

文章编号: 1001-0920(2008)09-1004-07

基于对象的扩展 Petri 网协同设计过程分析

安毅生^{1,2}, 李人厚¹

(1. 西安交通大学 系统工程研究所, 西安 710049; 2. 长安大学 信息工程学院, 西安 710064)

摘要: 针对协同设计中任务的执行流程缺乏柔性, 不利于分析实际设计过程的现状, 提出一种单元调用变迁对与决策变迁相集成的基于对象的扩展 Petri 网, 扩展了 Petri 网的可达图以适应分析 OEPNs 模型. 采用 OEPNs 中的过程网和单元网对协同设计过程建模, 利用模型中的单元调用变迁对和决策变迁对过程本身和可能状态进行分析. 最后与相关的研究工作进行比较并给出了结论.

关键词: 协同设计; Petri 网; 决策变迁; 单元调用变迁对

中图分类号: TP391 **文献标识码:** A

Analysis of cooperative design process using object-based extended Petri nets

AN Yi-sheng^{1,2}, LI Ren-hou¹

(1. System Engineering Institute, Xi'an Jiaotong University, Xi'an 710049, China; 2. School of Information Engineering, Chang'an University, Xi'an 710064, China. Correspondent: AN Yi-sheng, E-mail: aysm@chd.edu.cn)

Abstract: A cooperative design task is lack of the flexibility of executing flow, which is not suitable for analyzing practical design process. An object-based extended Petri nets which integrates the units call transition pairs and decision transition is proposed. The reachable graph of traditional Petri nets is extended to meet the analysis of OEPNs. The cooperative design process is modeled by using process nets and unit nets, its properties and potential states are analyzed by using units call transition pairs and decision transition. Finally, the research results are compared with other works proposed in literature and the corresponding evaluation is given.

Key words: Cooperative design; Petri nets; Decision transition; Units call transition pairs

1 引言

随着面向对象技术的发展, 对象 Petri 网正在成为 Petri 网理论和应用的研究热点, 并取得了一些初步成果. 文献 [1] 通过分析和比较赋色 Petri 网、分层赋色 Petri 网和面向对象 Petri 网所实现的电子数据交换系统的异同, 指出多层活动复杂系统的最佳描述方法应按不同的抽象层次进行建模. [2] 将继承、多态和动态绑定等机制与 Petri 网相集成, 提出了统一的类层次结构, 即托肯与子图均以类进行表示. [3] 提出的 LOOPN++, 是一种支持类层次结构和 PN 形式化方法的文本语言, 其中的类封装了数据、方法和行为, 突破了仅以网表示行为的限制. 为了建模和研究复杂并行、分布式的面向对象系

统, Milan 等开发了基于面向对象 Petri 网的 PNtalk^[4], 受 SmallTalk 语言的影响, PNtalk 支持对象技术、消息传递、并发和同步机制. [5] 采用基于状态空间的方法, 研究了 OOPN 模型的形式化分析及验证. 在应用方面, [6] 采用 LOOPN++ 建立了网络协议的对象 Petri 网模型, 结果表明对象 Petri 网能有效地克服基本 Petri 网建模中缺乏合成能力的弱点. [7] 采用 PNtalk 构建了协作编辑器的 OOPN 模型, 首先给出其对象模型, 然后转化为 OOPN 模型, 该模型可用于仿真、构建原形系统及分析和验证. [8] 提出了面向对象高级 Petri 网 (OPNets) 的概念, 并应用于实时系统的建模. [9] 在此基础上引入调度分派控制逻辑, 解决了柔性制造

收稿日期: 2007-05-18; 修回日期: 2007-09-21.

基金项目: 国家自然科学基金项目 (60473136).

作者简介: 安毅生 (1972—), 男, 西安人, 博士生, 从事计算机支持协同工作、 workflow 技术的研究; 李人厚 (1935—), 男, 浙江宁波人, 教授, 博士生导师, 从事智能控制、CSCW 理论与应用等研究.

系统中模型预测和分析系统时可能出现的死锁和冲突事件.

