

时态工作流过程模型及其合理性验证^{*}

余 阳¹⁺, 汤 庸¹⁺, 潘茂林², 郑婷婷¹, 麦健彬¹

¹(中山大学 信息科学与技术学院, 广东 广州 510275)

²(中山大学 软件学院, 广东 广州 510275)

Temporal Workflow Process Model and Its Soundness Verification

YU Yang¹⁺, TANG Yong¹⁺, PAN Mao-Lin², ZHENG Ting-Ting¹, MAI Jian-Bin¹

¹(School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China)

²(School of Software, Sun Yat-Sen University, Guangzhou 510275, China)

+ Corresponding author: E-mail: yuy@mail.sysu.edu.cn, issty@mail.sysu.edu.cn

Yu Y, Tang Y, Pan ML, Zheng TT, Mai JB. Temporal workflow process model and its soundness verification. *Journal of Software*, 2010,21(6):1233–1253. <http://www.jos.org.cn/1000-9825/3608.htm>

Abstract: To improve workflow products, handling of the validity of information, temporal workflow is presented as a new concept by introducing time into workflow concept model as a dimension, which leads to time attributes being assigned to all workflow basic concepts and relationships between them. Based on the research on the expression and calculation of temporal information and the process meta-model of temporal workflow, through mapping the concepts in the meta-model into the elements of Petri Net, a temporal workflow process model, TPWF-net, is presented, which can describe the process, resource, case and their time attributes in one model synthetically. Then, some theorems are proved, which include structural equipollence between TPWF-net and WF-net, and the soundness of free-choice synchronal TPWF-net. It is also proved that well-structured TPWF-net can be decided in polynomial time. Finally, a structured modeling method of TPWF-net and a structure-simplified method of model verification are presented. Temporal workflow makes it more comprehensive to describe and analyze the time-related problems in workflow area. A temporal workflow engine prototype based on TPWF-net has been implemented and applied in some projects supported by local government.

Key words: temporal workflow; process model; soundness; Petri net

摘 要: 为了提高工作流产品对信息时效性的处理能力,通过将时间维引入工作流概念空间,对构成工作流的基本概念及概念间的关系进行了全面的时间属性扩展,提出了时态工作流的概念.在前期对时态信息表示及演算、时

* Supported by the National Natural Science Foundation of China under Grant Nos.60573160, 60673135, 60736020, 60873162 (国家自然科学基金); the Special Foundation for Developing Modern Information Service Trades in Guangdong Province of China under Grant No.GDIID2008IS030 (广东省现代信息服务业发展专项资金); the Guangdong-Hongkong Key Domain Breakthrough Project of China under Grant No.2007Z1-D6111 (广州市粤港关键领域重点突破项目); the Research Foundation of Science and Technology Plan Project in Zhuhai City of China under Grant No.200922019 (珠海市科技计划); the Research Foundation of Science and Technology Plan Project in Yuexiu District of Guangzhou City of China under Grant No.2008-GX-017 (广州市越秀区科技计划)

Received 2008-11-21; Accepted 2009-03-05

态工作流元模型研究的基础上,提出了一种能够综合描述过程、资源、案例、时间四维信息的时态工作流过程模型 TPWF-net,并证明了 TPWF-net 与 WF-net 的结构等价性、自由选择同步 TPWF-net 和良构 TPWF-net 的合理性可以在多项式时间内判定等结论.在此理论的指导下,提出了结构化建模的思想,并阐述了基于结构化简的 TPWF-net 模型合理性验证的方法.时态工作流能够更加全面地描述和分析工作流领域的时间相关问题,开发的一个时态工作流引擎原型已在一些应用研究性项目中得到了验证.

关键词: 时态工作流;过程模型;合理性;Petri 网

中图法分类号: TP311 文献标识码: A

1993 年, workflow 管理联盟(WfMC)的成立和它发布的工作流参考模型^[1]标志着 workflow 技术开始进入相对成熟的阶段,涌现出了大量 workflow 系统产品.随着企业竞争的加剧和政府提高效率的迫切需要,对 workflow 基础平台的时效性、灵活性、可靠性提出了更高的要求.时态工作流通过将时间维度引入 workflow 概念空间来全面扩充其基本元素的时间属性描述,在关注过程效率的同时综合处理数据、资源、工具(即 invoked application^[1])等基本元素的有效性问题.本文在引入时态工作流(temporal workflow)概念并对其元模型研究^[2,3]的基础上,进一步研究了时态工作流的过程模型及其合理性验证问题.

1 相关研究

与其他事物一样, workflow 应用也处于宇宙的四维空间中,即表达空间位置的 3 个维度和时间维.但由于网络技术的发展,空间位置因素对 workflow 应用的影响主要转化为对时间要素的影响,如文献[4,5]提出的 workflow 模型扩充了对时区(time zone)的描述,并考虑了信息传输延迟.因此,本文的研究不再考虑表达空间位置的 3 个维度,而仅考虑时间维度对 workflow 应用的影响.时间因素对 workflow 应用的影响是广泛和重要的,目前,在 workflow 的时间过程模型及分析方面的主要研究有:

(1) 工作流的时间建模和验证:Eder 建立了包含活动时间属性的赋时活动图模型(workflow timed graph)^[6],定义了活动执行延迟、截止期限、固定日期以及时间距离等约束,并将所有时间约束换算为活动执行结束的时间点.Ling 等人扩展了基于 Petri 网的工作流模型 WF-net,提出了时间工作流网 TWF-net^[7],将活动时间映射为变迁时间对,并分析了它的活性、安全性和合理性.文献[2,3,8]等在此基础上进行了进一步的研究和扩展.文献[9,10]认识到 workflow 模型仅描述控制流会造成实际应用和语义验证的困难,其提出的模型扩展了对数据和资源的描述.

(2) 工作流的性能分析:基于赋时过程模型,Zhao 等人提出了一种过程时间预测算法和任务时间的分配策略^[11];李建强、范玉顺等人结合开环 Jackson 排队网络理论和马尔可夫链详细讨论了静态、动态性能分析的方法^[12,13].

综上所述,目前与时间相关的工作流问题的研究主要集中在 workflow 的过程建模上,并基于这些过程模型分析活动或过程的时间约束和时序约束,其研究目的主要是为了提高 workflow 实例的执行性能或效率.

文献[2,3]引入了时态工作流的概念,详细分析了 workflow 元模型中所有元素、元素间关系的时态属性,定义了时态信息元素的规范表示和一组时态数据演算,并对时态工作流的主要元素进行了形式化的描述.以上相关工作作为本文进一步研究时态工作流的过程模型提供了基础.

2 时态工作流的概念及其元模型

2.1 时态工作流的概念

Aalst 将 workflow 的组成元素分为 3 个维度:资源(resource)、案例(case)和过程(process),如图 1 所示^[14].在这个三维空间里有 5 个基本概念:资源(resource)、案例(case)、任务(task)、工作项(work item)、活动(activity).资源是活动的执行者,它可以是人,也可以是程序或设备的代理,资源具有主动性;案例体现为要处理的描述业务对

象的数据或称为信息,它包括应用数据和相关数据^[1];过程是由一系列任务组成的;任务与案例结合(实例化)叫工作项;工作项与资源结合(触发)成为活动。这些概念在学术论文里的使用很不统一,在此加以明确以便作为后续描述的基础。

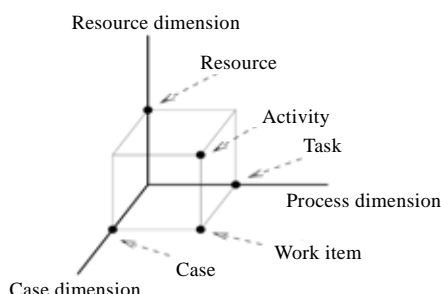


Fig.1 A three dimensional view of a workflow

图 1 工作流的三维视图

将时间维引入图 1 所示的工作流概念空间中后,组成工作流的基本元素并没有变化,它体现在工作流的所有组成元素及其关系都具有了时间属性,就如同现实世界中,时间维的存在表现在每个事物及其事务间的关系都具有时间属性。基于这种思想,提出了时态工作流(temporal workflow)的概念。

