软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(6):1199−1209 [doi: 10.3724/SP.J.1001.2011.04018]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

混合语义时间 Petri 网模型*

潘 理[1,2], 丁志军[2,3+], 郭观七[1]

[1](湖南理工学院 信息与通信工程学院,湖南 岳阳 414006)
[2](同济大学 计算机科学与技术系,上海 200092)
[3](武汉大学 软件工程国家重点实验室,湖北 武汉 430072)

Time Petri Net Model with Mixed Semantics

PAN Li[1,2], DING Zhi-Jun[2,3+], GUO Guan-Qi[1]

[1](School of Information and Communication Engineering, Hu'nan Institute of Science and Technology, Yueyang 414006, China)
[2](Department of Computer Science and Technology, Tongji University, Shanghai 200092, China)
[3](State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)
+ Corresponding author: E-mail: zhijun_ding@hotmail.com

Pan L, Ding ZJ, Guo GQ. Time Petri net model with mixed semantics. *Journal of Software*, 2011,22(6): 1199−1209. http://www.jos.org.cn/1000-9825/4018.htm

Abstract: This paper presents a time Petri net (TPN) model with mixed semantics that efficiently addresses the schedulability analysis problem of the existing semantics models. The proposed model associates each firable transition with a mandatory firing point that is set to the least upper bound of the transition and its non-conflicting transitions. This treatment not only eliminates the effect of conflicting transitions on the transition friability and extends the scheduling scope of the TPN model, but also ensures the time limit of the task scheduling. To determine the expressiveness of the mixed model, the study proves that the mixed semantics model has the power of Turing machines and its marking reachability problem is undecidable. The timed expressiveness of three semantics models with respect to timed language acceptance is also compared. A state class approach is then proposed for the scheduling analysis of the mixed model. Finally, a flexible manufacturing system is used as an example to compare the scheduling ability of three semantics models.
Key words: time Petri net; mixed semantics; turing equivalence; undecidability; schedulability analysis

摘 要: 提出了时间 Petri 网的混合语义模型,通过在变迁及其非冲突变迁集的最小上界处设置强制实施点,排除冲突变迁对变迁可实施性的影响,达到既能扩大模型调度范围又可保证任务调度时限性的目的,以解决现有语义模型在调度分析上的缺陷.进一步证明了混合语义模型的图灵等价性及标识可达性问题的不可判定性,然后界定了 3 种语义模型的时间语言接受能力.最后提出了状态类分析方法,用于模型的可调度性分析和时间计算,并以一个柔性制造系统为例,比较和验证了 3 种语义模型的调度分析能力.

* 基金项目: 国家自然科学基金(60803032, 60975049); 教育部新世纪优秀人才计划(NCET-10-0598); 上海市重点基础研究项目基金(09JC1414200); 上海市"曙光计划"; 上海市青年科技启明星计划(09QA1405900); 武汉大学软件工程国家重点实验室开放基金项目

收稿时间: 2010-07-10; 定稿时间: 2011-03-29

© 中国科学院软件研究所 http://www.jos.org.cn

关键词: 时间 Petri 网;混合语义;图灵等价性;不可判定性;调度分析
中图法分类号: TP301 **文献标识码:** A

1 引 言

1.1 动 机

时间 Petri 网[1,2]是描述和验证实时系统最常用的形式化模型之一,它在经典 Petri 网的基础上,为每个变迁关联一个静态实施间隔,用以表示该变迁最小和最大的持续使能时间.当某个变迁开始使能,就会联系一个动态实施间隔,其初值为变迁的静态实施间隔.间隔的上下界会随着时间的流逝同步减小.当动态间隔的下界到达 0 时,该变迁可以实施;当动态间隔的上界到达 0 时,存在两种语义行为[3−5]:强语义(strong semantic)和弱语义(weak semantic).强语义强迫变迁在上界为 0 时必须实施;而弱语义则允许上界低于 0,但此后,该变迁不能再实施(除非重新使能).一旦该变迁丧失使能,关联它的动态间隔就会被抛弃.变迁实施不计时间.

两种时间语义虽然在不同环境下拥有各自的用途,但在调度分析上都各自存在缺点.强语义非常适合建模和分析具有严格时间约束和最终期限的硬实时系统,但是存在部分变迁由于时间竞争处于劣势而永远得不到选择,极大地限制了模型的调度范围;弱语义可以方便地建模交互式系统和软实时系统,但是不能保证任务调度的时限性,从而导致某些调度序列因为过期而中断.

如图 1 所示,一项任务可以由活动 1(t_1)执行,也可以由活动 2(t_2)和活动 3(t_3)相继执行.在强语义模型里,由于不允许变迁超越时间上界,t_2 总是被迫优先于 t_1 实施,从而导致 t_1 永远得不到调度.其实,t_1 完成任务的时间比 t_2 和 t_3 要少.显然,这种调度是不合理的.因此,强语义模型的强时间约束并不利于任务的调度分析.另一方面,在弱语义模型里,由于允许变迁错过时间上限,故 t_1 和 t_2 都有被选择的可能.例如,t_2 可以错过期限,t_1 则在[5,6]内实施.但 t_1 也可能错过期限而得不到实施,这时整个任务就无法完成调度了.

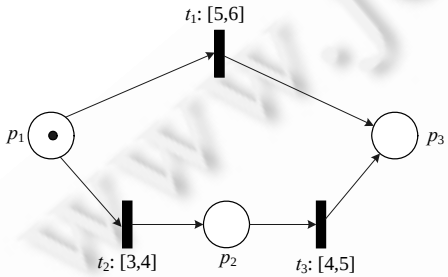


Fig.1 A simple task schedule
图 1 一个简单的任务调度

通过上例不难发现:强语义会极大地限制调度序列的实施,严重制约模型的调度范围;弱语义尽管解除了变迁的强制性限定,但却不能保证任务调度的时限性.因此,时间 Petri 网的现有语义模型并不适合实时系统的调度分析.本文试图结合现有强、弱语义模型的特点,建立一种混合语义模型来解决时间 Petri 网面临的调度分析问题.在混合语义模型中,可实施条件将被适度放宽,但仍然保留某种强时间限制,以达到既能扩大任务调度范围又能保证任务调度时限的目的.

1.2 相关工作

对于一个时间 Petri 网模型,首先应该定义其形式语义,才能准确刻画模型的动态行为和时态性质.其次,应界定其表达能力,包括模型的图灵模拟能力和时间表达能力.然后,应提出模型的分析方法,用于时间计算和性质分析.下面简要综述现有语义模型在上述 3 个方面的研究现状.