上述文献表明,对象 Petri 网在表示复杂的异步、并发系统方面是有效的.上述方法主要以子网的形式表示对象,在子网变迁之间以库所的形式保存对象间传递的消息;无论是协议分析还是柔性制造系统建模,对象 Petri 网建立的控制流程仍是固定的,缺乏对不同任务的适应性.就协同设计过程而言,设计人员、设计任务面临着不确定性,且大量设计活动需要反复执行或跳跃执行,特别是运行时过程必须可调.对于这类问题,现有方法还是不能很好地解决.

本文把工作流的分支与软件工程中的对象集成到 Petri 网中,提出了基于对象的扩展 Petri 网(OEPNs),建立了过程网与单元网的调用关系,扩展了基本 Petri 网的可达图,以适应 OEPNs 的新需求;在协同设计过程 OEPNs 模型的基础上,采用对象化技术缩减网络规模,阐述了 OEPNs 的单元调用变迁对、决策变迁对对协同设计过程控制的影响;最后与相关研究作了比较.

2 OEPNs 的定义

Petri 网的基本概念、符号及性质参见文献 [10]. 本文在此基础上定义了基于对象的扩展 Petri 网,部分符号如表 1 所示. 扩展 Petri 网由过程网和单元网两部分构成,如图 1 所示.

表 1 扩展 Petri 网的符号及说明

符号	说明
wPN	OEPNs 的过程网
UN _i	OEPNs 第 i 个单元网
t _{uc} , T _{uc}	过程网的单元调用变迁及集合
t _{dc} , T _{dc}	过程网的决策变迁及集合
	简单触发序列
	简单触发序列表达式
W(t _{uc} , p) UN _i	为 t _{uc} 后集中添加含 UN _i 实例的托肯
W(p, t _{dc}) UN _i	从 t _{dc} 前集中移出含 UN _i 实例的托肯
t _{i,call}	单元网入口变迁,与 t _{uc} 构成单元调用变迁对
t _{i,return}	单元网出口变迁,与 t _{uc} 构成单元调用变迁对
	复合触发序列表达式

定义 1 过程网用六元组表示为

$$wPN = (P, T, F, D, V, E).$$

其中: P 为库所的有限集合, P = P_{or} ∪ P_{cb}, P_{or} 和 P_{cb} 分别表示普通库所和对象库所; T 为变迁的有限集合, T = T_{or} ∪ T_{uc} ∪ T_{dc} (T_{or} ∪ T_{uc} ∪ T_{dc} = ∅), T_{or}, T_{uc} 和 T_{dc} 分别表示过程网的普通变迁、单元调用变迁和决策变迁; F ⊆ (P × T) (T × P) 表示过

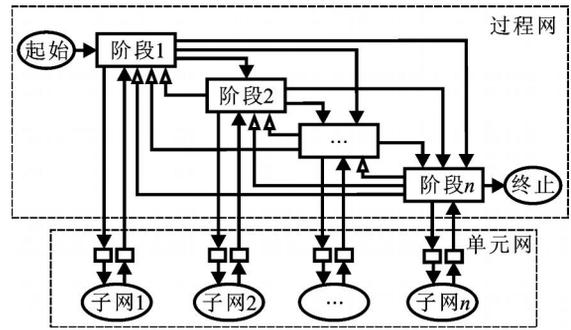


图 1 OEPNs 的构成

程网的弧集合,也称流关系; D 表示决策变迁对应的决策函数集合; V 表示弧表达式中变量的集合; E F Bags(V) 表示弧的标记函数.

定义 2 单元网用二元组表示为

$$UN_i = (P_i, T_{i,o}).$$

其中: P_i 为 P/T 网, T_{i,o} = {t_{i,call}, t_{i,return}} 表示单元网的入口和出口.

定义 3 OEPNs 用四元组表示为

$$OEPNs = (wPN, \{UN_i\}_{i \in N}, S, IM).$$

其中:wPN 表示过程网; UN_i 是第 i 个单元网; S 表示过程网与单元网的调用关系,由调用变迁对和返回变迁对组成,统称为单元调用变迁对; IM 表示初始标识.

定义 4 OEPNs 的初始标识表示为

$$IM = (m_{wPN}; m_{un}).$$

其中:m_{wPN} 表示过程网初始标识, m_{un} 表示单元网初始标识.

