

道义 Petri 网及其在协同组织社交建模分析中的应用

蔡国永 董荣胜 古天龙

(桂林电子科技大学计算机与控制学院 桂林 541004)

摘 要 在开放动态的、充满异质服务的网络环境下,构建服务协同虚拟组织的一个重要挑战是自治服务代理间的协同交互问题。提出一种扩展的引用网:道义 Petri 网(DPN)。DPN 结合了动态组织建模中的角色规范概念,并采纳了引用网的网中网和同步通道。给出了 DPN 在建模与分析自治服务代理协同交互时的一般方法,并以电子机构模型为例,具体论证了 DPN 在协同交互建模与分析上的优越性及可用性。

关键词 协同交互, Petri 网, 建模, 道义规范

中图法分类号 TP301 文献标识码 A

Deontic Petri Net for Modeling and Analyzing Social Interactions in Collaborative Organizations

CAI Guo-yong DONG Rong-sheng GU Tian-long

(School of Computer and Control, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract Over an open and dynamic networked environment filled with heterogeneous services, a challenge issue to be coped with, whenever constructing services oriented collaborative organizations, is the complicated social interactions among autonomic service agents. An extended reference Petri net (called deontic Petri net, DPN) was proposed, which incorporates the role and norm concepts of organizational modeling with the concepts of net within net and synchronous channel in reference net model. The modeling and analyzing methods toward collaborative interactions among service agents with DPN were presented. An electronic institution model was applied to demonstrate the advantages of DPN and a case study shows the usefulness of the presented approach.

Keywords Collaborative interaction, Petri net, Modeling, Deontic norm

随着面向服务计算技术和先进网络技术(如网格)的快速融合,网络基础环境(如服务网格)上出现了大量可供使用的异质服务,使得如何快速协调这些服务,组建服务协同的虚拟组织成为当前一个新的研究热点。由于在开放网络环境下,服务代理具有质异性、自主性的特点,使得开发服务协同虚拟组织时面临极大的挑战。一是由于参与组织提供服务的代理无法直接控制,因而难以预测和保证虚拟组织任务或目标的安全可靠实现;二是由于参与虚拟组织的服务个体预先无法确定,因而如何协调它们之间的交互,保证服务个体间利益公平存在很大的困难。

虽然面向多主体(或称代理)的软件工程研究提出的方法框架(如 GAIA^[1], AGR^[2], OMINI^[3]等)对于服务协同虚拟组织的开发提供了很好的参考,但现有的这些方法框架也还存在明显不足,如这些方法着重于概念结构的分析建模,缺乏从分析到设计的平滑过渡,缺乏与支持并发同步等机制的形式化技术结合等。Petri 网由于采用库所、变迁、托肯的三元结构,对实体间的并发、通信、同步等交互具有很好的建模能力,同时由于其可视化、可严格的数学分析特性,使得 Petri 网及其变形在柔性生产系统、通信协议及其它并发系统形式建模

与分析方面得到了许多成功的应用^[4]。结合面向多主体组织研究中提出的一些概念(如角色、规范、组织等),进一步丰富 Petri 网的语义内涵,将有利于使这两方面技术相融合,从而有利于组织个体协同交互的建模与分析,更好地服务于该类复杂系统的开发。

基于上述认识,本文进行了相关研究。第 2 节介绍有关 Petri 网的一些基础知识,并提出扩展的道义 Petri 网(DPN, Deontic Petri Net);第 3 节给出使用 DPN 建模与分析服务代理协同交互的基本方法;第 4 节以电子机构模型为例,给出以 DPN 作为其形式化版本的基本映射方法,并以一个实例机构说明 DPN 方法的使用;最后是相关工作说明和结束语。

1 Petri 网及其扩展——DPN

Petri 网首先是由 Carl Adam Petri 于 1962 年在其博士论文中提出的,随后得到很多人的丰富和发展,目前已形成了比较系统的形式化建模、模型分析验证方法技术体系,并在辅助软件工具的支持下在不少领域得到了成功应用^[4]。在基本 Petri 网基础上,已先后提出了多种表达能力更强或更方便分析的高级 Petri 网,如有色网^[5](Colored Petri Net, CPN)、引

到稿日期:2008-12-15 返修日期:2009-03-03 本文受国家自然科学基金(60663005),广西自然科学基金(0728089)资助。

蔡国永(1971—),男,博士,教授,CCF 高级会员,主要研究方向为软件形式化开发方法、服务协同的虚拟组织系统, E-mail: ccgycai@guet.edu.cn;董荣胜(1965—),男,教授,主要研究方向为安全协议软件、计算思维方法;古天龙(1964—),男,教授,博士生导师,主要研究方向为形式化软件开发方法、Petri 网及其应用、智能计算。

