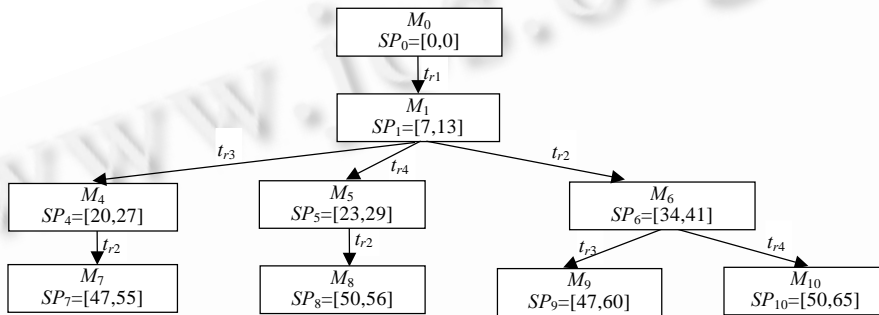


Fig.7 Reachability graph of cumulate price

图 7 累积代价可达图

(2) 时间分析.如果再考虑变迁上的时间约束,那么图 6 就是一个时间有色 Petri 网模型,可以使用状态类方法对模型作可达性分析,进而检验网构软件的时间特性.例如,当变迁  $t_{r1}$  实施以后,变迁  $t_{r2}$  和  $t_{r3}$  或者变迁  $t_{r2}$  和  $t_{r4}$  并发执行,在变迁  $t_{r3}$  实施后到达终止状态.可以根据时间区间计算公式计算出:1) 有变迁  $t_{r3}$  实施并到达终止状态的延迟时间区间为[4,13];2) 有变迁  $t_{r4}$  实施并到达终止状态的延迟时间区间为[5,10].对于实时性等分析验证见文献[11,21].

(3) 代价分析.在对网构软件分析时,也需要对网构软件的成本和能耗等非功能属性进行分析.本文可以将成本和能耗等属性映射到 PTCPN 的代价上,从而满足对这些非功能属性分析的需要.下面按照定义 17 计算了可达变迁序列的累积代价状态类,如图 7 所示.其中:因为  $D$  的值对代价分析没有影响,所以省去  $D$  的具体内容; $M$  表示相应的状态,图中只给出了相应代价变化的状态.从图 7 可以看出,在每个变迁实施后都能够计算出相应的累积代价;同时从图 7 可以看出,能够到达终止状态的 4 个变迁序列  $t_{r1}t_{r3}t_{r2}, t_{r1}t_{r4}t_{r2}, t_{r1}t_{r2}t_{r3}, t_{r1}t_{r2}t_{r4}$  的累积代价区间不同,这是因为在 4 个变迁序列中,变迁  $t_{r2}, t_{r3}, t_{r4}$  每次实施的延迟时间区间和代价不同,导致变迁序列累积代价的区间不同;并且由于变迁  $t_{r3}$  和  $t_{r4}$  的延迟时间区间和可变代价约束不同,导致多态性变迁  $t_{r4}$  实施的代价区间要高于变迁  $t_{r3}$  实施的代价区间.因此,变迁  $t_{r4}$  相应的累积代价区间也高于变迁  $t_{r3}$  相应的累积代价区间.通过代价分析可以找到最优代价的变迁序列(如本例中最优代价的变迁序列为  $t_{r1}t_{r3}t_{r2}$ ),并且可以得到该检索软件的价格区间为[47,60].这就为用户是否选择使用该网构软件提供了价格参考,也为评价该软件的质量(价格是软件质量的重要评价因素之一)提供了可靠的保证.

从实例可以看出,与原有的价格 Petri 网相比,在建模能力上,由于引入了有色集和代价函数,所以本文提出的 PCTPN 模型建模能力更强,可以对网构软件的多态性动态行为和可变价格进行建模;在分析方面上,本文的

模型和原有的价格 Petri 网在静态和时间方面的分析能力相同;但在代价分析上,由于解决了原有价格 Petri 网累积代价计算不够准确的问题,并且解决了因为引入代价函数带来的累积代价计算问题,所以在代价分析上比原有的价格 Petri 网更准确.