定义 5 过程网与单元网调用关系表示为

$$S = \{T_i \times T_j \mid \exists f \in F, f \in \{0,1\}, T_i = \bigcup_{i=1}^k T_i\}.$$

其中:1 表示对应变迁间存在调用(返回)关系,0 表示不存在调用(返回)关系.

定义 6 决策变迁表示为

$$T_{dc} = \{f \mid (\exists \text{Token in Pre}(t)) f(\text{Token}) \in E\},$$

其中 E ⊆ E 是决策变迁 t_{dc} 到它的后集库所的弧表达式集合.

在 OEPNs 中引入单元调用变迁对和决策变迁,大大丰富了变迁的语义,这种增强了表达能力的 Petri 网适合于具有多组活动的并发系统的建模.并发的各个活动组分别用单元网表示,活动组之间的关系用过程网描述.定义 5 中的调用关系和定义 6 中的决策变迁不同于普通的变迁,按以下规则触发:

规则 1 单元调用变迁对的触发. 设 (m_{wPN}; m_{un}) 是 OEPNs 的初始标识, t_{uc} ∪ T_{uc} 是过程网调用变迁, t_{i,call} ∪ T_i 是单元网 UN_i 的变迁. 则单元调用

变迁对 $(t_{uc}, t_{i,call})$ 的使能需满足以下条件:

1) $m_{wPN}(p) \geq W(p, t_{uc}), \forall p \in \text{Pre}(t_{uc}) \cap P$, 即 t_{uc} 的前集中托肯数目大于等于弧表达式上标记的托肯数目;

2) $(t_{uc}, t_{i,call})$ 触发后,不但为 $\text{Post}(t_{uc})$ 生成托肯,而且创建 $t_{i,call}$ 所在单元网的实例并触发 $t_{i,call}$. 具体如下:

$$\forall p \in \text{Pre}(t_{uc}) \cap P, m_{wPN}(p) = m_{wPN}(p) - W(p, t_{uc});$$

$$\forall p \in \text{Post}(t_{uc}) \cap P, m_{wPN}(p) = m_{wPN}(p) + W(t_{uc}, p) \cup N_i;$$

$$\forall p_i \in \text{Post}(t_{i,call}) \cap P_i, m_{un}(p_i) = m_{un}(p_i) + W_i(t_{i,call}, p_i).$$

类似地,可推导出 $(t_{uc}, t_{i,return})$ 的触发规则如下:

规则 2 决策变迁的触发. 设 $(m_{wPN}; m_{un})$ 是 OEPNs 的初始标识, T_{de} 是过程网的决策变迁,并且不存在某个单元网变迁与之构成单元调用变迁对. 则 $m_{wPN}(p) \geq W(p, t_{de}) \cup N_i$, 有唯一的 $p \in \text{Pre}(t_{de}) \cap P$ 成立; 决策变迁 t_{de} 触发后,库所标记更新如下:

$$1) m_{wPN}(p) = \begin{cases} m_{wPN}(p) - W(p, t_{de}) \cup N_i, \\ \text{有唯一的 } p \in \text{Pre}(t_{de}) \cap P; \\ m_{wPN}(p) + W(t_{de}, p) \cup N_i, \\ \exists p \in \text{Pre}(t_{de}) \cap P. \end{cases}$$

$$2) m_{un} = m_{un}.$$

根据定义 5,单元网的入口表明单元网只能通过外部调用触发,出口表明单元网的运行最终将影响外部系统. 只要在入口和出口变迁之间添加哑元库所及相应的弧,便可得到一个封闭的 Petri 网系统,用以分析其活性、公平性等性质. 但在过程网中,上述性质必须依赖于决策变迁.

定义 7 设 OEPNs 的触发序列 $\sigma = (t_1, t_2, \dots, t_k)$, 若 $\forall t_i \in \sigma, t_i \in (T_{de} \cup T_{uc})$, 则称 σ 为简单触发序列. 即仅由单元网或过程网普通变迁触发所形成的序列.