用网^[6,7] (Reference Petri Net, RPN)、良构网^[8] (Well-formed Petri Net, WPN)等。

1.1 基本概念

基本 Petri 网定义为一个四元组 $PN=(P, T, F, W)$ 。其中, P 表示有穷的库所集; T 表示有穷变迁集; F 表示库所到变迁和变迁到库所的连接弧集, 即 $F \subseteq (P \times T) \cup (T \times P)$; W 表示弧标注函数(或称权函数), 即 $W: F \rightarrow \mathbb{N}$, 其中 \mathbb{N} 表示自然数和零。网库所中的元素称为托肯, 它是一个库所到 \mathbb{N} 的映射, 亦称为网的标识(Marking)。通常, 标识函数记为 $m: P \rightarrow \mathbb{N}$, 也可记为矢量的形式, 即 $m \in \mathbb{N}^{|P|}$ 。初始标识为 m_0 的网系统记为 (P, T, F, W, m_0) 。网系统中变迁 $t \in T$ 的使能条件表示为 $m(P) \geq W(P, t)$ 。当使能的变迁 $t \in T$ 引发后, 则网系统的标识按规则 $m'(P) = m(P) - W(P, t) + W(t, P)$ 更新。通常这一引发过程记为 $m \xrightarrow{t} m'$ 。

初始标识为 m_0 的网系统可达标识集记为 $RS(PN, m_0) = \{m | m \in Occ(PN, m_0)\}$, 其中 $Occ(PN, m_0)$ 为网系统的引发序列集, 其定义如下:

$$Occ(PN, m_0) = \{m_0 \xrightarrow{t_0} m_1 \xrightarrow{t_1} \dots \xrightarrow{t_k} m_k \longrightarrow \dots | m_i(P) \geq W(P, t_i), i=0, 1, \dots\}$$

扩展的高级 Petri 网通常都可以展开(unfold)为基本 Petri 网, 但由于引入了高级概念, 可以更方便建模具有更高复杂度的系统。如 CPN 引入颜色的概念, 使得使用颜色可以更精细地刻画网系统中的托肯、库所、变迁的类型。同时, 采用多集和变量表达的弧权函数, 并引入变迁的哨(布尔)函数式等, 大大加强了网模型的表达方便性。通过引入分层次细化的思想, 高级网更进一步贴近了人直观建模时自顶向下的思维习惯。

除了引入颜色概念外, 有色网通常还引入超变迁的概念, 即顶层网中的变迁可以精化为一个单独的子网, 从而变迁可以分层建模。引用网则采用了更具一般性的“网中网(net within net)”的概念, 即高层网库所中的托肯可以指向一个动态的实体, 动态的托肯之间可以通过同步通道进行交互, 从而使 RPN 更适合于建模复杂的交互系统。按照 RPN 的观点, 高层网称为系统网, 下一层的子网称托肯网或对象网。通常, 系统网只创建一个实例, 而托肯网可以创建多个实例。当 RPN 中的托肯不再是动态的实体, 而是一般的数据载体时, RPN 就退化为一般的 CPN。同步通道用于同步两个或更多个变迁, 使它们能同时以原子方式引发。变迁同步时, 必须就通道名和参数名集达成一致。在 RPN 的辅助工具 RENEW^[9] 中, 通过采用一定格式的变迁动作表达式来区分。

1.2 道义 Petri 网

在面向多主体系统开发框架中, 主体(通常也称代理)指具有一定反应性、决策性的实体(软件进程或物理设备)或用户。角色用于规定主体在多主体协同组织中的职责, 主体只能通过扮演角色而参与组织, 从而方便对异质主体的管治。规范分为个体层和组织层。组织层规范指组织整体期望达到的目标, 如安全、公平、可信等; 个体层指具体的组织角色在组织中的权限、义务和禁忌。参与组织的自治主体如果遵照组织的规范行动, 则可以得到相应的收益, 否则会受到相应的处罚。基于角色的分离机制是多主体协同交互管治的主要方法。在高级 Petri 网中引入主体、角色和道义规范的概念, 将能更好和更方便地形式化建模和分析复杂开放环境下多自治