定义 1.1(时态工作流). 时态工作流就是将时间维引入工作流概念空间,从而可以全面、系统地研究工作流系统中各元素及元素间关系的时态特性及其规律。

从本文的相关研究分析可以看出,目前的研究主要涉及过程维(process dimension)的时间扩展,具体来说就是对 task 施加局部的时间约束或对 process 施加全局的时间约束,而对资源维(resource dimension)和案例维(case dimension)的时间约束却甚少涉及。更进一步地,对这些概念间关系的时间约束的研究就更加罕见。时态工作流强调时间维,就是强调时间约束引入的彻底性。值得注意的是,现实工作流应用中的时间问题往往是多个维度元素的时间属性(约束)共同作用的结果。

时态工作流或 Temporal Workflow 的命名是为了与目前所见到的时间工作流、时间约束的工作流、Time Workflow、Timed Workflow 等概念区别开来,它们都只对过程维的概念扩展时间属性;同时,时态工作流引入时态信息处理领域的成果,规范了时间系统定义、时间数据类型以及时间数据的演算规则。比较接近的研究有:文献[11]出现了“Temporal Workflow”这样的文字,文献[15]提到了时间维,但它们仍然只描述了过程维的时间属性;Combi 等人的研究与时态工作流的思想最为一致,在文献[16]中较全面地分析了 3 个维度的时间属性,并在文献[17,18]中使用了 Temporal Workflow 这个词,但并未给出明确的定义,其阐述也未涉及元素间关系的时间属性和时间信息的规范化表示。因此,对时间属性的引入仍是不够全面的,并且其研究主要在实现层面上(用时态数据库管理时间信息),没有从形式化模型及其分析等理论方面深入研究。

2.2 时态工作流元模型

工作流的元模型是用于描述组成工作流的各个元素、元素之间关系及其属性的模型。为了将时间维引入工作流概念空间,必须为所有要素及其关系赋予时间属性。因此,时态工作流元模型的研究是进行过程建模的重要基础,而其中的过程元模型是这一基础的核心。

本文仅对过程元模型的主要内容加以描述,有关元模型的完整研究见文献[2,3]。在研究分析的过程中,借鉴一些工作流应用开发的实践经验,对文献[1]定义的基础过程元模型进行了一定的扩展。这并不是工作流概念全面引入时间属性所必须的,但会为进一步研究时态工作流的其他问题打下良好的基础。一个扩展的工作流过程定义元模型如图 2 所示。为了支持结构化的过程定义,该模型引入了子过程(sub-process)概念。一个过程是由一系列子过程、任务(task)和连接符(connector)组成的。同时,子过程又继承了过程的定义,这就为过程定义提供了不同逻辑层次上基本任务粒度控制的灵活性。

3 时态工作流的过程模型(TPWF-Net)

在时态工作流过程元模型研究的基础上,本节通过将元模型相关概念映射到 Petri-Net 的元素,并将 Petri-Net 适当扩展形成时态工作流网(TPWF-net)来描述其过程模型.TPWF-net 继承了 WF-net 和 TWF-net 描述单案例控制流的思想,扩充了托肯的时间属性以表达案例数据的时间约束,而将资源及其时间属性映射为变迁的属性,从而建立了一个可同时描述过程、资源、案例和时间四维信息的时态工作流过程模型.

3.1 工作流概念到模型元素的映射

3.1.1 案例与托肯

案例(case)代表 WFMS 处理的业务对象,在 WF-net 和 TWF-net 中,案例被映射为 Petri 网中的托肯(token)^[17,19],但仅仅被作为前一个任务完成信号的载体^[20],Aalst 提到时间扩展时,为托肯扩展了一个时间戳属性,用以描述托肯何时可用^[19],主要用于处理两个任务间的时间延迟.Ling 以扩充一个时间映射函数的方式为活动增加时间延迟的属性^[7];周建涛的 3DWFN^[10]将托肯分为两类,分别代表案例数据和资源(注意,3DWFN 的资源与 Aalst 定义的资源概念不同),实际上是代表了两种到达信号.总之,托肯只被用作信号,即使扩展了一个时间属性,也是为了处理任务间的时间约束,它没有属于自身的独立属性.

TPWF-net 仍然将托肯作为业务对象在过程模型中的代表,但它不仅仅被作为任务完成信号的载体.根据时态工作流过程元模型对案例的定义^[2,3],托肯被赋予独立的时间属性:案例创建时间 CT 和案例有效期 VT,它们表达了所有案例都具有的时间语义,而相关数据代表的业务数据语义及验证不是目前过程模型考虑的问题.Aalst 认为过程定义了案例的生命周期^[7],其实质是只要过程不结束,案例数据永远有效,这是不符合现实的假定.时态工作流的观点则不同,托肯被赋予独立的有效期,反映了业务对象中数据的有效期限^[2,3].用户可以通过修改案例的有效期而不是修改过程定义中任务间的时间距离来表达对案例处理的时间约束要求.当然,过程应有根据案例有效期动态调整任务时间规划的能力.托肯作为独立的对象并被赋予时间属性可以增强模型的描述能力,同时更加符合面向对象的思想.

3.1.2 任务与变迁和库所

以往的研究将活动执行的条件映射为 Petri 网库所(place),将活动的执行映射为变迁(transition).WF-net 沿袭 Petri 网的定义,将库所中拥有托肯视为条件的满足.时态工作流不同,它必须考虑各种状态下的时间延迟,同时还必须分别考虑业务任务和管理任务的映射.

首先考虑业务任务的映射.基于 WfMC 工作流参考模型描述的任务/活动状态图,通过加入时间因素可以得到图 3 所示的时态工作流任务状态转换图.

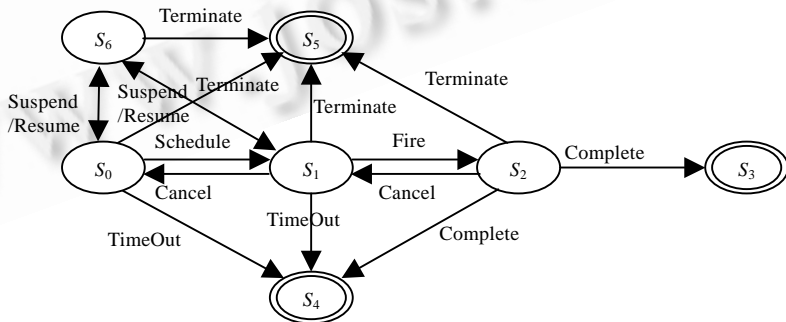


Fig.3 State transition diagram for activity

图 3 扩展的活动状态转换图

业务任务在等待状态 S_0 ,其控制权在 WFMS;而一旦业务任务被分配完成形成工作项并放入用户的工作项列表(就绪状态 S_1),则控制权已转移给了执行任务的资源.TPWF-net 以任务分配完成为分界线,将 S_0 及相关状态

下的业务任务映射为库所,将 S_1, S_2 及相关状态下的业务工作项和活动映射为普通变迁。

接着考虑连结符的映射.袁崇义教授在文献[20]中将库所定义为同步器,同步器承担了与本文路由变迁类似的职责.这样,每个库所中可能出现多个托肯,弧的权函数也必须定义,从而基于 P/T 网重新建立了一个不同于 WF-net 的过程模型.而为了继承 WF-net 的研究成果和保证网的安全性(库所中最多只能有一个托肯),本文仍然将代表管理任务的连结符视为一类具有特定职责的任务,将 S_0 状态的连结符映射为库所,将 S_1, S_2 状态的连结符映射为一类特殊的变迁——路由变迁。

概念映射后,在元模型分析时得到的任务时间属性^[20]也分别赋予映射产生的对象:库所和变迁.其中,任务分配延迟时间 SD 赋予库所,触发延迟时间 FD 和执行延迟时间 ED 赋予变迁。

可以仿照同步器来定义一个通用的路由变迁 $t(M, N, (m, n))$,其中 M, N, m, n 都是正整数,并且 $|t| = M, |t^*| = N, m \leq M, n \leq N$,其含义见下文 TPWF-net 的动态语义定义. $t(M, N, (m, n))$ 的符号表达如图 4 所示,中括号内包含的是时间属性.顺序,AND-split,AND-join,OR-split,OR-join 等路由变迁都是通用路由变迁的特例。