## 5 结论和将来的工作

本文在研究网构软件的特点及需求和 Petri 网领域相关工作的基础上,对时间 Petri 网扩展了可变代价及有色信息,使得 Petri 网在能够满足网构软件功能性验证的同时,可以对非功能性属性如时间和代价进行分析,有效地降低了形式化建模及验证的成本.本文主要的工作是:(1) 给出了 PCTPN 的语法,并使用代价时间变迁系统给出了 PCTPN 的语义,进而证明了时间 Petri 网是 PCTPN 的子类;(2) 对可变代价进行分析,在原有价格状态类的基础上进一步分析状态类的时间区间,给出了状态类时间区间的计算公式并证明了计算公式的正确性和完备性.在此基础上提出了累积代价状态类的概念,并证明了累积代价状态类的合理性和完备性;(3) 给出了基于 PTCPN 中的有色集对网构软件实体的多态性建模方法,进而可以满足对网构软件多态性的建模需求;(4) 给出了多 Agent 技术的 PTCPN 建模方法;(5) 最后,通过实例说明 PTCPN 在网构软件建模及分析方面的应用.得出的结论是对时间 Petri 网扩展代价函数及有色信息来描述软件实体的可变代价及多态性,并将代价时间有色 Petri 网应用到网构软件的建模及分析是可行的.

将来的工作包括:将软件实体的服务质量引入代价时间有色 Petri 网模型中,来分析网构软件的服务质量;对网构软件安全性约束的 Petri 网形式化建模与分析;对网构软件中存在的实时冲突进行验证分析.

致谢 感谢审稿专家提出的宝贵意见,感谢老师和同学们提出的意见及所给予的帮助.

## References:

- [1] Yang FQ, Mei H, Lü J, Jin Z. Some discussion on the development of software technology. *Acta Electronica Sinica*, 2002,30(12A): 1901–1906 (in Chinese with English abstract).
- [2] Lü J, Tao XP, Ma XX, Hu H, Xu F, Cao C. Internetwork model research based on agent. *Science in China (Series E) Technological Sciences*, 2005,35(12):1233–1253 (in Chinese with English abstract).
- [3] Mei H, Hang G, Zhao HY, Jao WP. A internetwork development method take software architecture as a center. *Science in China (Series E) Technological Sciences*, 2006,36(10):1100–1126 (in Chinese with English abstract).
- [4] Wang Y, Lü J, Xu F, Zhang L. A trust measurement and evolution model for internetwork. *Journal of Software*, 2006,17(4): 682–690 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/17/682.htm> [doi: 10.1360/jos170682]
- [5] Ping W, Chang SS, Li JL. Primary research on internetwork reliability technology. In: *Proc. of the 1st Int'l Multi-Symp. on Computer and Computational Sciences (IMSCCS 2006)*. Washington: IEEE Computer Society, 2006. 424–428.
- [6] Chandy KM, Jayadev M. *Parallel Program Design: A Foundation*. New York: Addison Wesley, 1989.
- [7] Yuan CY. *Theory and Application of Petri Net*. Beijing: Publishing House of Electronics Industry, 2005 (in Chinese).
- [8] Girault C, Valk R. *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications*. Berlin: Springer-Verlag, 2003. 473–566.
- [9] Jensen K. Colored Petri nets: A high level language for system design and analysis. *LNCS 483*, 1991. 342–416.
- [10] Ramchandani C. *Analysis of asynchronous concurrent systems by timed Petri nets* [Ph.D. Thesis]. Cambridge: MIT, 1974.
- [11] Cui M, Li XD, Zheng GL. Formal analysis on UML real-time activity diagram. *Chinese Journal of Computers*, 2004,27(3): 340–346 (in Chinese with English abstract).
- [12] Liu XM, Li SX, Jiang Y. A Petri net extended with stochastic priced transition. In: *Proc. of the 2005 IEEE Int'l Conf. on e-Business Engineering*. Los Alamitos: IEEE Computer Society, 2005. 545–548.
- [13] Liu WD, Song JX, Lin C. Modeling and analysis of grid computing application based price timed Petri net. *Acta Electronica Sinica*, 2005,33(8):1416–1420 (in Chinese with English abstract).