主体交互情境, 为协同虚拟组织的开发提供更大的支持。

定义 1 道义 Petri 网是一个六元组 $DPN=(Places, Transitions, Flows, Colors, Types, Mappings)$, 其中

- $Places$: 库所集。
- $Transitions$: 变迁集。
- $Flows$: 弧集, 分两大类: 离开库所的弧集 $F^- \subseteq P \times T$ 和进入库所的弧集 $F^+ \subseteq T \times P$, 即 $Flows = F^- \cup F^+$ 。
- $Colors$: 基本颜色类, 分两大类, 即 $Colors = RC \cup BC$, 其中 RC 称角色集, BC 称基本颜色集; RC 与 BC 不相交, 即 $RC \cap BC = \phi$ 。
- $Types$: 弧类型集, 包括普通弧类型 IT 和道义弧类型 DT , 即 $Types = IT \cup DT$, $IT \cap DT = \phi$, 其中 $DT = \{Obl, Prm, Prh\}$, Obl, Prm, Prh 分别表示道义算子: 义务、允许和禁止。普通弧与通常 Petri 网类似, 用于表示理想受控的托肯移动路由, 而道义弧用于表示自治托肯的可能路由。
- $Mappings$: 指映射函数集, $Mappings = \{m_1, m_2, m_3\}$; m_1 指库所集到颜色集的映射 $m_1: Places \rightarrow Colors^*$, $Colors^*$ 指基本颜色类的笛卡儿集; m_2 指弧集到类型集的映射 $m_2: Flows \rightarrow Types$ 。对于 m_2 , 要求满足一定的条件, 即只有离开角色库所的弧才能映射到道义类型, 形式表示为 $m_2(p, t) \in DT$, iff $f \cdot m_1(p) \in RC$ 。 m_3 指弧或变迁集到标注的表达式集上的映射 $m_3: F \cup T \rightarrow E$, 其中 E 为表达式集合。

表达式通常可采用程序语言合法语句块的形式, 如支持 CPN 的 CPN Tools^[10] 工具中采用 ML(meta language) 语言、支持引用网的工具 RENEW 中采用 Java 语言等, 当然也可以是在有关变量集、常量集、基本颜色集基础上通过关系算子和逻辑算子构造的简单表达式语言。通常变迁上的表达式可分为 3 部分: 布尔条件表达式(称“哨”)、动作表达式和事件表达式。这里对 DPN 网的表达式不做专门的规定, 可根据仿真环境的具体需要设定。

定义 2 道义网系统(或称网实例)定义为三元组 $(id, DPN, Marking)$ 。其中 id 表示网实例的身份索引标识符; DPN 表示道义网, 作为生成实例的模板; $Marking$ 表示生成实例时设置的道义网的标识函数或托肯分布函数。

$Marking$ 可具体定义为 $Marking: P \rightarrow Bag(ID \cup m_1(P))$ 。其中 ID 表示网环境中允许的网实例索引标识集, 即 $ID = \{id_1, id_2, \dots, id_n\}$; $Bag(A)$ 表示集合 A 上的包(或多集), 有限集合 $A = \{a_1, a_2, \dots, a_m\}$ 上的一个包元素定义为一映射 $b: A \rightarrow \mathbb{N}$, 通常也记为 $\sum_{a_i \in A} b(a_i) a_i$ 。包的具体运算规则可参考文献[4]。

DPN 系统的变迁使能规则、引发序列、可达标识集与基本有色网的定义相同。然而, 由于道义弧的引入, 对于道义弧相关的使能变迁能否实际引发, 取决于托肯网实例的自主决策, 即组织的系统网和托肯网的决策同步问题。

定义 3 网组织定义为五元组 (SN, ANs, ID, σ, M) , 其中:

- SN 为一个 DPN, 通常称为网组织的系统网或称组织网;
- ANs 是有穷 DPN 集, 称托肯网实例集, $ANs = \{(id, DPN_k^s, Marking_k) | id \in ID, k \in \mathbb{N}\}$, DPN_k^s 指第 k 个 DPN 网的实例, 索引标识为 id ;
- ID 表示网组织允许的托肯网身份索引标识符的有穷

集;

• σ 表示网实例间的同步交互关系, $\sigma \subseteq Transition_i \times Transition_j, i, j \in ID, i \neq j$;