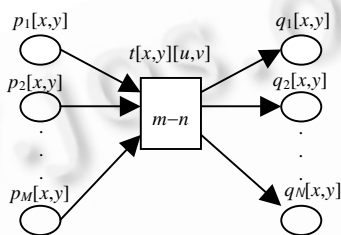


Fig.4 Symbol of all-purpose routing transition

图 4 通用路由变迁符号

3.1.3 任务间的依赖关系与弧

与 WF-net 和 TWF-net 一样,TPWF-net 将任务间的依赖关系映射为 Petri 网中的弧,弧的权值恒为 1.值得注意的是:1) 弧仅仅非定量地表示了相邻任务的时序关系,可以借助库所的时间属性进一步描述其定量关系;2) 在集中部署的 WFMS 中,任务间的数据传递延迟可以忽略不计,而在分布式 WFMS 中,数据的传递延迟往往是必须要考虑的.本文暂不涉及分布式 WFMS 的问题。

3.1.4 关于资源的映射

资源是任务的执行者,是任务/活动的组成要素,在实际的 WFMS 中,往往与任务/活动绑定在一起,并且不具有随控制流流动的特征.因此,TPWF-net 不将它映射为 Petri 网的独立元素(实际上也没有了),而将资源及其属性(包括时间属性)作为任务/活动的属性来看待,从而映射为变迁的属性。

资源维有许多概念,如组织、角色、部门、资源等,这些概念间有多种关系,如隶属、委任、委托等,时态 workflow 将这些概念和关系都赋予了时间属性^[3].一个组织有创建时间和有效时间,如:一个公司在某个日期注册成立,营业执照的有效期限是 10 年;一个用户也有入职时间和有效时间(劳动合同期);一个用户担任某个角色有开始时间和有效时间;一个用户从第 2 天起出差 3 天,将自己担任的某个角色职责委托给另一个用户,这个委托关系也有创建时间和有效时间(第 2 天起 3 天),等等.这些信息在工作流执行过程中可用于检查资源的有效性。

3.2 过程模型的形式化描述

TWF-net 和其他扩充时间属性的过程模型都是采用扩充若干个时间映射函数的方式为相应元素增加时间属性.由于 TPWF-net 要扩充的时间属性比较多,再采用这种标记方法会使模型过于复杂,且不易理解.为此引入面向对象的思想,将 Petri 网中的基本元素看作是带有若干属性的对象,并且通过对象可以单独存取某个属性.这种方法与引入时间映射函数的作法是等价的,只是标记方法不同,目的是使模型更加简洁、易懂.基于第 3.1 节在时态 workflow 概念与 Petri 网概念间建立的映射关系可以得到过程模型的形式化表示,见定义 3.1。

定义 3.1(时态 workflow 网, TPWF-Net). TPWF-net 是一个三元组 $TPN = \langle P, T, F \rangle$, 其中:

- (1) P 是库所的有限集合, T 是变迁的有限集合, 并满足: $P \cap T = \emptyset$ 且 $P \cup T \neq \emptyset$; $\exists i, o \in P, i^* = \emptyset, o^* = \emptyset$, 并且每一个节点 $x \in P \cup T$ 都位于从 i 到 o 的一条路径上.
- (2) $F \subseteq (P \times T) \cup (T \times P)$ 是弧集合, 并且 $dom(F) \cup cod(F) = P \cup T$.
- (3) $T = T_p \cup T_r$, 并且 $T_p \cap T_r = \emptyset$. 其中, T_p 是普通变迁的集合, T_r 是路由变迁的集合.
- (4) 对于 $\forall t \in T, t = \langle FD, ED, R, AP, ECA, guard(), s \rangle$, 其中: FD, ED 分别是点火延迟时间和执行延迟时间; R, AP 分别是执行该任务的角色和被调用的应用程序; ECA 是一个 ECA 规则集合; $guard()$ 表示执行该任务必须满足的控制条件函数; s 是任务/活动的状态. t 的执行具有原子性, 不可中断.
- (5) $\forall t \in T_p: |i^*| = 1 \wedge |t^*| = 1$, ECA 规则集包含了业务任务在 S_1, S_2 及相关状态的转移规则.
- (6) $\forall t \in T_r$: ECA 规则集包含了管理任务在 S_1, S_2 及相关状态的转移规则以及任务间的转移规则.
- (7) 对于 $\forall p \in P, p = \langle SD, ECA, TK, s \rangle$, 其中: SD 是分配延迟时间; ECA 是一个 ECA 规则集合, 描述了任务 S_0 及相关状态的转移规则; TK 是托肯的集合; s 是任务/活动的状态.
- (8) 对于 $\forall tk \in TK, tk = \langle CT, VT, RT, ST, FT, RdataList, CdataList, s \rangle$, 其中: CT 是托肯被创建的时间; VT 是托肯的生存周期; RT, ST, FT 分别是托肯到达的时间、调度完成时间和点火时间; $RdataList, CdataList$ 分别是相关数据和控制数据的集合; s 是最近一个任务结束时的状态.

注意:

与 WF-net 一样, TPWF-net 定义了单个案例的动态行为;

有关角色 R 、应用程序 AP 、ECA 规则等基础概念的形式化描述(含时间属性)参见文献[2,3,21]中元模型的形式化描述部分.

另外, 对于库所、变迁、托肯, 采用类似对象的形式存取它们的各个分量属性. 如: 对于 $\forall p \in P, p.RT, p.SD, p.ECA$ 分别表示库所 p 的托肯到达的时间、托肯滞留的时间区间和 ECA 规则集合.

从 TPWF-net 的定义, 可以直接得到它的下列性质:

- 1) 开始库所是唯一的. 即对于 $\forall p \in P: p = \emptyset \Leftrightarrow p = i$;
- 2) 结束库所是唯一的. 即对于 $\forall p \in P: p^* = \emptyset \Leftrightarrow p = o$;
- 3) 如果 TPN 是一个 TPWF-net, 添加一个变迁 t^* 到 TPN , 用来连接结束库所 o 和开始库所 i (即 $i^* = \{o\}$ 且 $t^* = \{i\}$), 那么所得的网 \underline{TPN} 是强连通的, \underline{TPN} 称作 TPN 的扩展网;
- 4) 普通变迁和路由变迁本质上是一样的. 除了属性的值不同外, 唯一的差别是普通变迁只有 1 个输入库所和 1 个输出库所, 而路由变迁则不受限制.

可以直观地理解性质 4) 带来的好处: 普通变迁所代表的业务任务只有一个输入和一个输出, 就如同在程序设计中提倡编写单入口/单出口的模块一样, 它使业务任务定义的独立性更强, 不会因某个过程定义不同的路由结构而改变, 因此更容易复用. 同时, 当某个过程的路由结构发生改变时, 也不必更改业务任务的定义, 为过程的柔性处理带来了方便. WFMS 为业务用户提供了一种编程和运行的环境, 其基本语句是用户可理解的(即活动, 也就是定义一个角色用一个工具处理一组数据), 可以看作是业务级别的编程. 因此, 程序设计理论的很多成果是可以借鉴的.

对前集、后集、状态(标识)的定义依然沿用 Petri 网的定义^[19].

单从定义的形式上看, TPWF-net 类似于 WF-net 的定义, 只是对各元素附加了一些属性, 对普通变迁的限制并不违反 WF-net 的定义, 但路由变迁是一个重要的不同. 与 WF-net 更大的不同是后面定义动态语义.

3.3 TPWF-Net 的动态语义

定义 3.2(库所的可用, available). 对于 $\forall p \in P, p$ 在状态 M 下是可用的, 当且仅当:

- (1) $M(p) \geq 1$, 并且
- (2) $\exists tk \in p.TK, tk$ 是可调度的, 即 $During(Now, tk.VT) \wedge During(Now, abs(p.SD))$.

注意:

$ing()$ 是一个时态运算函数, $abs()$ 是相对时间到绝对时间的转换函数, Now 表示当前时间,是一种特殊的时态元素,详细定义见文献[2,3,21];

如果 $\exists tk \in p.TK: Before(tk, VT, Now) \vee Before(abs(p.SD), Now)$,则会触发超时异常,可参照图3的状态图,将 S_0 及相关状态的所有转换规则记录在 $p.ECA$ 中.这里的 $Before()$ 也是一个时态运算函数.