定义 8 设 $\sigma_1, \sigma_2, \dots, \sigma_n$ 分别为简单触发序列, 定义连接运算 $(+)$ 和幂运算 $(^*)$, 则简单触发序列表达式按如下规则构造: 1) σ_i (简单触发序列); 2) $\sigma_i + \sigma_j$ (简单触发序列的连接运算); 3) $\sigma_i + (\sigma_k)^*$ (简单触发序列的循环运算).

定义 9 设 $\sigma_1, \sigma_2, \dots, \sigma_n$ 为简单触发序列表达式, $T_{uc} \times (T_i \setminus \{t_i\})$ 表示单元调用变迁对集合, T_{de} 为决策变迁集合. 复合触发序列表达式按下述规则构造: 1) $\sigma_k / t_{de} / (t_{uc}, t_{i,call}) / (t_{uc}, t_{i,return})$, 即可表示为简单触发序列表达式、决策变迁或单元调用变迁; 2) $(\sigma_k + t_{de}) / (t_{de} + \sigma_k) / (\sigma_k + t_{uc},$

$t_{i,call}) / ((t_{uc}, t_{i,return}) + \sigma_k)$, 即可表示为决策变迁或单元调用变迁对与简单触发序列表达式的连接.

定义 10 设 t_{de} 为决策变迁, p 为 t_{de} 的后集库所之一. 称 p 是饥饿的, 当且仅当包含 t_{de} 的任意有限复合触发序列, 使得 $m(p) = 0$.

定义 11 设 t_{de} 为决策变迁, p 为 t_{de} 的后集库所之一. 称 t_{de} 是半活性的, 当且仅当包含 t_{de} 的任意复合触发序列, 使得 p 饥饿.

定义 12 设 wPN 为 OEPNs 的过程网, t_{de} 为决策变迁. 称 σ_{de} 为 wPN 的含 t_{de} 子网, 当且仅当: 1) $p \subseteq P$; 2) $T \subseteq T \cup t_{de} \cup T$; 3) $F = F / p \times T \cup T \times p$. 称 σ_{de} 为 wPN 的最小完全含 t_{de} 子网, 当且仅当: 1) σ_{de} 是 wPN 的一个含 t_{de} 子网; 2) 以 t_{de} 的后集库所作为入集的变迁必须包含于 σ_{de} ; 3) t_{de} 的前集库所包含于 σ_{de} ; 4) 包含 t_{de} 和 t_{de} 前集库所及 t_{de} 某一库所的闭合链路上, 所有库所和变迁也必须包含于 σ_{de} .

命题 1 已知最小完全含 t_{de} 子网 σ_{de} , 若 t_{de} 是半活性的, 则该子网是非公平和非活的.

证明 (反证法) 已知 t_{de} 是半活性的, 则 t_{de} 会使其某个后集库所处于饥饿状态. 设 p 便是这样一个库所, 且处于含 t_{de} 的反馈回路上.

1) 假设 σ_{de} 是活的, 则 $\forall m \in R(m_0), \exists m \in R(m)$, 使得 $\forall t \in \sigma_{de}$, 有 $m[t]$. 已知 p 处于饥饿状态, 它意味着 p 的后集变迁无法触发, 这与假设矛盾, 因此 σ_{de} 是非活的.

2) 假设 σ_{de} 是公平的, 按照公平性的定义, $\forall t_i, t_j \in \sigma_{de}$, 对于任意复合触发序列 σ , t_i 不触发时 t_j 最多被触发有限次. 如果由于 t_{de} 的原因, 仅在位于反馈回路上的库所 p 置入托肯, 则 $\exists \sigma^*$, 使得 $\#(\sigma, t) = 0, \#(\sigma, t_{de}) = 1$, 其中 t 为不在反馈回路上且以 t_{de} 的后集库所为前集的变迁. 显然这与假设矛盾, 因此 σ_{de} 是非公平的.

命题 2 如果决策变迁 t_{de} 不导致饥饿, 则最小完全含 t_{de} 子网复合触发序列表达式不唯一.