在形式语义方面,Boyer 等人利用时间变迁系统定义了强、弱两种时间 Petri 网的操作语义[4].Felder 等人利用 TRIO 公理描述了强、弱时间 Petri 网的公理语义[5].Koutny 基于时间 Petri 盒演算探讨了时间 Petri 网的指称语义[6].关于时间 Petri 网的代数语义描述,目前还未发现公开的研究成果.

在表达能力方面,Jones 等人证明了强语义时间 Petri 网具有图灵机模拟能力[7],因为它能模拟双计数器机,而后者是图灵等价的.Cerone 证明弱语义时间 Petri 网不是图灵等价的[3],因为它不能模拟零测试.此外,Berthomieu 证明标识可达性问题、有界性问题、活性问题对强语义时间 Petri 网是不可判的[2],但对有界的强语义时间 Petri 网是可判的[8].最近,文献[9]证明了弱语义时间 Petri 网的标识可达性问题也是可判定的.

© 中国科学院软件研究所 http://www.jos.org.cn

在状态类方法方面,Berthomieu 等人首先提出经典状态类方法[2].该方法采用相对时间模式,因此不能有效计算状态类的全局延时.为了解决这一问题,Wang 等人提出时间戳状态类方法,通过引入全局时间戳,计算状态类的全局时间[10].但是该方法并不能准确判断变迁的可实施性,因此并未得到推广应用.此后,Vicario 提出一种全局集方法计算任何两个状态类之间的时间延时[11],但是一个全局集需要包括所有已访问的状态类的时间约束,因此这种计算方法的空间成本很高.

值得注意的是,上述研究工作都是针对单一语义(强语义或弱语义)进行的.

1.3 我们的贡献

针对时间 Petri 网现有语义模型面临的调度分析问题,本文提出时间 Petri 网的混合语义模型(M-TPN).在 M-TPN 中,变迁的可实施性不再依赖于冲突变迁的时间约束,而是依赖于非冲突变迁的时间约束.这种处理不但可以消除强语义模型中冲突变迁对变迁可实施性的影响,增大模型的调度范围,而且可以通过在变迁及其非冲突变迁的最小上界处设置强制实施点,保证任务调度的最终时限性.进一步,本文利用时间变迁系统定义了 M-TPN 的形式(操作)语义;利用寄存器机证明 M-TPN 具有图灵机的模拟能力,同时证明标识可达性问题是不可判定的;证明 M-TPN 的时间语言接受能力处于强语义和弱语义模型之间,即证明了 M-TPN 既能扩展强语义模型的任务调度范围,又能克服弱语义模型缺乏调度时限性的弊端;最后,提出 M-TPN 的状态类方法,通过引入特殊参考变迁(零变迁),有效解决了任务调度的全局时间计算问题,而且无须构造全局集.

本文第 2 节给出时间 Petri 网的基本以及强、弱语义模型的形式语义.第 3 节提出混合语义时间 Petri 网模型,给出模型的形式语义,界定其图灵模拟能力和时间语言接受能力,并提出模型的状态类方法.第 4 节以一个柔性制造系统为例,比较 3 种模型的调度分析能力.最后一节总结本文.

2 时间 Petri 网

2.1 基本定义

假设 N 是自然数集(包括 0),R 是实数集,R^+是非负实数集.定义实数闭区间 $I=\{x\in R|a\leqslant x\leqslant b\}$,记为 $I=[a,b]$.这里,$a,b\in R$ 且 $a\leqslant b$.用 IR 和 IR^+分别表示所有实数闭区间和非负实数闭区间的集合.若 $a=b$,则 $a=[a,a]$称为点区间.设 $I,I_1,I_2\in IR$,且 $I=[a,b]$,定义闭区间的一元和二元运算:$\downarrow I=a,\uparrow I=b;I_1+I_2=[\downarrow I_1+\downarrow I_2,\uparrow I_1+\uparrow I_2];I_1-I_2=[\downarrow I_1-\uparrow I_2,\uparrow I_1-\downarrow I_2]$;若 $I_1\cap I_2\neq\varnothing$,则 $I_1\cap I_2=[\max\{\downarrow I_1,\downarrow I_2\},\min\{\uparrow I_1,\uparrow I_2\}]$.

定义 1[2]. 一个时间 Petri 网是一个六元组 TPN=$(P,T,Pre,Post,M_0;SI)$,其中:

(1) $P=\{p_1,p_2,\ldots,p_m\}$是有穷、非空的位置集;

(2) $T=\{t_1,t_2,\ldots,t_n\}$是有穷、非空的变迁集;

(3) $Pre:P\times T\rightarrow N$ 是向前关联矩阵;

(4) $Post:P\times T\rightarrow N$ 是向后关联矩阵;

(5) $M_0:P\rightarrow N$ 是初始标识;$(P,T,Pre,Post,M_0)$构成一个 Petri 网;

(6) $SI:T\rightarrow IR^+$是变迁的静态实施间隔映射函数.对$\forall t\in T,SI(t)$表示变迁 t 的静态实施间隔.

$Pre(p,t)=i(i>0)$当且仅当存在一条从库所 p 到变迁 t 的权值为 i 的弧;$Pre(p,t)=0$ 当且仅当不存在从库所 p 到变迁 t 的弧.用 $Pre(t)\in N^P$ 表示 t 的输入库所多重集,即对应向前关联函数 Pre 中 t 列所对应的向量.类似地定义 $Post(p,t)$和 $Post(t)$.Petri 网的标识 M 可表示为一个$|P|$维向量,其中,第 i 个元素 $M(p_i)$表示库所 p_i 中的令牌(token)数.因此,标识能反映资源在库所集中的分布情况.

定义 2[2]. 如果 $Pre(t)\leqslant M$,则称变迁 $t\in T$ 在标识 M 是使能的(enabled).如果变迁 t 在标识 M 是使能的,那么 t 可以实施(fire)并产生一个新的后继标识 M',且 $M'=M-Pre(t)+Post(t)$.

用 $En(M)$表示在标识 M 使能的所有变迁的集合,用 $Newly(M,t_f)$表示在标识 M 实施 t_f后新使能的所有变迁的集合,即 $Newly(M,t_f)=\{t\in T|(t\in En(M'))\wedge((t\notin En(M-Pre(t_f)))\vee(t=t_f))\}$.这里,$M'=M-Pre(t_f)+Post(t_f)$.为简化问题的处理,本文不考虑变迁的多重使能[2,11].