定义 3.3(普通变迁的使能、可点火). 普通变迁的使能(enabled)和可点火(firable).对于 $\forall t \in Tp$:

(1) t 是 M 状态下使能的,当且仅当对于 $\forall p \in \bullet t$:

1) p 是 M 状态下可用的,并且

2) $\exists tk \in p.TK, tk$ 是可调度的,并且 $(t.s=S_0) \wedge t.guard(tk)$ (因普通变迁只有一个输入库所,所以 $guard$ 只与一个可调度托肯的相关数据和控制数据相关).

(2) t 是 M 状态下可点火的(记作 $M[t>]$,当且仅当:

1) t 是 M 状态下使能的,并且

2) $(t.s=S_1) \wedge During(Now, tk, VT) \wedge During(Now, abs(t.FD)) \wedge During(Now, t.R.VT) \wedge During(Now, t.AP.VT)$.

定义 3.4(普通变迁的实施). 普通变迁的点火/实施(fire).如果普通变迁 t 可点火,即 $M[t>]$,那么 t 能够点火/实施,点火后, t 从唯一的输入库所中删除一个可调度的托肯,并在唯一的输出库所中产生一个托肯,达到一个新的状态 M' ,记作 $M[t>M']$.

一个路由变迁可以用 $t(M, N, (m, n))$ 的形式表示,其中 M, N, m, n 都是正整数,并且 $|\bullet t|=M, |t^{\bullet}|=N, m \leq M, n \leq N$.

定义 3.5(路由变迁的使能、可点火). 路由变迁的使能(enabled)和可点火(firable).对于 $\forall t(M, N, (m, n)) \in Tr$:

(1) $t(M, N, (m, n))$ 是 M 状态下使能的,当且仅当:

1) $\exists P' \subseteq \bullet t$,使得 $|P'|=m$,对于 $\forall p_i \in P'$,其中 $0 < i \leq m, p_i$ 是 M 状态下可用的,并且

2) $\exists tk_i \in p_i.TK, tk_i$ 是可调度的,并且 $(t.s=S_0) \wedge t.guard(tk_1, tk_2, \dots, tk_m)$ ($guard$ 与每一个可调度托肯的相关数据和控制数据相关).

(2) $t(M, N, (m, n))$ 是 M 状态下可点火的(记作 $M[t>]$,当且仅当:

1) $t(M, N, (m, n))$ 是 M 状态下使能的,并且

2) $(t.s=S_1) \wedge During(Now, tk, VT) \wedge During(Now, abs(t.FD)) \wedge During(Now, t.R.VT) \wedge During(Now, t.AP.VT)$.

定义 3.6(路由变迁的点火/实施(fire)). 如果路由变迁 $t(M, N, (m, n))$ 可点火,即 $M[t>]$,那么 $t(M, N, (m, n))$ 能够点火/实施,点火后, t 从 M 个输入库所中选择 m 个各删除一个可调度的托肯,并在 N 个输出库所中选择 n 个为它们各产生一个托肯,达到一个新的状态 M' ,记作 $M[t>M']$.

从以上定义可以看出:

普通变迁的动态语义是路由变迁的特例,它相当于 $t(1, 1, (1, 1))$ 的动态语义.因此,它们的动态语义是一致的,这为后续模型分析和实现带来了方便.

TPWF-net 虽然在静态结构上可以对应一个结构相同的 WF-net,但在动态语义上却有很大不同.这一方面表现在变迁的使能、点火条件结合了时态元素及其演算;另一方面是路由变迁的点火规则,它与 Petri 网的定义已有很大的不同,它不需要同步(即对于有 M 个输入库所的变迁 $t(M, N, (m, n))$,只要其中有 m 个可用变迁就可点火).这是根据工作流应用场景中一些经常性的路由需求而进行的扩充.

定义 3.7(可达性). 若存在 $t \in T$,使得 $M[t>M']$,则称 M' 从 M 直接可达;若存在变迁序列 $\sigma=t_1, t_2, \dots, t_{n-1}$ (σ 可为空),使得 $\forall i=1, 2, \dots, n-1$,都有 $M_i[t_i>M_{i+1}]$,记作 $M_1[\sigma>M_n]$,则称 M_n 从 M_1 可达,一般性地记作 $M_1[\ast>M_n]$.

使用 $(TPWN, M)$ 表示一个拥有初始状态 M 的 TPWF-net 网 TPWN,其他有关 TPWF-net 网活性、有界性、安全性、良构性、路径(基本路径、无冲突路径)、强连接、自由选择等概念的定义仍然沿用 Petri 网的定义,相关内容见文献[19].

4 TPWF-Net 的合理性分析

按 TPWF-net 规则构建的过程定义在投入执行前,必须对其进行合理性验证.合理性验证是保证过程定义质量的重要环节,它属于静态验证,即过程定义被执行前的检查,主要用于消除过程定义中的异常结构(比如死语句、死锁等),这一点如同在对源程序进行静态编译时,只能检查程序的语法结构是否正确,而不能查出与程序语义相关的错误.

4.1 合理性的概念

Aalst 定义了 WF-net 合理性的概念^[19].由于 WF-net 不考虑案例的语义,因此其合理性反映的是过程结构的正确性.对于时态工作流过程模型 TPWF-net 来说,其合理性不仅涉及过程的结构,还与模型各元素(包括案例)的时间属性相关,其中,案例的时间属性也可以理解为是它的时态语义.这里,以是否涉及业务语义为界线来区分合理性与正确性两个概念,以避免与程序正确性证明相关概念的类比联想.当前,不少相关研究中的正确性概念实际是指合理性,即与业务语义无关.

定义 4.1(TPWF-net 的合理性). 一个 TPWF-net $TPN=(P,T,F)$ 是合理的,当且仅当:

对于每一个从状态 i 可达的状态 M ,存在一个变迁序列,使得可从 M 到状态 o 是可达的,形式化表示为

$$\forall_M(i \xrightarrow{*} M) \Rightarrow M \xrightarrow{*} o.$$

状态 o 是从状态 i 可达的唯一最终状态,且结束时其中至少会有一个托肯,形式化表示为

$$\forall_M(i \xrightarrow{*} M \wedge M \geq o) \Rightarrow M = o.$$

在 (PN, i) 中没有死变迁,形式化表示为 $\forall_{t \in T} \exists_{M, M'} i \xrightarrow{*} M \wedge M[t \rightarrow M']$.

可以看到,定义 4.1 与 WF-net 的合理性定义^[19]只有一个符号不同,即 $\xrightarrow{*}$ 与 \rightarrow .这说明模型的合理性与其动态语义相关.

比较 TPWF-net 和 WF-net 的动态语义,如果 TPWF-net 忽略对模型元素时态属性及拖肯所包含的案例业务语义的约束,即假定 TPWF-net 动态语义中的时态元素逻辑运算表达式以及 $guard()$ 恒为 True,则 TPWF-net 和 WF-net 唯一的差别是:路由变迁的使能和点火规则不同,前者不要求同步,即对于有 M 个输入库所的路由变迁 $t(M, N, (m, n))$,只要其中有 m 个包含多于一个拖肯,变迁就被使能并可以点火实施,并且实施结果是在 N 个输出库所中选择 n 个产生拖肯(非确定性).这一点也违反了传统 Petri 网的变迁实施规则.

4.2 TPWF-Net 合理性分析

由 TPWF-net 合理性定义可见,其验证可以分为两个方面:

1) 过程结构合理性.假定 TPWF-net 动态语义中的时态元素逻辑运算表达式以及 $guard()$ 恒为 True,合并使能和可点火两个概念.在这个假定前提下验证 TPWF-net 过程结构的合理性;

2) 时间约束一致性.保证 TPWF-net 所有动态语义中时态元素逻辑运算表达式为 True,并且满足过程的整体时间约束(如,案例的有效时间必须在过程实例有效时间范围内,等等).注意,这里仍然假定 $guard()$ 恒为 True,因为 $guard()$ 的运算涉及案例的业务语义信息,包含案例业务语义的模型正确性验证不属于本文研究的内容,这个问题实际上类似于程序正确性证明.

4.2.1 TPWF-net 过程结构合理性分析

为了清楚地分析 TPWF-net 的过程结构合理性问题,需要定义一些概念.