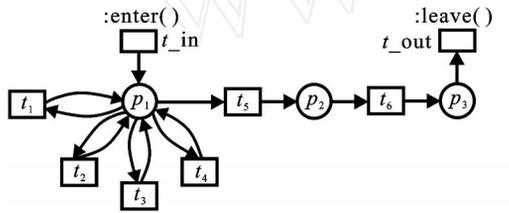
证明 设 σ_{de} 为 wPN 的最小完全含 t_{de} 子网, t_1 和 t_2 分别是以 t_{de} 的后集库所为前集库所的变迁, 且 t_1, t_2, t_1 位于 t_{de} 的反馈分支上, t_2 位于 t_{de} 的正向推进分支上. 由于 t_{de} 不导致饥饿, 则 t_1 和 t_2 均有可能触发, 且 t_2 一定会触发. 如果 t_2 触发之前 t_{de} 触发一次以上, 则最小完全含 t_{de} 子网复合触发序列表达式 $\sigma_1 = (\sigma_{k1} + t_{de} + t_1 + \sigma_{k2})^* + t_2$, 其中 σ_{k1} 为过程网上决策变迁触发之前的简单触发序列表达式, σ_{k2} 为反馈分支的简单触发序列表达式. 如果 t_2 触发之前 t_{de} 仅触发一次, 则 $\sigma_2 = (\sigma_{k1} + t_{de}) + t_2$. 显然 $\sigma_1 \neq \sigma_2$, 说明最小完全含 t_{de} 子网复合触发序列表达式不唯一.

3 基于 OEPNs 的协同设计过程模型分析

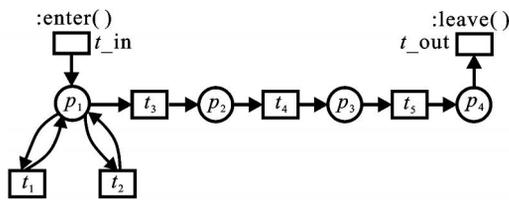
3.1 协同设计过程描述

本节以一个由规范设计 (SD), 概念设计 (CD) 和详细设计与优化 (DDO) 所构成的设计过程为例, 说明 OEPNs 的建模和分析方法.

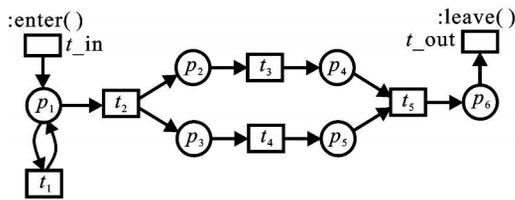
首先以规范设计为例说明 OEPNs 的单元网. 规范设计阶段包括确定客户需求、关键问题、产品功能需求、设计目标、实现方式、系统噪声等活动, 其单元网如图 2(a) 所示. 入口变迁为 t_{in} , t_{in} 的触发将为库所 p_1 生成表示规范设计信息的托肯, 进而由变迁 $t_1 \sim t_4$ 所表示的确定客户需求、确定关键问题、确定产品功能需求和设计目标, 可有选择地并发执行; 由变迁 t_5 和 t_6 所表示的确定实现方式和系统噪声, 必须顺序执行. 最终为库所 p_3 生成表示完成规范设计的托肯, 通过变迁 t_{out} 的触发返回过程网. 图 2(b)



(a) 规范设计



(b) 概念设计



(c) 详细设计与规划

图 2 SD, CD 和 DDO 的单元网

表 2 变迁的语义

单元网	t_1	t_2	t_3
图 2(a)	客户需求	关键问题	产品功能需求
图 2(b)	基本设计任务	产品功能结构	学科知识
图 2(c)	任务分解	任务分配	鲁棒性设计
单元网	t_4	t_5	t_6
图 2(a)	设计目标	实现方式	系统噪声
图 2(b)	创新设计	可行解空间	
图 2(c)	优化设计	成本设计	

和 (c) 分别表示概念设计和详细设计与优化阶段的单元网, 其中变迁的语义如表 2 所示.

OEPNs 过程网建模的设计过程如图 3 所示, 它由一系列对单元网的调用 (返回) 和决策变迁共同构成. 其中: 变迁 t_1 和 t_2 实现对规范设计单元网的调用 (返回); 类似地, t_7 和 t_8 , t_{13} 和 t_{14} 分别实现对概念设计和详细设计与优化单元网的调用 (返回). 图中标记黑色的变迁 $t_{d1} \sim t_{d3}$ 为决策变迁, 用于表示对 3 个阶段性设计结果的评审.