• M 表示系统网的初始标识设定函数, 要求 $m_1(p) \in RC^*$ 时, $M(p) \in ID$ 。

网组织中每一个变迁的使能规则与单个网实例的使能规则相同, 但引发规则需要满足各网实例间的同步交互关系。即对于使能的变迁 $t \in Transition$, 如 $\sigma(t) = \phi$, 则使能变迁可以独立引发; 当 $\sigma(t) = \{t_i, t_{i+1}, \dots, t_{i+k}\}$ 时, 令 $\sigma^+(t)$ 表示 t 的交互关系传递正闭包, 则只有 $\forall t_i \in \sigma^+(t)$ 和 t 本身都使能时, 变迁集 $\sigma^+(t)$ 才能同时引发。具体的引发更新规则与一般高级引用网的引发更新规则相同。

通常在研究网组织时, 组织网给托肯网提供服务。如果不考虑多级委派, 则只需要采用两级结构就可以满足通常组织中的协同交互建模分析需要。非形式的, 可用等式表示: 网组织 = {组织网}¹ + {托肯网}^{1..m}, 表示网组织由一个组织网加一到多个托肯网复合而成。

2 基于 DPN 网建模的基本策略和分析方法

通常 Petri 网模型构建时, 可以采用的基本策略有: 自底向上、自顶向下、面向对象(主体)和面向功能分解。考虑到 DPN 主要是针对服务协同组织, 并且假定组织个体元素具有一定的交互自治性, 因此在 DPN 模型的构建中建议采用自顶向下和面向主体的复合策略方式。这里所谓面向主体主要指建模时首先找出组织中协同的基本主体对象, 然后根据组织协同任务的要求, 找出主体对象的交互事件或活动, 进而建立起协同任务的基本交互结构模型。自顶向下是指建模时先建立顶层的协同交互结构, 表达顶层的实体流、数据流和控制流; 然后对上一层模型中的主体对象和交互活动进行细化, 分别建立协同组织的主体对象网和交互活动网。通过逐渐细化, 直至达到刻画足够精细的、满足需要的协同组织交互模型。由于 DPN 主要引入了组织角色和规范的概念, 因此下面提出面向角色的 DPN 模型构建方法。

2.1 面向角色的 DPN 模型构建过程

面向角色的 DPN 模型构建过程建立在组织主体分析、角色分析、规范分析和交互分析的基础上, 具体而言主要包括下面一些步骤。

Step 1 根据服务协同组织的基本需求描述, 抽取协同组织中的基本主体元素及数量, 得到基本主体元素集和数量约束条件集;

Step 2 分析基本主体元素在组织中可能呈现出的角色, 建立角色之间的基本转化关系和职权, 得到基本的角色集和抽象规范说明;

Step 3 从协同组织任务工作流程的角度, 进一步明确和细化角色的规范, 得到面向事件或活动的角色规范说明集;

Step 4 分析协同组织的数据流和数据类型, 建立基本的颜色集、条件表达式集。结合角色转化关系和协同组织业务流程, 建立顶层的 DPN 图;

Step 5 对顶层 DPN 图进行细化, 包括数据类型(颜色集)、角色集、同步变迁、内部变迁等, 根据模拟分析的需要建立托肯对象网;

Step 6 根据协同组织分析需要, 建立模型分析的基本

特性, 如有界、安全和活性、性能等, 得到组织网系统的动态特性规范集;

Step 7 通过仿真、自动或人工计算分析模型基本动态特性需求的可满足性;

Step 8 审核和重复上述的基本步骤, 直至得到满足需求的正确的 DPN 模型。

2.2 DPN 模型的基本分析方法

与基本 Petri 网的分析方法类似, DPN 模型的分析包括 3 种基本技术: 静态结构特性分析、动态行为可达性验证分析、动态仿真分析。通常对于结构比较简单、有规律的网(通常建模时要求模型满足一定的语法约束条件), 可以进行静态结构分析, 找出基本的结构不变量, 如环结构、对称结构等, 从而可得到一些不变量特性。如果网的状态空间是有穷的或可数无穷的, 理论上可以采用可达性分析技术对网行为的活性、安全性等进行模型检查验证。如果网的状态空间无穷, 或虽有穷但无法进行模型检查, 则可以进行仿真分析, 选择一些典型的初始设置, 测试网的行为符合需求, 如功能正确、库所有界、无冲突或死锁等。通常进行结构分析时, 可以假定托肯实例是有限的常量(通常假定库所容量为 1), 然后对网的安全、有界、可达性、死锁和活性等进行分析。在进行动态仿真分析时, 需要设定托肯实例的各种情况, 如托肯实例数量分布、性质、实例优先调度策略等, 从而重点分析协同业务实施的统计性能。