定义 4.2(WF-Net 的直接可达关系). 一个 WF-net $PN=(P,T,F)$, μ 是 PN 上的状态集,令 $\mu = \{(M_1, M_2) | M_1, M_2 \in \mu, \exists t \in T \text{ 使得 } M_1[t \rightarrow M_2]\}$,称 PN 上的关系 $\Phi = \langle \mu, P \rangle$ 为 PN 的状态直接可达关系.

定义 4.3(TPWF-Net 的直接可达关系). 一个 TPWF-net $TPN=(P,T,F)$, μ 是 TPN 上的状态集,令 $\mu = \{(M_1, M_2) | M_1, M_2 \in \mu, \exists t \in T \text{ 使得 } M_1[t \rightarrow M_2]\}$,称 TPN 上的关系 $\Phi = \langle \mu, P \rangle$ 为 TPN 的状态直接可达关系.

利用 WF-net 合理性的已有成果来分析 TPWF-net 的合理性是一个低成本方案,为此,首先要研究 TPWF-net 和 WF-net 之间的关系,这里在定义 4.2 和定义 4.3 的基础上定义它们之间的等价关系.

定义 4.4(结构等价). 设 TPN 是一个 TPWF-net, PN 是一个 WF-net, 若它们所对应的直接可达关系同构, 则称 TPN 与 PN 是结构等价的.

定义 4.5(等价性). TPN 与 TPN' 是两个 TPWF-net, 若它们所对应的直接可达关系同构, 则称 TPN 与 TPN' 是等价的.

同样, 可以仿照定义 4.5 来定义两个 WF-net 的等价性.

在本节过程结构合理性的假定前提下, TPWF-net 与 WF-net 的重要差别是 TPWF-net 的非同步性和非确定性, 为此定义以下概念:

定义 4.6(同步 TPWF-net). 设一个 TPWF-net $TPN=(P, T, F)$, 其中 $T=TP \cup Tr$, TP 是普通变迁的集合, Tr 是路由变迁的集合. 若对于 $\forall t(M, N, (m, n)) \in Tr$ 都有 $M=m$, 则称 TPN 是一个同步 TPWF-net.

定义 4.7(确定 TPWF-net). 设一个 TPWF-net $TPN=(P, T, F)$, 其中 $T=TP \cup Tr$, TP 是普通变迁的集合, Tr 是路由变迁的集合. 若对于 $\forall t(M, N, (m, n)) \in Tr$ 都有 $N=n$, 则称 TPN 是一个确定 TPWF-net.

定义 4.8(同步确定 TPWF-net). 设一个 TPWF-net $TPN=(P, T, F)$, 其中 $T=TP \cup Tr$, TP 是普通变迁集合, Tr 是路由变迁集合. 若对于 $\forall t(M, N, (m, n)) \in Tr$ 都有 $M=m$ 且 $N=n$, 则称 TPN 是一个同步确定 TPWF-net.

WF-net 的两个子类自由选择 WF-net 和良构 WF-net 的合理性可以在多项式时间内判定, 并且如果它们是合理的则是安全的, 并且也是 S 可覆盖的^[19]. 自由选择的 WF-net 能够满足日常应用的绝大部分需求, 结构良好意味着过程定义具有更好的可读性和可维护性. 为此, 也定义了 TPWF-net 的相应子类.

定义 4.9(自由选择). TPWF-net 是自由选择的, 当且仅当对于任意两个变迁 t_1 和 t_2 , $\bullet t_1 \cap \bullet t_2 \neq \emptyset \Rightarrow t_1 = t_2$.

定义 4.10(良构). 一个 TPWF-net TPN 是良好结构(良构)的, 当且仅当 TPN 是良好处理的.

有关良好处理等概念见文献[19].

为了证明 TPWF-net 和 WF-net 的等价关系, 先证明两个引理.

引理 4.1. 设 $TPN=(P, T, F)$ 是一个 TPWF-net, $\Phi=\langle \mu, P \rangle$ 是 TPN 的直接可达关系; $PN=(P', T', F')$ 是一个 WF-net, $\Phi'=\langle \mu', P' \rangle$ 是 PN 的直接可达关系. 如果存在一个双射函数 $f: \mu \rightarrow \mu'$, 使得 $\forall M_1, M_2 \in \mu, t \in T$, 且 $M_1[t \rightarrow M_2, \exists t' \in T'$ 使得 $f(M_1)[t' \rightarrow f(M_2)]$, 则 TPN 与 PN 结构等价.

证明: 根据引理已知条件, 因为 $f: \mu \rightarrow \mu'$ 是一个 TPN 与 PN 状态集之间的双射, 又因为对于 $\forall M_1, M_2 \in \mu, t \in T$, 且 $M_1[t \rightarrow M_2, \exists t' \in T'$ 使得 $f(M_1)[t' \rightarrow f(M_2)]$, 即 f 保持 TPN 上的状态之间的直接可达关系, 所以根据同构的定义, TPN 和 PN 上的状态直接可达关系同构. 根据定义 4.4, TPN 与 PN 结构等价. 本引理得证.

同理可以证明:

引理 4.2. 设 $TPN=(P, T, F)$ 和 $TPN'=(P', T', F')$ 是两个 TPWF-net, $\Phi=\langle \mu, P \rangle$ 和 $\Phi'=\langle \mu', P' \rangle$ 分别是 TPN 和 TPN' 的直接可达关系. 如果存在一个双射函数 $f: \mu \rightarrow \mu'$, 使得 $\forall M_1, M_2 \in \mu, t \in T$, 且 $M_1[t \rightarrow M_2, \exists t' \in T'$ 使得 $f(M_1)[t' \rightarrow f(M_2)]$, 则 TPN 与 TPN' 等价.

有了以上概念和引理, 下面证明 TPWF-net 与 WF-net 之间的结构等价性.

定理 4.1. 对任何一个同步确定 TPWF-net 可构造一个 WF-net 与它结构等价.

证明: 给定一个同步确定 TPWF-net $TPN=(P, T, F)$, 其中 $T=TP \cup Tr$, TP 是普通变迁的集合, Tr 是路由变迁的集合. 在假定 TPWF-net 动态语义中的时态元素逻辑运算表达式以及 $guard()$ 恒为 True 的前提下, 采用如下方法来构造一个 WF-net $PN=(P', T', F')$, 设 $\Phi=\langle \mu, P \rangle$ 是 TPN 的直接可达关系, $\Phi'=\langle \mu', P' \rangle$ 是 PN 的直接可达关系. 其中:

- 1) 对于 $\forall p_k \in P$, 其中, $k=1, 2, 3, \dots, Kp$, Kp 是 P 中元素的个数, 创建一个 p'_k 加入集合 P' .
- 2) 对于 $\forall t_k \in T$, 创建一个 t'_k 加入集合 T' , 并且对于 $\forall (p_i, t_k) \in F$ 和 $\forall (t_k, p_j) \in F$, 分别创建弧 (p'_i, t'_k) 和弧 (t'_k, p'_j) 加入集合 F' , 其中, $k=1, 2, 3, \dots, Kt$, Kt 是 T 中元素的个数.

由 $PN=(P', T', F')$ 的构造方法知, TPN 与 PN 的所有元素一一对应, 具有相同的静态连接关系. 并且在假定 TPWF-net 动态语义中的时态元素逻辑运算表达式以及 $guard()$ 恒为 True 的前提下, 由于:

- 1) 对于 $\forall t \in TP$, 普通变迁 t 只有一个输入库所和一个输出库所, 在假定前提下, 根据定义 4.12、定义 4.13 和定义 4.14 得知, 它的实施规则与 WF-net 的实施规则一致, 都遵从传统 Petri 网的变迁实施规则;

- 2) 对于 $\forall t(M, N, (m, n)) \in Tr$, 由于 TPN 是同步确定 TPWF-net, 由定义 4.8 可知 $M=m$ 且 $N=n$, 在假定前提下, 根据定义 3.2、定义 3.5 和定义 3.6 得知, 它的实施规则也与 WF-net 的实施规则一致。

所以, TPN 和 PN 的静态结构和动态实施规则都是一致的, 因而其可达图是同构的, 亦即存在一个双射函数 $f: \mu \rightarrow \mu'$, 使得 $\forall M_1, M_2 \in \mu, t \in T$, 且 $M_1[t > M_2, \exists t' \in T'$, 使得 $f(M_1)[t'] > f(M_2)$ 。由引理 5.1 可知, TPN 与 PN 结构等价。