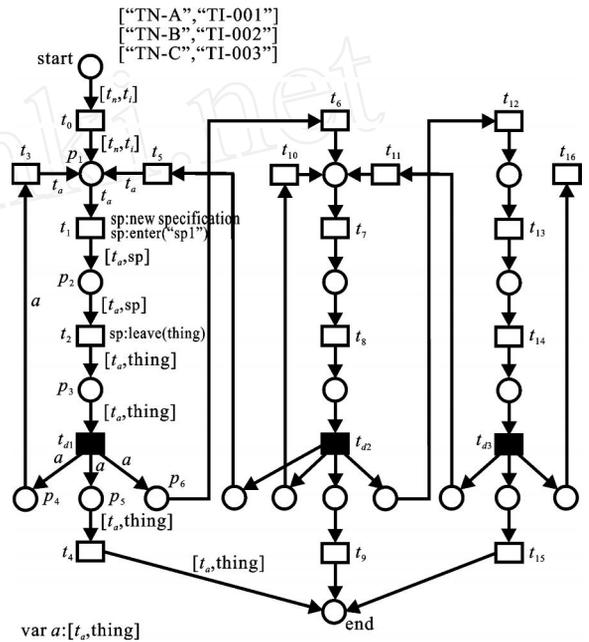


图 3 过程网实例

过程网的具体执行过程如下: 变迁 t_0 触发为库所 p_1 中生成表示等待开始规范设计的托肯, 这样由图 3 中 t_1 和图 2(a) 中 t_{in} 构成的单元调用变迁对 (t_1, t_{in}) 使能. 一旦 (t_1, t_{in}) 触发, 便创建 t_{in} 所属单元网的实例并传递参数, 图 3 中 p_2 会出现一个表示正在执行规范设计任务的托肯. 同时, 图 2(a) 的 p_1 中也会生成一个表示规范设计信息的托肯. 单元网运行结束后, 通过由图 3 中 t_2 和图 2(a) 中 t_{out} 构成的单元调用变迁对 (t_2, t_{out}) 返回过程网. (t_2, t_{out}) 的触发将为图 3 中 p_3 生成一个表示待评审规范设计信息的托肯. 决策变迁 t_{d1} 表示上述结果的评审: 如果评审合格且该任务只需执行规范设计即可, 则在 p_5 中生成托肯, 随着 t_4 的触发, 设计任务终止; 如果评审合格且该任务还要完成后续概念设计过程, 则在 p_6 中生成托肯, 随着变迁 t_6 的触发, 进入概念设计阶段; 如果评审不合格, 则在 p_4 中置入表示重新执行规范设计的任务, 随着 t_3 的触发, 启动该任务的重复执行.

3.2 OEPNs 的分析工具

基本 Petri 网的可达状态还可用可达树、可达图

和库所的共享,它通过过程网的调用(返回)和决策变迁建立联系.从过程网的层面看,它表示的是一种流程,这类流程的本质特征是进展性,因此可认为过程网不会也不应陷入死锁状态.但在单元网内部组织设计活动时,应避免库所的前集包含于后集的结构.一旦控制流程进入这种结构,会使单元网陷入死锁状态,进而导致整个设计活动的中止.

尽管过程网不会陷入死锁状态,但可能处于活锁状态.活锁是由 OEPNs 反映的一种特殊的系统状态,即控制流程始终保持在某个阶段(某个单元网),不能推进到下一个阶段.它是由该阶段对应的单元网及其后决策变迁共同造成的.为了判断设计过程是否处于活锁状态,首先必须记录该过程对应的 OEPNs 变迁触发序列;然后从它对应的 OEPNs 决策变迁入手:1) 求出最小完全含 t_{de} 子网;2) 计算该子网的可达图;3) 根据可达图弧上的标识,计算可能的复合触发序列表达式,进而判断当前状态是否为活锁.