DPN 可以看作是一种特殊的引用网(角色是一种刻画更精细的托肯网), 因此 DPN 的一种便捷的分析验证途径是通过把 DPN 转换到适当的高级网工具进行的。如果不考虑弧的道义类型, 可以转换为引用网工具 RENEW 进行分析; 当要考虑弧的道义类型时, 可以采用概率(或随机选择)的方式, 来引发义务、允许和禁止类型的变迁, 从而采用类似于概率网的分析方法对 DPN 模型进行统计性能分析。具体的转换方法将另文讨论。

3 DPN 在电子机构形式化分析中的应用

为了说明 DPN 网的建模能力和实用性, 下面以一个具有相当复杂度的电子机构模型到 DPN 的转换来说明。

3.1 电子机构简介

在开放环境下的协同系统开发研究中, Marc Esteva 等提出了电子机构(EI, Electronic Institution)模型^[11]。电子机构被认为是一种虚拟的管治环境, 它对参与该机构的相关实体的社交行为进行管治。形象地说, 电子机构好比是一个舞台, 参与机构的自治主体类似于演员, 每个演员扮演机构中的一个或多个角色。主体之间的基本交互单元通过机构场景来表达, 场景之间可以组合成为一个场景网络, 称为机构的实施结构。通过机构, 主体可加入不同的机构场景, 并扮演不同场景中的角色。主体的交互行为通过使用言语动作(speech act)完成, 机构对话框架定义了机构允许使用的交互言语词汇和本体。主体在机构中的言语行为会直接影响其在机构的权力和职责。机构对参与主体的管治通过机构规范来具体规定。按文献[11], 电子机构定义如下。

定义 4 电子机构定义为一个五元组 $EI = \langle PS, IR, \pm, ssd, N_{PS} \rangle$, 其中 PS 表示机构实施结构, IR 表示机构角色集, \pm 表示机构角色之间的偏序关系, ssd 表示机构角色之间的静

态职责分离约束集, N_{PS} 表示机构规范规则集。

定义 5 实施结构定义为十一元组, 即 $PS = \langle S, T, s_0, s_n, E; f_L, f_T; f_E^I, C, ML, \mu \rangle$, 其中 S 表示机构场景集, T 表示机构场景迁移集, $s_0 \subseteq S$ 表示初始场景, $s_n \subseteq S$ 表示终止场景, $E = E^I \cup E^O$ 表示弧标识集, $E^I \subseteq S \times T$ 表示从机构场景到机构变迁的弧集, $E^O \subseteq T \times S$ 表示从机构变迁到机构场景的弧集; $f_L: E \rightarrow DNF_2^Y \times R$ 表示弧的标记函数, 每一弧映射到一个由角色标识与主体变量的序偶形成的合式公式; $f_T: T \rightarrow \mathcal{S}$ 为机构变迁的类型标记函数, 变迁类型有与 (and)、或 (or) 两大类; $f_E^I: E^I \rightarrow \mathcal{S}$ 为进入机构场景的弧的类型标记函数, 弧的类型有 4 种: one, some, all, new, 分别表示进入一个 (一些、所有) 场景实例, 还是进入一个新创建的场景实例; $C: E \rightarrow ML$ 表示弧的约束条件标记函数, ML 为表达约束条件式的元语言; $\mu: S \rightarrow \{0, 1\}$ 表示场景实例类型的标记函数, 表示场景在运行时是否可以同时有多个实例。

定义 6 场景定义为一个十一元组, 即: $S = \langle R, DF_S, W, w_0, W_f, (WA_r)_{r \in R}, (WE_r)_{r \in R}, \theta, \lambda, \min, \max \rangle$, R 表示机构场景的角色集, DF_S 表示机构场景的对话框架 (具体见定义 7), W 表示场景状态集, $w_0 \in W$ 表示场景初始状态, $W_f \subseteq W$ 表示场景终止状态集; $(WA_r)_{r \in R} \subseteq W$ 表示所有角色的可达状态集簇, WA_r 表示角色 $r \in R$ 的可达状态集; $(WE_r)_{r \in R} \subseteq W$ 表示角色的退出状态集簇, WE_r 表示角色 r 的退出状态集; $\theta \subseteq W \times W$ 为状态间的有向边集; $\lambda: \theta \rightarrow L$ 表示状态边的标记函数, 其中 L 为言语行动、超时事件及约束条件式的集; $\min, \max: R \rightarrow \mathbb{N}$ 表示可以扮演角色的主体数。函数 $\min(r)$ 和 $\max(r)$ 分别返回能扮演角色 r 的主体的最小个数、最大个数。