定理 4.2. 对任何一个 TPWF-net 可构造一个同步确定 TPWF-net 与它等价。

证明: 给定一个 TPWF-net $TPN=(P, T, F)$, 其中 $T=Tp \cup Tr$, Tr 是路由变迁的集合。采用如下方法构造一个 TPWF-net $TPN'=(P', T', F')$, 设 $\Phi=\langle \mu, P \rangle$ 是 TPN 的直接可达关系, $\Phi'=\langle \mu', P' \rangle$ 是 TPN' 的直接可达关系。其中:

- 1) $P'=P$.
- 2) $Tr'=Tr$, 并且对于 $\forall (p_i, t_k) \in F$ 和 $\forall (t_k, p_j) \in F$, 分别创建弧 (p_i, t_k) 和弧 $\forall (t_k, p_j)$ 加入集合 F' , 其中, $k=1, 2, 3, \dots, K$, K 是 Tr 中元素的个数。
- 3) 对于 $\forall t_k(M, N, (m, n)) \in Tr$, 创建 $C_M^m C_N^n$ 个变迁组成的集合 U_k , 其中, $U_k \subseteq Tr'$, $Tr' \subseteq T'$, $t_{kij}(m, n, (m, n)) \in U_k$ (其中: $k=0, 1, 2, \dots, K$, K 是 Tr 中元素的个数; $i=1, 2, 3, \dots, C_M^m$, i 代表从 M 个输入库所中任取 m 个的第 i 种组合; $j=1, 2, 3, \dots, C_N^n$, j 代表从 N 个输出库所中任取 n 个的第 j 种组合), 并且对于 $\forall i, j, g, h$, 其中, $g=1, 2, 3, \dots, m, h=1, 2, 3, \dots, n$, 创建弧 (p_{kig}, t_{kij}) 和 (t_{kij}, p_{kjh}) 加入集合 F' 。这里, p_{kig} 表示 t_k 的 M 个输入库所中取 m 个的第 i 种组合的第 g 个库所, p_{kjh} 表示 t_k 的 N 个输出库所中取 n 个的第 j 种组合的第 h 个库所。

为了清楚说明这种构造思想, 以路由 $t(3, 2, (2, 1))$ 为例, 将构造与之等价的同步确定 WF-net 元素的方法描述在图 5 中。为了图示简洁性, 同时不失一般性, 图 5 中的变迁下标省却了第 1 维 k 。

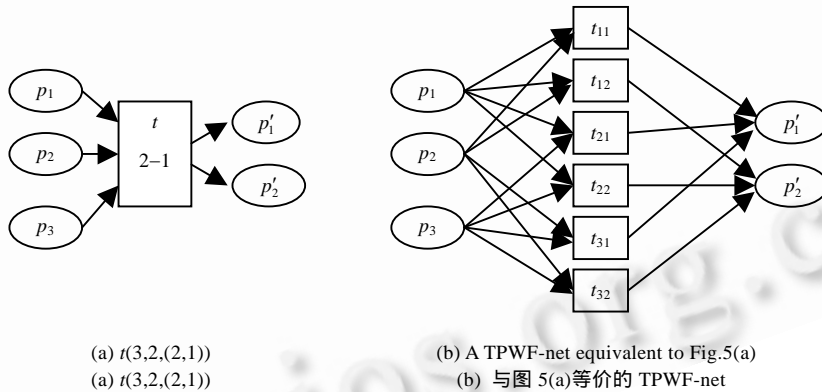


Fig.5 An example for equivalent conversion of TPWF-net's router

图 5 TPWF-net 路由的等价转换示例图

由以上构造方法得知, 存在一个双射函数 $f_{tr}: Tr \rightarrow Tr'$, 其中, $Tr' \subseteq T'$, 使得对于 $\forall k \in [1 \dots K]$ 都有 $f_{tr}(t_k) = U_k$, 其中, $t_k \in Tr$, $U_k \in Tr'$, K 是 Tr 中元素的个数, t_k 与 U_k 的前后集是一一对应的。并且由于 U_k 中变迁与前后集库所的连接穷尽了所有输入库所、输出库所的组合, 对于 t_k 实施后的每一种结果状态, 在 U_k 中总能够找到唯一一个 t_{kij} , 其实施后的状态与之相等。而且 Tr' 中的所有路由变迁都是同步和确定的, 因此 TPN' 是一个同步确定 TPWF-net。

- 4) 最后, 合并以上各步创建的变迁集合 $Tr'=U_1 \cup U_2 \cup \dots \cup U_K$, 其中, K 是 Tr 中元素的个数, 进而 $T'=Tp' \cup Tr'$ 。

由 TPN 构造 TPN' 的过程可知, TPN 和 TPN' 具有相同的状态集, 即 $\mu = \mu'$, 因此在 μ 和 μ' 之间存在一个自双射。对于 $\forall M_1, M_2 \in \mu, t \in T$, 且 $M_1[t > M_2$:

- (1) 当 $t \in Tp$ 时, $\exists t' \in T'$, t 与 t' 同名, 使得 $M_1[t'] > M_2$;
- (2) 当 $t \in Tr$ 时, $\exists t' \in f_{tr}(t) \subseteq T'$, 使得 $M_1[t'] > M_2$ 。

由引理 4.2 可知, TPN 与 TPN' 等价。定理得证。

推论 4.1. 对任何一个 TPWF-net 可构造一个 WF-net 与它结构等价.

证明:给定一个 TPWF-net TPN ,由定理 4.2 可知,可以构造一个等价的同步确定 TPWF-net TPN' ,再根据定理 4.1,可以构造一个 WF-net PN 与 TPN' 结构等价.于是, TPN 与 PN 结构等价.

推论 4.2. 对任何一个自由选择的同步 TPWF-net 可构造一个自由选择 WF-net 与它结构等价.

证明:由定理 4.2 的构造过程可知,与同步 TPWF-net 路由变迁等价的变迁集合中的所有变迁拥有共同的前集,因此,新构造的网不会改变原来网的自由选择特性;再由定理 4.1 可知,其构造过程也不改变网的自由选择特性.因此结论成立.

推论 4.3. 对任何一个良构的 TPWF-net 可构造一个良构的 WF-net 与它结构等价.

证明:由定理 4.2 的构造过程可知,与同步 TPWF-net 路由变迁等价的变迁集合以及它的前集和后集所构成的网是同步确定 TPWF-net,它本身是良构的;再由定理 4.1 得知,其构造过程也不改变网的良构特性.因此结论成立.

由以上 3 个推论,结合文献[19]已证明的结论(自由选择 WF-net 和良构 WF-net 的合理性可以在多项式时间内判定,如果它们是合理的则是安全的,并且也是 S 可覆盖的),可以得到以下定理:

定理 4.3. 一个自由选择的同步 TPWF-net 的过程合理性可以在多项式时间内判定.

定理 4.4. 如果一个自由选择的同步 TPWF-net 是过程合理的,则它也是过程安全的.

定理 4.5. 一个良构的 TPWF-net 的合理性可以在多项式时间内判定.

定理 4.6. 如果一个良构的 TPWF-net 是过程合理的,则它也是过程安全的.

由以上定理可知,TPWF-net 是自由选择的,与它结构等价的 WF-net 不一定是自由选择,只有它同时也是同步的才可以保证其自由选择特性.而良构 TPWF-net 可以得到与 WF-net 相似的结论.

进一步仿照文献[19]对 WF-net 的合理性组合定理的证明,可以得到 TPWF-net 的相应性质:

定理 4.7(组合定理). $TPN_1=(P_1,T_1,F_1)$ 和 $TPN_2=(P_2,T_2,F_2)$ 是两个 TPWF-net,满足 $T_1 \cap T_2 = \emptyset, P_1 \cap P_2 = \{i, o\}$,而且 $t^+ \in F_1$.假设 $TPN_3=(P_3,T_3,F_3)$ 是用 P_2 替换 P_1 中的 t^+ 后得到的 TPWF-net,即 $P_3=P_1 \cup P_2, T_3=(T_1 \setminus \{t^+\}) \cup T_2$,且 $F_3=\{(x,y) \in F_1 | x \neq t^+ \wedge y \neq t^+\} \cup \{(x,y) \in F_2 | \{x,y\} \cap \{i,o\} = \emptyset\} \cup \{(x,y) \in P_1 \times T_2 | (x,t^+) \in F_1 \wedge (i,y) \in F_2\} \cup \{(x,y) \in T_2 \times P_1 | (t^+,y) \in F_1 \wedge (x,o) \in F_2\}$. (TPN_1,i) 和 (TPN_2,i) 是安全且合理的,当且仅当 (TPN_3,i) 是合理且安全的.