© 中国科学院软件研究所 http://www.jos.org.cn

2.2 强语义和弱语义

时间 Petri 网的形式语义可以使用带时间标号的状态变迁系统(timed transition system)来定义,因此状态刻画非常重要.时间 Petri 网的状态通常采用两种表示方法:时钟方法[4]和间隔方法[2,10].时钟方法给每个使能变迁联系一个时钟,用来记录该变迁最近一次的持续使能的时间.间隔方法给每个使能变迁联系一个动态实施间隔,用以表示该变迁相对于当前状态的所有可能的实施时间.本文使用间隔方法.

定义 3[2]. 一个时间 Petri 网的状态是一个二元组 $s=(M,f)$,其中:

(1) $M\in N^P$ 是标识;

(2) $f:T\to IR$ 是一个动态实施间隔函数.$\forall t\in T,f(t)$表示 t 相对于当前状态的实施时间间隔.

时间 Petri 网的初始状态 $s_0=(M_0,f_0)$,其中,M_0 是初始标识,$\forall t\in En(M_0),f_0(t)=SI(t)$.

定义 4[4]. 一个时间变迁系统是一个四元组 $TTS=(S,S_0,\Sigma,\to)$,其中:

(1) S 是状态集;

(2) $S_0\subseteq S$ 是初始状态集;

(3) Σ是描述时间事件的标号集;

(4) $\to\subseteq S\times\Sigma\times S$ 表示状态转移关系.

定义 5. 一个强语义时间 Petri 网 S-TPN=$(P,T,Pre,Post,M_0;SI)$的形式语义定义为一个时间变迁系统 $TTS_1=(S_1,S_0,\Sigma,\to_1)$,且使得:

(1) $S_1\subseteq N^P\times IR^T$ 是 S-TPN 的状态集;

(2) $S_0=\{s_0\}$是 S-TPN 的初始状态集;

(3) $\Sigma=T\times R^+$是时间变迁标号集;

(4) $\to_1\subseteq S_1\times\Sigma\times S_1$ 是状态转移关系,且使得$\forall d\in R^+,\forall t_f\in T$,

$$(M,f)\xrightarrow{(t_f,d)}{}_1(M',f') \text{ iff } \begin{cases} t_f\in En(M) & \text{(a)}\\ \downarrow f(t_f)\leqslant d\leqslant\uparrow f(t_f) & \text{(b)}\\ \forall t\in En(M), d\leqslant\uparrow f(t) & \text{(c)}\\ M'=M-Pre(t_f)+Post(t_f) & \text{(d)}\\ \forall t\in En(M'), f'(t)=\begin{cases} SI(t), & \text{if } t\in Newly(M,t_f)\\ f(t)-d, & \text{otherwise}\end{cases} & \text{(e)}\end{cases}$$

从状态转移关系可以看出,状态转移是指从状态 s 流逝一段时间 d,然后实施变迁 t_f到达另一个状态 s'.转移关系(a)表明 t_f在 M 使能,转移关系(b)保证流逝时间 d 位于 t_f的动态实施间隔内,转移关系(c)确保 d 不超过所有使能变迁的实施时间上界(强语义),转移关系(d)计算新状态的标识,转移关系(e)计算新状态下使能变迁的实施间隔.

定义 6. 一个使能变迁 t 在状态 s 是过期的(over-due),当且仅当$\uparrow f(t)<0$.

定义 7. 一个弱语义时间 Petri 网 W-TPN=$(P,T,Pre,Post,M_0;SI)$的形式语义定义为一个时间变迁系统 $TTS_2=(S_2,S_0,\Sigma,\to_2)$,它与强语义定义的不同仅在状态转移关系上,即删除条件(c).

可以看出,弱语义只保证 t_f在自己的实施间隔内实施,它不关心时间流逝 d 是否会超过其他使能变迁的时间上界.因此,弱语义模型中会出现过期变迁.过期会导致变迁丧失可实施性,因为它不满足条件(b).

3 混合语义模型

3.1 形式语义

从定义 5 可知,强语义模型将强制实施点设置在所有使能变迁的最小上界处,因此,每个使能变迁的时间约束都对变迁可实施性造成影响,极大地制约了模型的调度范围.从定义 7 可知,弱语义模型并不设置强制实施点,因此不能保证任务调度的时限性.为了弥补两种语义模型带来的缺陷,我们提出混合语义模型,将变迁的强制实

© 中国科学院软件研究所 http://www.jos.org.cn

施点延长至该变迁及非冲突变迁集的最小上界处,不仅放宽了变迁的可实施条件,而且还保留了某种强时间限制.因此,与 S-TPN 相比,变迁的调度范围被进一步扩大;与 W-TPN 相比,又保证了任务调度的最终时限性.

定义 8. 一个混合语义时间 Petri 网 M-TPN=$(P,T,Pre,Post,M_0;SI)$的形式语义定义为一个时间变迁系统 $TTS_3=(S_3,S_0,\Sigma,\rightarrow_3)$,它与强语义定义的不同仅在状态转移关系上,即修改条件(c)为

$$\forall t\in En(M),d\leqslant Tb(s,t),$$

这里,$Tb(s,t)=\min\{\uparrow f(t),\uparrow f(t')|t'\in En(M-Pre(t))\}$.

容易看出,判断变迁 t 在状态 s 的可实施性需要时间 $O(n)$.这里,$n=|T|$.

下面以图 2 为例说明 M-TPN 的状态演变.初始状态 $s_0=(M_0,f_0)$,其中:

- $M_0=(1\ 1)$;
- $f_0(t_1)=[1,2],f_0(t_2)=[3,4],f_0(t_3)=[5,6],f_0(t_4)=[3,4]$;
- $Tb(s_0,t_1)=2,Tb(s_0,t_2)=4,Tb(s_0,t_3)=2,Tb(s_0,t_4)=2$.

根据定义 8 的状态转移关系(b)和(c),有 $d_1\in f_0(t_1)$且 $d_1\leqslant Tb(s_0,t_1),d_2\in f_0(t_2)$且 $d_2\leqslant Tb(s_0,t_2)$,故 t_1 和 t_2 在 s_0 可实施.但由于 $d_3\in f_0(t_3)$且 $d_2>Tb(s_0,t_3),d_4\in f_0(t_4)$且 $d_4>Tb(s_0,t_4)$,故 t_3 和 t_4 在 s_0 不可实施.值得注意的是,对于 S-TPN,在 s_0 仅 t_1 可实施;对于 W-TPN,4 个变迁均可实施.