定义 7 对话框架表示为一个六元组, 即 $DF = \langle O, L, I, R_I, R_E, R_S \rangle$, 其中 O 表示电子机构的领域本体论; L 表示主体间言语动作的信息内容表示语言; I 表示言语行为词的集合, 其元素的基本形式是 $i(a, r, a', r', m, t)$, 含义是在时刻 t , 扮演角色 r 的主体 a 向扮演角色 r' 的主体 a' 发送内容为 m 的消息的言语动作 i ; R_I 表示机构的内部角色集, R_E 表示机构的外部角色集, R_S 表示角色间的关联关系。

机构的对话框架、场景、实施结构定义了主体在电子机构中的可能交互空间。针对该交互空间的基本元素, 可以定义一些谓词或函数, 表达相应的机构判定或计算操作, 如定义谓词 $uttered(s, w, i)$ 表示在机构场景 s 的状态 w 处, 发生了言语行为 i 。机构的规范断言可以定义在这些基本的谓词之上, 通常使用如下的规则形式。

$$\left(\bigwedge_{j=1}^n uttered(s_j, [w_{k_j}], i_{j_1}) \wedge \bigwedge_{k=0}^m e_k \right) \rightarrow \left(\bigwedge_{j=1}^{n'} uttered(s'_j, [w'_{k'_j}], i'_{j_1}) \wedge \bigwedge_{k=0}^{m'} e'_k \right)$$

其直观的意思是: 如果言语行为 i_{1_1}, \dots, i_{1_n} 在相应的场景状态发生, 且表达式 e_1, \dots, e_m 成立, 则使表达式 $e'_{1_1}, \dots, e'_{m'}$ 成立的言语行为 $i'_{1_1}, \dots, i'_{1_n}$ 必须在相应的场景状态处发生。

尽管上述定义给电子机构描绘了一幅清晰的概念图像, 可以作为分布协同交互系统规约的元模型使用, 然而当用于建模及分析协同交互系统时, 也存在明显不足。首先, 对这种基于状态定义的形式模型, 没有给出明确的语法和语义, 因此模型缺乏自动分析或模拟执行的能力; 其次, 在这种形式模型中, 机构场景角色间的并发性并没有得到很好的处理, 不具有组合分析能力。而可视化的、有严格语法和语义定义的、支持

并发和通信能力的高级 Petri 网刚好可以弥补上述定义的不足。为此, 下面给出上述定义的 EI 到 DPN 的映射方式, 从而使得能够以 DPN 方式建立 EI 的形式化模型, 便于 EI 模型的形式建模及 EI 模型分析。

3.2 EI 模型到 DPN 的映射规则

从上述 EI 的定义可以看出, EI 是一种自顶向下的分层结构, 因此与自顶向下的 DPN 模型兼容。对于 EI 的实施结构 PS, 可以建模为网组织的系统 DPN, EI 中的场景结构可以表示为 DPN 形式的托肯网。EI 的规范相当于组织网的活性和安全性。EI 中的角色集和角色冲突关系可以分离为 DPN 的角色集 (一类颜色集) 及颜色之间的约束关系。实施结构中的迁移关系对应于 DPN 网中各类型变迁结构, 因此可以建立起如表 1 所列的 EI 到 DPN 组织网的映射规则。

表 1 EI 到 DPN 组织网的映射规则

EI 模型元素	DPN 表示的网组织模型元素
实施结构 PS	网组织的系统网
PS 中的迁移	系统网中的变迁
PS 中的弧	系统网中的弧
弧表达式	弧表达式和变迁上的哨
输出弧类型 (one, all, some, new)	变迁上的动作表达式 (相应设计四种)
迁移类型	变迁类型 (相应设计 and, or 类型)
场景	托肯网
场景实例	托肯网实例
场景实例个数	库所托肯容量限制函数
场景状态	托肯网中的库所标识
场景中状态关联边	托肯网中的变迁
角色集	DPN 的角色集
会话言语	同步通道 (变迁动作表达式)
对话框架	领域标记表达式语言
规范	网组织的安全性、活性
(其它待补充的规范)	网的其它特性