上述定理 4.3~定理 4.7 证明均省略.

4.2.2 TPWF-net 的结构化建模规则

根据第 4.2.1 节对 TPWF-net 结构特性和过程合理性的分析,自由选择的同步 TPWF-net 或良构 TPWF-net 的合理性和安全性可以被高效地分析(多项式时间),建模时应尽量满足 TPWF-net 的自由选择、同步特性或良构性.

根据定理 4.7 的结论,可以先选择一些合理且安全的基本构造块,通过组合这些构造块形成的过程模型也是合理且安全的.事实上,这与程序设计领域的结构化编程思想非常类似,其理论基础是结构化定理.在工作流领域,由于过程模型中并行、同步/非同步、确定/非确定等概念的出现,使得它比程序设计理论更加复杂,至今还无人证明工作流领域的结构化定理,或者证明该定理不存在.该问题是工作流理论中最基础的问题,被解决前工作流理论始终存在重大缺陷.Aalst 在 <http://www.workflowpatterns.com/>上已公布了 43 种工作流模式,并且还在继续研究和补充,目前无法确定基本构造块是有限多还是无限多.

工作流理论上的缺陷并不妨碍利用它显现的优势,Aalst 给出了 7 个基础结构^[19],它们都是合理且安全的.虽然无法从理论上精确地说明这 7 个基础结构的构造能力有多强,但 Aalst 还是指出它们的能力已经足够强大了.通过对 TPWF-net 结构特性的分析,本文提出了一组基础过程结构(见表 1),同样,现在还不能证明它们比 Aalst 的 7 个基础结构具有更强的能力,但它们在表达一些常见的业务过程时会更加简洁.

通过对 TPWF-net 结构特性和过程合理性的综合分析,并在工作流建模领域引入结构化编程思想的理念,TPWF-net 的结构化建模必须遵守以下 5 个规则:

规则 4.1. 使用表 1 所列的基础过程结构的组合来建模,组合的方式是:

- 1) 顺序连接(一个结构的输出库所与另一个结构的输入库所合并);或者
- 2) 完全嵌套(用一个结构去替换另一个结构中的一个元结构).

规则 4.2. 在建模时使用反复/循环结构的前提是保证反复/循环能在有限次后终止,否则,任何包含反复/循环的过程模型都是不合理的.

规则 4.3. 当规则 4.1 无法适用时,尝试使用符合自由选择且同步、或者良构的形式构建 TPWF-net.

规则 4.4. 当规则 4.1、规则 4.3 无法适用时,慎重采用其他建模形式,并仔细分析其合理性.

规则 4.5. 将新识别的合理且安全的基本结构补充到表 1 中.

Table 1 Basic building blocks of TPWF-net

表 1 TPWF-net 的基础过程结构

Name	Basic building block
1. Sequence	
2. Implicit selection	
3. Explicit selection	
4. Parallel	
5. Selective parallel	
6. Iteration	
7. Repeat	
8. While	

Note: For each block, the most left place is the input place, and the most right place is the output place.

表 1 的基础构造模块参考了 Aalst 的 7 个基本结构^[19],并针对 TPWF-net 结构特性和合理性分析的结论做如下调整:

- 1) 去除显式或并结构(explicit OR-join construct).在 TPWF-net 中,路由变迁不承担业务任务,因此,显式或并与隐式或并的语义一致,而后者更简洁.隐式或并的用法参见表 1 中的隐选结构,它通常与隐式选择路由(implicit OR-split)搭配使用;
- 2) 增加了选并结构.这是 TPWF-net 提供的对常用业务流程中常见结构形式的描述.虽然用基本的 Petri 网也可以描述,但描述形式却复杂得多.该结构中,由于路由变迁 t 的非同步特性,导致该结构不是自由选择的同步 TPWF-net,但它仍是良构的;
- 3) 增加了循环结构(repeat 和 while).限定循环在有限次结束的前提下,它们仍是合理且安全的.

表 1 的基本过程结构有一个共同的特点:只有一个输入接口库所和一个输出接口库所.因为 TPWF-net 的所有普通变迁(代表业务任务)都是单入口/单出口的,这些基本的构造块可以很好地支持 TPWF-net 的结构化建模方法.

4.2.3 TPWF-net 的时间约束一致性分析

过程模型的时间约束一致性分析是指在不考虑业务语义的前提下(假定 $guard()$ 恒为 True),分析其局部的和全局的时间约束是否满足.所谓局部时间约束是指保证 TPWF-net 所有动态语义中时态元素逻辑运算表达式为 True,而全局时间约束是指案例的有效时间必须在过程实例有效时间范围内.

注意,这里的分析是静态分析.

• 时间约束一致性的指标

在静态分析时,过程模型中所有的时间属性都是以相对时间表示的.通过对时态工作流元模型中元素的时间属性和 TPWF-net 动态语义的分析,局部动态语义的时态元素逻辑运算表达式都是定义的动态约束,即保证动态的调度时间、点火时间符合相应的时间区间约束,还要检查执行时间是否在用户有效期和工具有效期,因此不属于静态分析的范围.在进行静态分配时,假定这些局部时间约束是满足的(否则会触发超时异常,WFMS 按相应的异常处理策略进行处理).

在时态工作流中,案例被赋予独立的生命周期,它可能在过程实例尚未完成时失效.因此,必须通过静态分析保证这一点.但这个保证要区分为保证有可能满足约束和保证绝对满足约束两种情况:前一种情况是指在过程实例的执行路径按照时间最短执行时能够满足案例的有效时间约束;后一种情况是指在过程实例的执行路径按照时间最长执行时也能够满足案例的有效时间约束.它们都是在假定过程执行中不发生超时异常情况下的指标.为了给发生超时异常时进行异常处理提供依据,还必须给出最大的可调整时间.

设案例的全局有效期 $VT=[V_{\min}, V_{\max}]_{PCT}$, PCT 是过程创建时间.基于以上分析,给出以下验证时间约束一致性需要的指标:

- 1) 最大过程执行时间 T_{\max} :表示在过程实例的执行路径按照时间最长进行路由时所耗费的时间.以时间距离类型^[2,3]表示.
- 2) 最小过程执行时间 T_{\min} :表示在过程实例的执行路径按照时间最短进行路由时所耗费的时间.以时间距离类型表示.
- 3) 绝对可调整时间 δ_{\min} : $\delta_{\min}=V_{\max}-end(T_{\max})$.案例创建后不一定立刻生效,即 V_{\min} 可能不为 0,但它并不影响 δ_{\min} 的计算.
- 4) 相对可调整时间 δ_{\max} : $\delta_{\max}=V_{\max}-end(T_{\min})$.

定义 4.11(时间约束绝对一致). 当 TPWF-net 的绝对可调整时间 $\delta_{\min} \geq 0$ 时,称该过程是时间约束绝对一致的.

定义 4.12(时间约束相对一致). 当 TPWF-net 的相对可调整时间 $\delta_{\max} \geq 0$ 且 $\delta_{\min} < 0$ 时,称该过程是时间约束相对一致的.

定义 4.13(绝对合理). 当 TPWF-net 同时满足过程合理性和时间约束绝对一致性时,称其绝对合理.

定义 4.14(相对合理). 当 TPWF-net 同时满足过程合理性和时间约束相对一致性时,称其相对合理.

注意,静态验证不考虑平均情况或期望值,那是动态验证时关心的问题.

- 基本过程结构的时间计算方法

TPWF-net 时间计算的基础是对表 1 中基本过程结构的时间指标进行计算,这里借鉴了 Marjanovic 计算过程延迟的思想^[22],并结合 TPWF-net 的特点来计算.