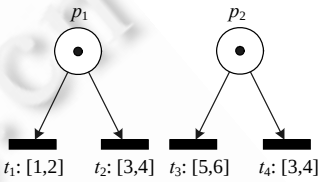


Fig.2 A TPN with choice structure

图 2 一个具有选择结构的时间 Petri 网

假如在时刻 3(d=3)实施 t_2,导致状态 $s_1=(M_1,f_1)$,其中,

$$M_1=(0\ 1);f_1(t_3)=[2,3],f_1(t_4)=[0,1];Tb(s_1,t_3)=3,Tb(s_1,t_4)=1.$$

根据 M-TPN 的状态转移关系(b)、状态转移关系(c),t_3 和 t_4 在状态 s_1 可实施.假如在时刻 6(d=3)实施 t_3,导致状态 $s_2=(M_2,f_2)$,其中,

$$M_2=(0\ 0);En(M_2)=\{\}.$$

从上例可以看出,M-TPN 比 S-TPN 具有更大的调度范围.例如,在 M-TPN 中,图 2 的可实施序列有 t_1t_3,t_1t_4,t_2t_3 和 t_2t_4;但在 S-TPN 中,可实施序列只有 t_1t_3.另一方面,M-TPN 能够保证任务调度的时限性,即能够克服 W-TPN 中因为变迁过期导致整个选择结构被中断的缺点.例如:对于 W-TPN,变迁 t_3 在状态 s_0 可调度,于是 t_1 和 t_2 都变为过期而丧失实施性;对于 M-TPN,在状态 s_0,变迁 t_3 不会被调度.

定义 9. 给定标识 M,对于变迁 $t_1,t_2\in En(M)$,若$\exists p\in P$,使 $Pre(p,t_1)+Pre(p,t_2)>M(p)$,则称 t_1 和 t_2 在标识 M 是(有效)冲突的.对于 $U\subseteq T$,若 U 中任何两个变迁在 M 是有效冲突的,则称 U 是 M 的冲突集.若不存在其他冲突集 U',使 $U\subset U'$,则称 U 是 M 的极大冲突集.用 $MCS(M)$表示 M 的所有极大冲突集的集合.

每个极大冲突集就表示一个选择结构,图 2 所示 Petri 网的极大冲突集为$\{t_1,t_2\}$和$\{t_3,t_4\}$.

下面证明 M-TPN 不会出现过期的极大冲突集,即不会出现因变迁过期而导致选择结构被中断的现象.

定理 1. M-TPN 中不存在过期的极大冲突集.

证明:使用归纳法.当 n=0,即在初始状态 s_0,$\forall t\in En(M_0)$,有$\uparrow f(t)\geqslant 0$,故定理对 n=0 成立.

假设定理对 $n\leqslant k$ 成立,考虑 $n=k+1$ 的情况.

若 $s_k\xrightarrow{(t_k,d_k)}_3 s_{k+1}$,则蕴涵着 $t_k\in En(M_k)$且$\downarrow f_k(t_k)\leqslant d_k\leqslant\uparrow f_k(t_k)$且$\forall t\in En(M_k),d_k\leqslant Tb(s_k,t)$.接下来证明 d_k 不会超过任何一个极大冲突集的上界.不妨假设存在 $U\in MCS(M_k)$,使得$\uparrow U<d_k$,则$\forall t\in En(M_k)$:

- 若 $t\notin U$,则 $Tb(s_k,t)=\min\{\uparrow f_k(t),\uparrow f_k(t')|t'\in En(M_k-Pre(t))\}\leqslant\min\{\uparrow f_k(t')|t'\in U\}\leqslant\uparrow U<d_k$,与 $d_k\leqslant Tb(s_k,t)$矛盾;
- 若 $t\in U$,则有 $Tb(s_k,t)\leqslant\uparrow f_k(t)\leqslant\uparrow U<d_k$,矛盾.

因此,在 s_{k+1} 不会出现过期的极大冲突集.故 M-TPN 不存在过期的极大冲突集.证毕. □

3.2 时间语言接受

定义 10[4]. 时间 Petri 网的时间实施序列是一个由变迁和时间序对组成的有限(或无限)序列:

$$w=(t_0,d_0)(t_1,d_1)\ldots(t_n,d_n),$$

© 中国科学院软件研究所 http://www.jos.org.cn

且使得 $s_i \xrightarrow{(t_i,d_i)} s_{i+1}$.这里,$t_i \in T, d_i \in R^+, i=0,1,\ldots,n$.

定义 11[4]. 一个时间 Petri 网可接受的时间语言定义为该时间 Petri 网的所有时间实施序列的集合.

定义 12[4]. 如果一个 A 类时间 Petri 网所接受的时间语言都能被 B 类时间 Petri 网所接受,则称 A 是 B 的子类,记为 $A \subseteq_L B$;若反之不成立,则称 A 是 B 的严格子类,记为 $A \subset_L B$.

下面比较 M-TPN,S-TPN 和 W-TPN 关于时间语言接受的表达能力.

定理 2. S-TPN$\subset_L$M-TPN$\subset_L$W-TPN.

证明:根据 3 种语义模型的语义定义可知,三者的初始状态相同,唯一区别在于变迁关系的条件(c),故只需证明 $\xrightarrow{(t_f,d)}_1 \subset \xrightarrow{(t_f,d)}_3 \subset \xrightarrow{(t_f,d)}_2$.

若 $s \xrightarrow{(t_f,d)}_1 s'$,根据定义 5,有$\forall t \in En(M): d \leqslant \uparrow f(t)$,即 $d \leqslant \min\{\uparrow f(t) | t \in En(M)\}$.于是有

$$d \leqslant \min\{\uparrow f(t), \uparrow f(t') | t' \in En(M-Pre(t))\}.$$

根据定义 8,有 $s \xrightarrow{(t_f,d)}_3 s'$.再根据定义 7,得 $s \xrightarrow{(t_f,d)}_2 s'$.因此,$\xrightarrow{(t_f,d)}_1 \subseteq \xrightarrow{(t_f,d)}_3 \subseteq \xrightarrow{(t_f,d)}_2$.

考虑图 2,不妨设 $a \in [1,2], b \in [3,4], c \in [5,6]$,则时间实施序列$(t_3,c)$在 W-TPN 可接受但在 M-TPN 不可接受;时间实施序列$(t_2,b)(t_3,c)$在 M-TPN 可接受但在 S-TPN 不可接受,故得 $\xrightarrow{(t_f,d)}_1 \subset \xrightarrow{(t_f,d)}_3 \subset \xrightarrow{(t_f,d)}_2$.