以图 5 为例,它是图 3 中 t_{d1} 的最小完全含 t_{d1} 子网,该子网的动态行为完全依赖于决策变迁 t_{d1} ,图 4 为该子网的可达图.若该子网任意时刻的复合触发序列表达式总表示为 $(t_1 + t_2 + t_{d1} + t_3)^n$,其中 t_1, t_2, t_3 是图 2(a) 规范设计单元网的简单触发序列,则 t_{d1} 是半活性的且仅为 t_3 的前集库所生成托肯.因此,包含该子网的 OEPNs 所反映的协同设计过程正处于活锁状态.根据命题 1,可推知处于活锁状态的设计过程的 OEPNs 中,一定存在不公平的含 t_{de} 子网,即陷入活锁状态的含 t_{de} 子网无限次地触发它所调用的单元网.

由于设计人员能力和设计任务难度的不确定性,实际的工作流程中包含了分支结构,这种分支结构只有在系统动态执行时才能确认具体流向.当某个设计阶段完成后,除了正常的任务推进外,还存在重复执行和跳转执行的可能,而且转移的具体位置只能依赖于当时任务的执行.决策变迁正是对这种可能路由选择的建模.如果设计过程陷入活锁状态,即某个单元网被反复执行且无法得到一个能通过评审的结果,则说明被调用的单元网不适合该任务,通过对其再分析,可重新制定该任务执行时的活动依赖关系,进而从相应的单元网簇中选择能反映这种新关系的单元网.在 OEPNs 中,通过把决策变迁与单元网簇相结合,能较好地解决运行时设计过程可调的问题.

基于以上分析,进一步可得如下结论:

1) 协同设计过程不应存在死锁状态,它要求对应的 OEPNs 模型中单元网是活的,且决策变迁不导

致饥饿.这是对设计过程可推进性的保证.

2) 协同设计过程不能陷入活锁状态,判断条件是决策变迁的最小完全含 t_{de} 子网的复合触发序列表达式不唯一.这是对结论 1) 的细化,它要求每个设计阶段经有限次执行后都能得到所需的中间结果.如果多次执行未果,则可通过调用同一单元网簇中的不同单元网来实现.

3) 在协同设计过程中,每个阶段只能被有限次调用,而不能陷入其中.这就要求对应的 OEPNs 模型是公平的,等价于 OEPNs 模型不能陷入活锁状态.

4 相关工作比较

与过程建模领域两类代表性的方法相比,本文工作具有以下特点:

1) OEPNs 具有更直观的模块化、层次化特征,便于面向对象方法建模.文献[12]用特定变迁替代子网体现了分层的思想,但该变迁仅作为占位符使用,是一种展开表示形式.文献[13]采用时间 Petri 网对并行设计过程建模,但模型过于宏观,无法判断某一阶段中具体活动对总体设计的影响.OEPNs 借助于单元网,通过相关数据和方法的聚合,减少了复杂性,增加了部件的可复用性,提高了协同设计过程的可分析性;同时,单元网与实际对象模型具有更直接的映射关系,易于构建实际系统.

2) OEPNs 支持过程运行时可调.Aalst 提出的工作流网是一种重要的过程建模方法,文献[14]给出了工作流网的定义,分析了 4 类工作流结构及在工作流网中的表示,但 WF-net 只适用于标准过程的建模.相对而言,OEPNs 的决策变迁及同一阶段反映不同活动依赖关系的单元网簇,能很好地支持运行期过程的修正和改变,是解决运行时过程可调的有益探索.

5 结 论

协同设计过程的建模与分析是目前研究的热点.本文工作具体可归纳为:1) 提出一种适用于协同设计过程描述和分析的基于对象的扩展 Petri 网,它由过程网和单元网两部分构成,通过单元调用变迁对建立二者的联系,通过决策变迁及同一阶段反映不同活动依赖关系的单元网簇实现运行时过程可调;2) 扩展了 Petri 网的可达图,以适应 OEPNs 的可达性计算;3) 借助于 OEPNs 分析了协同设计过程及其潜在的状态.本文工作适用于复杂的分布式面向对象系统的描述、建模和分析.