分析方法: (待补充完善的分析方法) 分析方法: 基于 DPN 的可达、仿真分析等

从以上 EI 到 DPN 表示的网组织的映射中可以看出, 通过 DPN 不仅使得 EI 模型建立在坚实的形式化理论基础之上, 而且可以借助 Petri 网研究中积累起来的许多成功的分析方法, 从而对电子机构的分析提供科学的方法。另一方面, 对 EI 模型没有考虑到的方面或特性, 可以从 Petri 网的相关研究中得到启示, 可以进一步完善 EI 模型。

3.3 电子机构示例分析

设有一个承办学术会议的电子机构, 其基本目标是协助会议组织者安全可靠、公平公正地完成学术会议的承办工作。按照 2.1 节介绍的建模策略和 3.1 节的 EI 模型, 假设该示例会议承办机构涉及的基本角色分类为: 申请者 (applicant)、组织者 (organizer)、作者 (author)、评阅人 (reviewer), 机构涉及的基本场景有: 申请者注册 (Register)、注册作者投递稿件 (submit) 和注册评阅人对投递稿件进行评审 (review)。假设承办机构对这 3 个场景按顺序的方式组织, 则得到基本的实施结构。按照引用网和 DPN 的建模符号, 示例机构的实施结构 DPN 图可用图 1 表示。图中 3 个大的椭圆形表示的库所分别对应 3 个场景, 白色圆形库所代表的分别是机构角色的库所、变迁表示相应的动作或触发事件。设计好具体的动作和触发事件表达式, 并设置申请者或组织者的初始托肯通过网的仿真或分析工具, 则可以对实施结构的 DPN 模型进行分析确认 (由于表达式的表示与具体的分析或仿真工具密切相关, 图中没有具体给出, 可考虑用 RENEW 工具中的类 java 语言形式)。

场景库所的托肯,对应场景网的实例。对于图1中的场景库所托肯,可进一步细化,即用托肯网表示。图2和图3分别给出了示例电子机构中的申请者注册场景(register)和论文投递场景(submit)对应的托肯网。这里每一个场景描述的都是一个角色扮演者之间的交互协议。作为模拟分析,这里用两个库所来简单模拟理想的输入通道和输出通道。另外假设角色扮演者是遵守协议中角色的职责的,即相应事件或动作总是会发生的。如果要考察扮演者的自治特性,则可以补充一些库所,以一定概率的方式来模拟事件的发生,从而对机构的整体特性做综合分析。由于篇幅关系,本文不对此做详细讨论,可参见文献[13]。

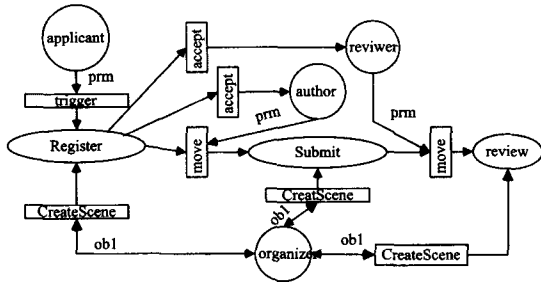


图1 示例机构的实施结构(顶层 DPN 图)

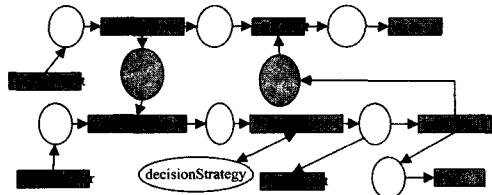


图2 示例机构的注册场景 DPN 图

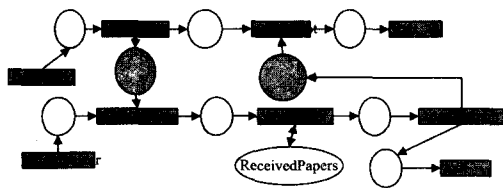


图3 示例机构的投稿场景 DPN 图

结束语 Petri 网由于其并发和通信描述能力,有严格形式化语义的特点,而在各个领域的建模与分析得到广泛研究和应用(如柔性制造系统^[4]、通信协议系统^[12]、工作流系统^[13]和多主体系统^[14]等)。然而,由于 Petri 网定义中的元素都是脱离了上下文的抽象概念(如库所、托肯都是非常抽象的概念),建模系统的领域复杂性,使得在用 Petri 网建立复杂领域的系统模型时面临极大的困难。面向对象网、引用网等高级网的提出,使得其形式化建模过程更贴近人的逻辑认知过程规律,从而建模更直观便捷,对克服此复杂性起了一定的积极作用。然而由于动态环境中主体对象的自治性,导致了领域系统更复杂的交互过程,因此 Petri 网需要进一步扩展,以符合新的领域认知过程规律。在文献[15]提出的 OgPN 网中,通过把变迁分类为理想(ideal)变迁、违约变迁(violation)和修复变迁(repair),同时给每一个变迁引入成本(cost)函数,从而可以描述一些组织道义方面的状态。与 OgPN 的做法不同,本文提出的 DPN 首先是引入角色的概念来描述组织结构,然后直接使用道义逻辑中的算子(义务、允许、禁止)来标识角色的职责。同时 DPN 引入了双层的网结构,上层的组织网负责为下层的角色托肯网协调提供服务。托肯网之间、组织网与托肯网之间采用同步通道的方式进行通信。由此使