因此,S-TPN$\subset_L$M-TPN$\subset_L$W-TPN.证毕. □

可见,相对于 S-TPN 模型,M-TPN 扩展了一些可接受的时间语言,这些时间语言代表了那些可能使某个极大冲突集中部分变迁超过期限、但不会使其中所有变迁超过期限的变迁实施.相对于 W-TPN 模型,M-TPN 则限制了一些可接受的时间语言,这些时间语言代表了那些使某个极大冲突集中所有变迁都超过时限的变迁实施.因此,M-TPN 模型既扩展了 S-TPN 的调度范围,又保证了变迁选择的最终时限性.

3.3 模拟能力

下面借助寄存器机模型[12]证明 M-TPN 具备图灵机的模拟能力.

定义 13[12]. 一个寄存器机 RM 是一个由有限个寄存器组成的计算模型,每个寄存器可以存储一个任意大的自然数,并使用下列指令操作寄存器:

1. $Add(k)$:使寄存器 k 加 1;
2. $Dec(k)$:使寄存器 k 减 1(当寄存器 k 不为 0 时);
3. $Jump(k)[h]$:若寄存器 k 为 0,转到第 h 条指令.

一个有限的指令序列构成一个程序.为了在 RM 上执行一个计算,必须提供一个程序和初始格局.一个格局(configuration)由指令计数器的指针值和所有寄存器的内容值组成.程序从第 1 条指令开始执行,然后按照指令序列依次往下执行,除非碰到跳转指令.程序终止(RM 停机)仅当指令计数器指针指向一个大于程序长度的数字(通常假设为程序长度+1).程序终止时,寄存器的内容即为计算的结果.因此,RM 上的一个计算可以定义为一个有限(或无限)的格局序列.已经证明,具有上述 3 条指令的寄存器机是图灵等价的[12].

定理 3. 混合语义时间 Petri 网能够模拟寄存器机.

证明:给定一个寄存器机 RM,假设 RM 由 m 个寄存器 $R_1,\ldots,R_m$ 组成,第 i 个寄存器的内容用 r_i 表示.不妨设在 RM 上计算的程序有 s 条指令 $I_1,\ldots,I_s$.我们将使用 M-TPN 的变迁实施模拟这些指令的执行.

首先,我们按下列方式将 RM 变换成相应的 M-TPN(如图 3 所示):

a) 添加库所 $p'_1,\ldots,p'_m$ 到库所集 P,用以表示 RM 的 m 个寄存器;添加库所 $p_1,\ldots,p_{s+1}$ 到库所集 P,用以表示程序指令计数器的指针值;

b) 若第 i 条指令为 $Add(k)$,添加变迁 t_i 到变迁集 T,且使得

$$Pre(p_i,t_i)=Post(p_{i+1},t_i)=Post(p'_k,t_i)=1; SI(t_i)=0.$$

c) 若第 i 条指令为 $Dec(k)$,添加变迁 t_i 到变迁集 T,且使得

$$Pre(p_i,t_i)=Pre(p'_k,t_i)=Post(p_{i+1},t_i)=1; SI(t_i)=0.$$

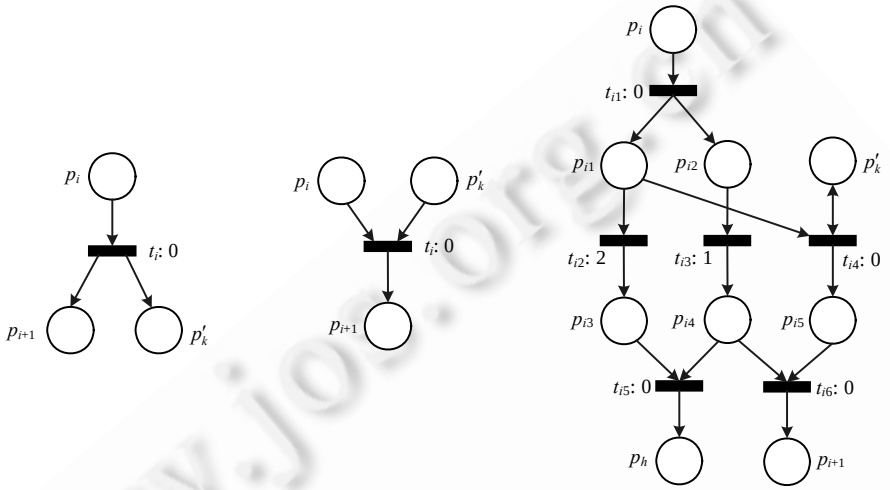
d) 若第 i 条指令为 $Jump(k)[h]$,添加变迁 $t_{i1},t_{i2},t_{i3},t_{i4},t_{i5},t_{i6}$ 到变迁集 T,添加库所 $p_{i1},p_{i2},p_{i3},p_{i4},p_{i5}$ 到库所集

© 中国科学院软件研究所 http://www.jos.org.cn

P,且使得:

$$Pre(p_i,t_{i1})=Post(p_{i1},t_{i1})=Post(p_{i2},t_{i1})=1;SI(t_{i1})=0,$$
$$Pre(p_{i1},t_{i2})=Post(p_{i3},t_{i2})=1;SI(t_{i2})=2,$$
$$Pre(p_{i2},t_{i3})=Post(p_{i4},t_{i3})=1;SI(t_{i3})=1,$$
$$Pre(p_{i1},t_{i4})=Pre(p'_k,t_{i4})=Post(p_{i5},t_{i4})=1;SI(t_{i4})=0,$$
$$Pre(p_{i3},t_{i5})=Pre(p_{i4},t_{i5})=Post(p_h,t_{i5})=1;SI(t_{i5})=0,$$
$$Pre(p_{i4},t_{i6})=Pre(p_{i5},t_{i6})=Post(p_{i+1},t_{i6})=1;SI(t_{i6})=0.$$

e) 对于初始格局 $c_0=(l_0,r_1,\ldots,r_m)$,$l_0=1$,我们令 $M_0(p_1)=1$,$M_0(p_j)=0$ 且 $M_0(p'_i)=r_i$.这里,$1\leqslant i\leqslant m$,$2\leqslant j\leqslant s+1$.



(a) Simulation of $Add(k)$ (b) Simulation of $Dec(k)$ (c) Simulation of $Jump(k)[h]$

(a) 模拟指令$Add(k)$ (b) 模拟指令$Dec(k)$ (c) 模拟指令$Jump(k)[h]$

Fig.3 Transformation of RM into M-TPN

图 3 转换 RM 为 M-TPN

接下来,用归纳法证明 M-TPN 的运行能够模拟 RM 的计算.