参考文献(References)

[1] Lakos C A. From colored Petri nets to object Petri nets

- [C]. LNCS '935. Torino: Springer-Verlag, 1995: 278-297.
- [2] Lakos C A. Object-oriented modeling with object Petri nets[C]. LNCS '2001. Berlin: Springer-Verlag, 2001: 1-37.
- [3] Keen C D, Lakos C A. Information systems modeling using LOOPN++: An object Petri net scheme[C]. Proc of 4th Int Working Conf on Dynamic Modeling and Information Systems. Netherlands: Delft University Press, 1994: 31-52.
- [4] Milan C, Vladimir J, Tomas V. Modeling, prototyping and verifying concurrent and distributed applications using object-oriented Petri nets[J]. Kybernetes, 2002, 31(9/10): 1289-1299.
- [5] Tomas V. Towards formal analysis and verification over space states of object-oriented Petri nets[D]. Bron: Bron University of Technology, 2001.
- [6] Lakos C A, Lamp J W, Keen C D. Modeling network protocols with object Petri nets[C]. Proc of Workshop on Petri Nets Applied to Protocols. Torino: Springer, 1995: 31-42.
- [7] Radek K, Tomas V. A PNTalk-based model of a cooperative editor[C]. Proc of 35th Spring Int Conf on Modelling and Simulation of Systems. Hradec nad Moravicí, 2001: 165-172.
- [8] Lee Y K, J Park S. OPNets: An object-oriented high level Petri net model for real-time system modeling[J]. J of System Software, 1993, 20(1): 69-86.
- [9] Wang L. Object-oriented Petri nets for modeling and analysis of automated manufacturing systems[J]. Computer Integrated Manufacturing Systems, 1996, 26(2): 111-125.
- [10] Girault C, Valk R. Petri nets for systems engineering: A guide to modeling, verification and applications[M]. Berlin: Springer-Verlag, 2002.
- [11] Murata T. Petri nets: Properties, analysis and application[J]. Proc of IEEE, 1989, 77(4): 541-580.
- [12] 黄洪钟, 刘伟, 李丽. 产品协同设计过程建模研究[J]. 计算机集成制造系统, 2003, 9(11): 955-959.
(Huang H Z, Liu W, Li L. Process modeling of product collaborative design[J]. Computer Integrated Manufacturing Systems, 2003, 9(11): 955-959.)
- [13] 宋玉银, 褚秀萍, 蔡复之. 基于时间 Petri 网的实时并行设计过程建模研究[J]. 计算机集成制造系统, 1999, 5(6): 17-22.
(Song Y Y, Chu X P, Cai F Z. Research of the process modeling of real-time concurrent design based on time Petri net [J]. Computer Integrated Manufacturing Systems, 1999, 5(6): 17-22.)
- [14] Aalst W M P, Hee K M. Workflow management models, methods and systems[M]. London: The MIT Press, 2002.

(上接第 1003 页)

- [11] Ding F, Chen T. Parameter estimation of dual-rate stochastic systems by using an output error method[J]. IEEE Trans on Automatic Control, 2005, 50(9): 1436-1441.
- [12] Ding F, Chen T. Performance analysis of multi-innovation gradient type identification methods[J]. Automatica, 2007, 43(1): 1-14.
- [13] 刘英玉, 申东日, 陈义俊, 等. 基于前向神经网络的多新息随机梯度辨识算法[J]. 哈尔滨商业大学学报, 2006, 22(2): 83-86.
(Liu Y Y, Shen D R, Chen Y J, et al. Multi-innovation stochastic gradient identification algorithm based on feedforward neural networks[J]. J of Harbin University of Commerce, 2006, 22(2): 83-86.)
- [14] Ding F, Chen H B, Li M. Multi-innovation least squares identification methods based on the auxiliary model for MISO systems[J]. Applied Mathematics and Computation, 2007, 186(1): 84-192.
- [15] 丁锋, 萧德云, 丁韬. 多新息随机梯度辨识方法[J]. 控制理论与应用, 2003, 20(6): 870-874.
(Ding F, Xiao D Y, Ding T. Multi-innovation stochastic gradient identification methods[J]. Control Theory and Applications, 2003, 20(6): 870-874.)
- [16] Ding F, Shi Y, Chen T. Auxiliary model based least-squares identification methods for Hammerstein output-error systems[J]. Systems & Control Letters, 2007, 56(5): 373-380.