- [14] Liu XM, Li SX, Li WJ, Pan L. A time Petri net extended with price information. Journal of Software, 2007,18(1):1-10 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/1.htm> [doi: 10.1360/jos180001]
- [15] Stalk G. Time—The next source of competition advantage. Harvard Business review, 1988,66(4):41-51.
- [16] Magretta J. Fast, global and entrepreneurial: Supply chain management, Hong Kong style. Harvard Business Review, 1998,76(5):102-114.
- [17] Behrmann G, Larsen KG, Rasmussen JI. Optimal scheduling using priced timed automata. ACM Sigmetrics Performance Evaluation Review, 2005,32(4):34-40. [doi: 10.1145/1059816.1059823]
- [18] Berthomieu B, Diaz M. Modeling and verification of time dependent systems using time Petri nets. IEEE Trans. on Software Engineering, 1991,17(3):259-273. [doi: 10.1109/32.75415]
- [19] Roux OH, Lime D. Time Petri nets with inhibitor hyperarcs: Formal semantics and state space computation. LNCS 3009, Berlin: Springer-Verlag, 2004. 371-390.
- [20] Berthomieu B, Vernadat F. State class constructions for branching analysis of time Petri nets. LNCS 2619, 2003. 442-457.
- [21] Li HF, Fan YS. Workflow model analysis based on time constraint Petri nets. Journal of Software, 2004,15(1):17-26. <http://www.jos.org.cn/1000-9825/15/17.htm>
- [22] Reijers HA. Design and Control of Workflow Processes. Berlin: Springer-Verlag, 2003. 32-59.

#### 附中文参考文献:

- [1] 杨芙清,梅宏,吕建,金芝. 浅论软件技术发展. 电子学报,2002,30(12A):1901-1906.
- [2] 吕建,陶先平,马晓星,胡昊,徐锋,曹春. 基于 Agent 的网构软件模型研究. 中国科学(E 辑),2005,35(12):1233-1253.
- [3] 梅宏,黄罡,赵海燕,焦文品. 一种以软件体系结构为中心的网构软件开发方法. 中国科学(E 辑),2006,36(10):1100-1126.
- [4] 王远,吕建,徐锋,张林. 一个适用于网构软件的信任度量及演化模型. 软件学报,2006,17(4):682-690. <http://www.jos.org.cn/1000-9825/17/682.htm> [doi: 10.1360/jos170682]
- [7] 袁崇义. Petri 网原理与应用. 北京:电子工业出版社,2005.
- [11] 崔萌,李宣东,郑国梁. UML 实时活动图的形式化分析. 计算机学报,2004,27(3):340-360.
- [13] 刘卫东,宋佳兴,林闯. 基于代价时间 Petri 网的网格计算应用模型及分析. 电子学报,2005,33(8):1416-1420.
- [14] 刘显明,李师贤,李文军,潘理. 一种扩展了价格信息的时间 Petri 网. 软件学报,2007,18(1):1-10. <http://www.jos.org.cn/1000-9825/18/1.htm> [doi: 10.1360/jos180001]



周航(1978 - ),男,江苏徐州人,博士,讲师,CCF 会员,主要研究领域为软件工程,软件测试与形式化验证,Petri 网应用.



祝义(1976 - ),男,博士生,讲师,主要研究领域为软件工程,形式化方法.



黄志球(1965 - ),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程,软件度量与测试,形式化验证,数据仓库.



胡军(1973 - ),男,博士,副教授,CCF 会员,主要研究领域为软件工程,软件分析与验证,形式化方法.



张广泉(1965 - ),男,博士,教授,CCF 高级会员,主要研究领域为软件体系结构,形式化方法.