当 $n=0$,根据前面的构造,有 $M_0(p_1)=1$,$M_0(p_j)=0$ 且 $M_0(p'_i)=r_i^0$. 这里,$1\leqslant i\leqslant m$,$2\leqslant j\leqslant s+1$.因此,初始标识 M_0 能够模拟初始格局 $c_0=(1,r_1^0,\ldots r_m^0)$.

假设断言对 $n\leqslant v$ 成立,考虑 $n=v+1$ 的情况.

由归纳假设,M-TPN 的运行 $M_0\to M_1\to\ldots\to M_v$ 能够模拟 RM 的计算 $(1,r_1^0,\ldots r_m^0)(l_1,r_1^1,\ldots r_m^1)\ldots(l_v,r_1^v,\ldots r_m^v)$,使得 $M_v(p_{l_v})=1$,$M_v(p_j)=0$ 且 $M_v(p'_i)=r_i^v$.这里,$1\leqslant i\leqslant m$,$1\leqslant j\leqslant s+1$ 且 $j\neq l_v$.现在分成 3 种情况讨论:

(a) 若第 l_v 条指令为 $Add(k)$,则第 $v+1$ 个格局 $c_{v+1}=(l_{v+1},r_1^{v+1},\ldots r_m^{v+1})$ 且 $l_{v+1}=l_v+1$, $r_k^{v+1}=r_k^v+1$ 和 $r_i^{v+1}=r_i^v$.这里,$1\leqslant i\leqslant m$ 且 $i\neq k$.相应地,在标识 M_v,唯一可实施的变迁为 t_{l_v}.实施 t_{l_v} 将到达标识 M_{v+1},且有

$$M_{v+1}(p'_k)=M_v(p'_k)+1=r_k^v+1,M_{v+1}(p'_i)=M_v(p'_i)=r_i^v,M_{v+1}(p_{l_v+1})=1,M_{v+1}(p_j)=0.$$

这里,$1\leqslant i\leqslant m$ 和 $i\neq k$,$1\leqslant j\leqslant s+1$ 和 $j\neq l_v+1$.所以,标识 M_{v+1} 能够模拟格局 c_{v+1};

(b) 若第 l_v 条指令为 $Dec(k)$,证明过程与情况(a)类似(区别仅在于 $r_k^{v+1}=r_k^v-1$ 和 $M_{v+1}(p'_k)=r_k^v-1$);

(c) 若第 l_v 条指令为 $Jump(k)[h]$,再分成两种子情况来考虑:

(i) 若 $r_k^v=0$,则第 $v+1$ 个格局 $c_{v+1}=(l_{v+1},r_1^{v+1},\ldots r_m^{v+1})$ 且 $l_{v+1}=h$, $r_i^{v+1}=r_i^v$.这里,$1\leqslant i\leqslant m$.相应地,在标识 M_v,实施变迁序列 $t_{l_v1}t_{l_v3}t_{l_v2}t_{l_v5}$ 导致标识 M_{v+1},且使得 $M_{v+1}(p'_i)=M_v(p'_i)=r_i^v$,$M_{v+1}(p_h)=1$,$M_{v+1}(p_j)=0$.

© 中国科学院软件研究所 http://www.jos.org.cn

这里,$1\leqslant i\leqslant m$,$1\leqslant j\leqslant s+1$ 和 $j\neq h$.所以,标识 M_{v+1} 能够模拟格局 c_{v+1}.

(ii) 若 $r_k^v=1$,则第 $v+1$ 个格局 $c_{v+1}=(l_{v+1},r_1^{v+1},...r_m^{v+1})$ 且 $l_{v+1}=l_v+1$, $r_i^{v+1}=r_i^v$.这里,$1\leqslant i\leqslant m$.相应地,在标识 M_v,实施变迁序列 $t_{l_v1}t_{l_v4}t_{l_v3}t_{l_v6}$ 导致标识 M_{v+1},且使得

$$M_{v+1}(p_i')=M_v(p_i')=r_i^v, M_{v+1}(p_{l_v+1})=1, M_{v+1}(p_j)=0$$

这里,$1\leqslant i\leqslant m$,$1\leqslant j\leqslant s+1$ 和 $j\neq l_v+1$.所以,标识 M_{v+1} 能够模拟格局 c_{v+1}.

因此,混合语义时间 Petri 网能够模拟寄存器机.证毕. □

由于寄存器机是图灵等价的,因此 M-TPN 具备图灵机的模拟能力.下面讨论 M-TPN 的标识可达性问题的不可判性.

定义 14. 令 $RS(M_0)$表示从初始标识 M_0 可达的所有标识的集合.给定 M-TPN=$(P,T,Pre,Post,M_0;SI)$和标识 M.标识可达性问题是指标识 M 是否能从初始标识 M_0 可达,即 $M\in RS(M_0)$.

定理 4. M-TPN 的标识可达性问题是不可判定的.

证明:由定理 3 可知,RM 停机当且仅当 RM 到达格局 $c=(l,r_1,\ldots,r_m)$且 $l=s+1$,当且仅当 M-TPN 到达标识 M 且 $M(p_{s+1})=1$.已经证明,寄存器机的停机问题是不可判定的,且 Petri 网的标识可达性问题和子标识可达性问题是递归等价的[12].所以,标识可达性问题对 M-TPN 是不可判定的.证毕. □

3.4 状态类方法

基于变迁实施区间的密度性,时间 Petri 网的状态空间通常是无限的,因此对模型的分析是非常困难的,这就需要一种有限的结构来描述这无限的空间.Berthomieu 等人首先提出经典状态类方法[2],将从初始状态出发实施相同变迁序列达到的所有状态聚合为一个状态类,因此时间 Petri 网的时间计算主要采用状态类方法.为了更有效地计算调度序列的全局实施时间,我们在经典状态类方法的实施域中引入两个特殊的参考变迁:t_0 和 t_*,其中:t_0 的全局实施时间为 0,称为零变迁;t_*表示最近一次被实施的变迁.由于实施变迁不计时间,t_*相对于 t_0 的实施时间即为当前状态类的全局到达时间,当然也是(到当前状态类为止的)调度序列的实施时间.因此,不需要额外构造全局集.