得 DPN 能更好地保持真实组织的结构,更易于组织建模。

总之,与已有的一些 Petri 网扩展不同,本文基于动态环境下对复杂组织社交规范的逻辑认知规律,把面向多主体协同的社交规范的认知逻辑和引用网的形式化方法相融合,提出了一种道义 Petri 网(DPN),并讨论了其在动态环境下的协同组织建模及分析上的应用。通过与电子机构模型结合的实例说明,这两种方法可以很好互补,从而能更好地服务于动态协同组织系统的开发。然而,由于动态开放环境下复杂组织社交规范认知逻辑规律尚不完全,本文的 DPN 也还有许多地方需要进一步完善,如对于不同层次的 DPN,如何说明抽象的道义规范与细化后的网或局部子网的行为规范是一致的,尚需要更深入的研究。

参考文献

- [1] Wooldridge M, Jennings N R, Kinny D. The Gaia Methodology for Agent-oriented Analysis and Design[J]. Autonomous Agents and Multi-Agent Systems, 2000, 3: 285-312
- [2] Ferber J, Gutknecht O, Michel F. From Agents to Organizations: an Organizational View of Multi-Agent Systems[C] // Giorgini P, Müller J, Odell J. eds. Agent-Oriented Software Engineering (AOSE) IV, Melbourne, July 2003, LNCS 2935. 2004: 214-230
- [3] Dignum V, Vazquez - Salceda J, Dignum F. OMNI: Introducing Social Structure, Norms and Ontologies into Agent Organizations[C] // Bordini R H, et al. Eds. PROMAS, 2004, LNAI 3346. 2005: 181-198
- [4] Girault C, Valk R. 系统工程 Petri 网—建模、验证与应用指南[M]. 王生原,余鹏,霍金键,译. 北京:电子工业出版社,2005
- [5] Jensen K. Coloured Petri Nets: Volume 1; Basic Concepts, Analysis Methods and Practical Use[M]. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin Heidelberg New York, 1992
- [6] Valk R. Petri Nets as Token Objects; An Introduction to Elementary Object Nets[C] // Desel J. eds. 19th International Conference on Application and Theory of Petri Nets, LNCS 1420. Berlin; Springer-Verlag, 1998
- [7] Valk R. Object Petri net; Using the nets - within - nets Paradigm [J]. Lecture Notes in Computer Science, 1999(3089): 819-848
- [8] Chiola G, Dutheillet C, Franceschinis G, et al. Stochastic well-formed colored nets and symmetric modeling applications[J]. IEEE Transactions on Computers, 1993, 42(11): 1343-1360
- [9] <http://www.renew.de/>
- [10] Kristensen L M, Christensen S, Kurt J. The practitioner's guide to colored Petri nets[J]. International Journal on Software Tools for Technology Transfer; Special section on coloured Petri nets, 1998, 2(2): 98-132
- [11] Esteva M. Electronic institutions; from specification to development[D]. Monography, 2003, 19
- [12] 古天龙,蔡国永. 网络通信协议的形式化分析与设计[M]. 北京:电子工业出版社,2003
- [13] Bigham J, Jin Xuan, Gamez D, et al. Hybrid Workflow and Bayesian Networks to Correlate Information in the Protection of Large Scale Critical Infrastructures[J]. Electronic Notes in Theoretical Computer Science, 2005(11): 87-99
- [14] Xu Haiping, Shatz S M. A Framework for Model-based Design of Agent-oriented Software[J]. IEEE Transactions on Software Engineering, 2003, 29(1): 15-30
- [15] Combettes S, Hanachi C, Sibertin-Blanc C. Organizational petri Nets for protocol design and enactment[C] // Proceeding of AAMAS'06. Hadodate, Japan, May 2006: 1384-1386