定义 15. 一个状态类是一个二元组 $C=(M,D)$,其中:

(1) $M:P\rightarrow N$ 是标识;

(2) $D:\mathcal{T}\times\mathcal{T}\rightarrow IR$ 是实施域,这里,$\mathcal{T}=T\cup\{t_*,t_0\}$.

$\forall t,t'\in En(M)$,$D(t',t)$表示变迁 t 相对于变迁 t'的实施时间间隔,$D(t_*,t)$表示变迁 t 相对于状态类 C 的实施时间间隔,$D(t_0,t)$则表示变迁 t 的全局实施时间间隔,$D(t_0,t_*)$则表示状态类 C 的全局到达时间间隔.初始状态类为:$C_0=(M_0,D_0)$,其中,M_0 是初始标识;$\forall t,t'\in En(M_0)$,$D(t,t)=0$,$D_0(t',t)=SI(t)-SI(t')$,$D_0(t_0,t)=SI(t)$且 $t_*=t_0$.

下面给出 M-TPN 的状态类演化规则,即可实施条件和实施规则.

定义 16(可实施条件). 一个变迁 t_f 在状态类 $C_k=(M_k,D_k)$是可实施的,当且仅当同时满足下列条件:

(1) $t_f\in En(M_k)$;

(2) $\uparrow D_k(t_*,t_f)\geqslant 0$;

(3) $\forall t\in En(M-Pre(t_f))$,$\uparrow D_k(t_f,t)\geqslant 0$.

条件(1)保证 t_f 是使能的,条件(2)保证 t_f 在状态类 C_k 不是过期变迁,条件(3)保证 t_f 在每个与 t_f 不冲突的变迁之前实施.用 $Fr(C_k)$表示在 C_k 可实施的所有变迁的集合.

定义 17(实施规则). 假设从状态类 $C_k=(M_k,D_k)$实施变迁 $t_f\in Fr(C_k)$后到达状态类 $C_{k+1}=(M_{k+1},D_{k+1})$,则状态类 C_{k+1} 可由下列实施规则确定:

(1) M_{k+1}:

$$M_{k+1}=M_k-Pre(t_f)+Post(t_f).$$

(2) D_{k+1}.令 $t_*=t_f$,且$\forall t_i,t_j\in En(M_{k+1})\cup\{t_0,t_*\}$:

© 中国科学院软件研究所 http://www.jos.org.cn

$$\text{(i)}\quad D_{k+1}(t_i,t_*)=\begin{cases}0, & \text{if } t_i=t_*\\ -SI(t_i), & \text{if } t_i\in Newly(M_k,t_f).\\ [\downarrow D_k(t_i,t_f),DMIN(M,t_f)], & \text{otherwise}\end{cases}$$

这里,$DMIN(M,t_f)=\min\{\uparrow D_k(t_i,t_f),\uparrow D_k(t_i,t)|t\in En(M-Pre(t_f))\}$.

$$\text{(ii)}\quad D_{k+1}(t_i,t_j)=\begin{cases}0, & \text{if } t_i=t_j\\ D_{k+1}(t_i,t_*)-D_{k+1}(t_j,t_*), & \text{if } t_i,t_j\in Newly(M_k,t_f).\\ D_k(t_i,t_j)\cap(D_{k+1}(t_i,t_*)-D_{k+1}(t_j,t_*)), & \text{otherwise}\end{cases}$$

规则(1)计算新标识 M_{k+1}.规则(2.i)计算变迁相对于当前状态类 C_{k+1} 的实施间隔;规则(2.ii)计算任何两个变迁之间的相对实施间隔.

4 一个柔性制造系统的调度分析

下面以一个典型的柔性制造系统(FMS)为例,分析和比较 3 种语义模型下 FMS 的调度问题.我们考虑一个 FMS,它由 3 台机器 M_1,M_2 和 M_3 组成,能够处理两种类型的作业 J_1 和 J_2,每种作业均需要两道工序才能完成.为方便起见,用 $O_{i,j,k}$ 表示作业 i 的工序 j 在机器 k 上操作.表 1 描述了每种作业的资源需求.例如,作业 J_1 的第 1 道工序可以在机器 M_1 操作,也可以在机器 M_3 上操作,其相应的操作记为 $O_{1,1,1}$ 和 $O_{1,1,3}$.根据上面的结构和需求描述,我们建立该柔性制造系统的时间 Petri 网模型,如图 4 所示.其库所和变迁的详细说明在表 2 中描述.

Table 1 Resource requirements for jobs

表 1 作业的资源需求

Process	J_1	J_2
1	M_1/M_3	M_1/M_2
2	M_2/M_3	$M_1/M_2/M_3$

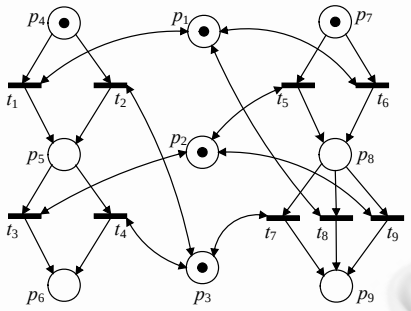


Fig.4 TPN model of the FMS

图 4 柔性制造系统的 TPN 模型

Table 2 Legend for Fig.4

表 2 图 4 结点描述

Place	Description	Transition	Description	Interval
p_1	Machine 1	t_1	Operation $O_{1,1,1}$	[3,6]
p_2	Machine 2	t_2	Operation $O_{1,1,3}$	[5,7]
p_3	Machine 3	t_3	Operation $O_{1,2,2}$	[4,6]
p_4	A part for the first process of Job 1	t_4	Operation $O_{1,2,3}$	[2,3]
p_5	A part for the second process of Job 1	t_5	Operation $O_{2,1,2}$	[4,5]
p_6	A part after the second process of Job 1	t_6	Operation $O_{2,1,1}$	[2,2]
p_7	A part for the first process of Job 2	t_7	Operation $O_{2,2,3}$	[3,5]
p_8	A part for the second process of Job 2	t_8	Operation $O_{2,2,1}$	[1,4]
p_9	A part after the second process of Job 2	t_9	Operation $O_{2,2,2}$	[5,7]

© 中国科学院软件研究所 http://www.jos.org.cn

表 3 给出了该 FMS 模型在 3 种语义模型中的调度情况数据.从调度序列数可以看出:S-TPN 调度范围远远不及其他语义模型,它的调度序列数还不到 M-TPN 和 W-TPN 模型的调度范围的 1/8.由于 S-TPN 的调度范围较小,因此通常很难分析出系统的最优调度.尽管 W-TPN 具有最大的调度范围,但是在该模型的调度序列中会产生一些过期的导致任务中断的调度,这些调度破坏了时限性原则,对实时系统来说是非常致命的问题.从每个结点的平均生成时间可以看出,W-TPN 生成单结点的执行时间比 S-TPN 和 M-TPN 要小一个数量级,而 S-TPN 和 M-TPN 处于一个数量级.这说明 W-TPN 的可调度性计算的时间复杂度更低,而 S-TPN 和 M-TPN 基本持平.

Table 3 Scheduling analysis of the FMS in three different semantics models

表 3 FMS 在 3 种语义模型中的调度情况

Item	S-TPN	M-TPN	W-TPN
State class nodes	24	131	178
Schedules	10	81	112
Optimal schedule	$t_6\ t_1\ t_4\ t_8$	$t_1\ t_5\ t_8\ t_4$	$t_1\ t_5\ t_8\ t_4$
Interval of optimal schedule	[7,9]	[5,8]	[5,8]
Generation time of all nodes (second)	0.0313	0.2031	0.0372
Average generation time of each node (second)	1.304×10^{-3}	1.550×10^{-3}	2.090×10^{-4}

从以上分析可知,M-TPN 在很大程度上扩展了 S-TPN 的调度范围,同时又确保不会出现 W-TPN 中调度中断而影响调度时限性的现象.这得益于 M-TPN 排除了冲突变迁时间约束对变迁可调度性产生的影响,同时又在每个非冲突变迁的上界处设置强制实施点,这样既能增大模型的可调度范围,又可保证任务调度的时限性.因此,M-TPN 是一种适合于可调度分析的时间 Petri 网语义模型.

5 结 论

时间 Petri 网主要采用两种时间语义:强语义和弱语义,但它们在调度分析上都存在各自的缺陷.本文提出混合语义模型,通过调整可实施条件和设置强制实施点,兼顾了强、弱语义的特点,解决了现有语义模型的调度分析缺陷.本文还证明了混合语义模型的图灵等价性、标识可达性问题的不可判定性等重要的理论问题,给出了适合全局延时计算的状态类方法,形成了较为完善的理论体系.此外,我们基于 Matlab 实现了 3 种语义模型的时间 Petri 网工具箱,可应用于实际实时系统的建模及调度分析.表 4 总结了 3 种语义模型在时间约束强度、调度范围、图灵机模拟能力、可调度性计算复杂度、时间语言接受和过期极大冲突集这 6 个方面的比较结果.

Table 4 Comparison of three semantics models

表 4 3 种语义模型的比较分析

Item	S-TPN	M-TPN	W-TPN
Time constraint strength	The strongest	The second strongest	The weakest
Scheduling scope	The smallest	The second largest	The largest
Turing power	Yes	Yes	No
Time complexity of schedulability	$O(\|T\|)$	$O(\|T\|)$	$O(1)$
Over-Due maximal conflict set	No	No	Maybe
Timed language acceptance	S-TPN$\subseteq_L$M-TPN$\subseteq_L$W-TPN		

进一步工作将探讨 3 种语义模型关于时间互相似性(timed bisimilarity)的比较结论.此外,混合语义模型在典型实时系统(比如嵌入式实时系统、实时工作流系统等)中的应用也是未来研究的重点.

References:

[1] Merlin PM, Farber DJ. Recoverability of communication protocols implications of a theoretical study. IEEE Trans. on Communications, 1976,24(9):1036−1043. [doi: 10.1109/TCOM.1976.1093424]

[2] 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]

[3] Cerone A, Maggiolo-Schettini A. Time-Based expressivity of time Petri nets for system specification. Theoretical Computer Science, 1999,216(1-2):1−53. [doi: 10.1016/S0304-3975(98)00008-5]

© 中国科学院软件研究所 http://www.jos.org.cn

[4] Boyer M, Roux OH. Comparison of the expressiveness of arc, place and transition time Petri nets. In: Proc. of the 28th Int'l Conf. on Application and Theory of Petri Nets and other Models of Concurrency. LNCS 4546, Berlin: Springer-Verlag, 2007. 63–82. [doi: 10.1007/978-3-540-73094-1_7]

[5] Felder M, Mandrioli D, Morzenti A. Proving properties of real-time systems through logical specifications and Petri net models. IEEE Trans. on Software Engineering, 1994,20(2):127–141. [doi: 10.1109/32.265634]

[6] Koutny M. A compositional model of time Petri nets. In: Proc. of the 21st Int'l Conf. on Application and Theory of Petri Nets. LNCS 1825, Berlin: Springer-Verlag, 2000. 303–322. [doi: 10.1007/3-540-44988-4_18]

[7] Esparza J. Decidability and complexity of Petri net problems—An introduction. In: Proc. of the Lectures on Petri Nets I: Basic Models. LNCS 1491, 1998. 374–428. [doi: 10.1007/3-540-65306-6_20]

[8] Berthomieu B, Lime D, Roux OH, Vernadat F. Reachability problems and abstract state spaces for time Petri nets with stopwatches. Discrete Event Dynamic System, 2007,17(2):133–158. [doi: 10.1007/s10626-006-0011-y]

[9] Alain RP, Arnaud S. Weak time Petri nets strike back. In: Proc. of the 20th Int'l Conf. on Concurrency Theory. LNCS 5710, Berlin: Springer-Verlag, 2009. 557–571. [doi: 10.1007/978-3-642-04081-8_37]

[10] Wang JC, Deng Y, Xu G. Reachability analysis of real-time systems using time Petri nets. IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, 2000,30(5):725–736. [doi: 10.1109/3477.875448]

[11] Vicario E. Static analysis and dynamic steering of time dependent systems using time Petri nets. IEEE Trans. on Software Engineering, 2001,27(8):728–748. [doi: 10.1109/32.940727]

[12] Peterson JL, Wrote; Wu ZH, Trans. Petri Net Theory and the Modeling of Systems. Xuzhou: China University of Mining and Technology Press, 1989 (in Chinese).

附中文参考文献:

[12] Peterson JL,著;吴哲辉,译.Petri 网理论与系统模拟.徐州:中国矿业大学出版社,1989.

潘理(1975－),男,湖南平江人,博士,讲师,CCF 会员,主要研究领域为 Petri 网,工作流.

郭观七(1963－),男,博士,教授,主要研究领域为智能计算,并行算法.

丁志军(1974－),男,博士,副教授,CCF 高级会员,主要研究领域为 Petri 网,服务计算.

© 中国科学院软件研究所 http://www.jos.org.cn