1) 串行(sequence)

$$T_{\max}=T_{\min}=[p.x+t_1.x+t_1.u+q.x+t_2.x+t_2.u+r.x.p.y+t_1.y+t_1.v+q.y+t_2.y+t_2.v+r.y],$$

其中, $p.x, p.y$ 分别表示库所 p 调度延迟 $[x, y]$ 的开始时间点和结束时间点.其他表示类似,以后不再解释.

因为这里的时间点是相对时间,并且各相对时间点相对的参照基准点各不相同,且基准点之间有顺序发生的限制,所以这里的相对时间点加法是有意义,其值是整数值相加的结果.

2) 隐选(implicit selection)

$$T_{\max}=[p.x+\max(t_1.x+t_1.u, t_2.x+t_2.u)+q.x.p.y+\max(t_1.y+t_1.v, t_2.y+t_2.v)+q.y],$$

$$T_{\min}=[p.x+\min(t_1.x+t_1.u, t_2.x+t_2.u)+q.x.p.y+\min(t_1.y+t_1.v, t_2.y+t_2.v)+q.y].$$

3) 显选(explicit selection)

$$T_{\max}=[p.x+t.x+t.u+\max(q_1.x+t_1.x+t_1.u, q_2.x+t_2.x+t_2.u)+r.x.p.y+t.x+t.u+\max(q_1.y+t_1.y+t_1.v, q_2.y+t_2.y+t_2.v)+r.y],$$

$$T_{\min}=[p.x+t.x+t.u+\min(q_1.x+t_1.x+t_1.u, q_2.x+t_2.x+t_2.u)+r.x.p.y+t.x+t.u+\min(q_1.y+t_1.y+t_1.v, q_2.y+t_2.y+t_2.v)+r.y].$$

4) 并行(parallel)

$$T_{\max}=[p.x+t.x+t.u+\max(q_1.x+t_1.x+t_1.u+r_1.x, q_2.x+t_2.x+t_2.u+r_2.y)+t'.x+t'.u+s.x.p.y+t.y+t.v$$

$$+\max(q_1.y+t_1.y+t_1.v+r_1.y, q_2.y+t_2.y+t_2.v+r_2.y)+t'.y+t'.v+s.y],$$

$$T_{\min}=[p.x+t.x+t.u+\min(q_1.x+t_1.x+t_1.u+r_1.x, q_2.x+t_2.x+t_2.u+r_2.y)+t'.x+t'.u+s.x.p.y+t.y+t.v$$

$$+\min(q_1.y+t_1.y+t_1.v+r_1.y, q_2.y+t_2.y+t_2.v+r_2.y)+t'.y+t'.v+s.y].$$

5) 选并(selective parallel)

$$T_{\max}=[p.x+t.x+t.u+\max(q_1.x+t_1.x+t_1.u+r_1.x, q_2.x+t_2.x+t_2.u+r_2.y, \dots, q_N.x+t_N.x+t_N.u+r_N.y)+t'.x+t'.u+s.x.p.y+t.y+t.v$$

$$+\max(q_1.y+t_1.y+t_1.v+r_1.y, q_2.y+t_2.y+t_2.v+r_2.y, \dots, q_N.y+t_N.y+t_N.v+r_N.y)+t'.y+t'.v+s.y],$$

$$T_{\min}=[p.x+t.x+t.u+\min_n(q_1.x+t_1.x+t_1.u+r_1.x, q_2.x+t_2.x+t_2.u+r_2.y, \dots, q_N.x+t_N.x+t_N.u+r_N.y)+t'.x+t'.u+s.x.p.y+t.y+t.v$$

$$+\min(q_1.y+t_1.y+t_1.v+r_1.y, q_2.y+t_2.y+t_2.v+r_2.y, \dots, q_N.y+t_N.y+t_N.v+r_N.y)+t'.y+t'.v+s.y].$$

其中, $\min_n()$ 表示倒数第 n 小,即将参数从大到小排序,倒数第 n 个元素的值.

6) 重复(Iteration)

$$T_{\max}=[K \times (p.x+t_1.x+t_1.u+q.x+t_2.x+t_2.u), K \times (p.y+t_1.y+t_1.v+q.y+t_2.y+t_2.v)],$$

其中, K 是最大重复次数.

$$T_{\min}=[p.x+t_1.x+t_1.u+q.x.p.y+t_1.y+t_1.v+q.y].$$

7) Repeat

$$T_{\max}=[K \times (p.x+t_1.x+t_1.u+q.x+t_2.x+t_2.u)+r.x, K \times (p.y+t_1.y+t_1.v+q.y+t_2.y+t_2.v)+r.y],$$

其中, K 是最大循环次数.

$$T_{\min}=[p.x+t_1.x+t_1.u+q.x+t_2.x+t_2.u+r.x.p.y+t_1.y+t_1.v+q.y+t_2.y+t_2.v+r.y].$$

8) While

$$T_{\max}=[K \times (p.x+t.x+t.u+q.x+t_1.x+t_1.v)+r.x, K \times (p.y+t.y+t.v+q.y+t_1.y+t_1.v)+r.y],$$

其中, K 是最大循环次数.

$$\min=[p.x+t.x+t.u+r.x.p.y+t.y+t.v+r.y].$$

5 TPWF-Net 的合理性验证

目前,过程模型合理性的验证方法主要有两类:基于结构化简的方法和基于可达图的方法.前者更加直观并便于人工验证,而后者在构造可达图和基于它进行验证时,需要大量的时间空间消耗,只适用于计算机辅助的自

动化验证,并且太复杂的过程验证会导致状态爆炸.本文主要采用基于结构化简的方法.

定理 4.7、结构化建模规则和表 1 为进行结构化简打下了重要基础.基于结构化简的过程实质上是结构化建模的逆过程,其基本思想是:用一个简单的标准构造块去替换过程模型中一个基础的构造块,并保持两个构造块的输入/输出一致和时间约束指标一致,从而得到一个更简化的结构;重复以上步骤直到不能再简化;如果最终得到的简化结构是安全且合理的,则化简前的过程模型是安全且合理的.

首先,基于所有业务逻辑都可以表示为单入口/单出口的思想,定义如图 6 所示的标准构造块.

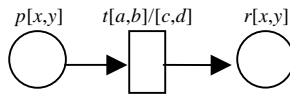


Fig.6 Standard building block

图 6 标准构造块

图 6 中, $t[a,b]/[c,d]$ 的 $[a,b]$ 是按最大过程延迟计算时的待替换结构的等效时间延迟,而 $[c,d]$ 是按照最小过程延迟计算时的等效时间延迟.在进一步化简过程中,根据两种情况在两者之间选择.

过程模型化简的步骤:

- 1) 从开始库所以深度优先或广度优先遍历过程模型,识别符合表 1 定义基础过程结构,并以标准构造块替换它.
- 2) 重复以上步骤直到过程模型没有表 1 定义的基础过程结构.
- 3) 辨别化简后的过程结构是否是标准构造块:若是,则该过程是过程合理且安全的;否则,人工根据过程结构合理且安全的定义进行判断.
- 4) 若过程合理且安全,从化简后的模型计算出 T_{\max} 和 T_{\min} ,并给出时间约束一致性的条件:
 - (1) 过程模型绝对合理的条件是 $V_{\max} \geq \text{end}(T_{\max})$;
 - (2) 过程模型相对合理的条件是 $\text{end}(T_{\min}) \leq V_{\max} < \text{end}(T_{\max})$.
- 5) 计算并输出 δ_{\min} 和 δ_{\max} .

对于以上化简步骤有几点需要说明:

- 1) 对于任意一个复杂流程,不一定总能化简为一个标准构造块或一个足够简单的结构模型,当无法确切证明化简后模型的合理性时该方法就会失败.只有在建模阶段遵守结构化建模的规则才能为后续模型验证带来便利.
- 2) 在使用以上方法进行模型验证时,如果最终得不到标准构造块,而又能确切判别化简模型的合理和安全性时,可以将该化简后的模型增补到表 1 中.在工作流领域的结构化定理得出确切结论之前,无法确定还有多少需要的基本结构.
- 3) 连通图的遍历复杂性为 $O(n^2)$ 或 $O(n+e)$,该方法步骤 1 最多重复 n 次,因此,该方法的复杂度是 $O(n^3)$ 或 $O(n(n+e))$.

6 案例分析

如图 7 所示,是一个工厂报建审批的业务流程,各普通变迁的含义分别是:

- t_1 ——填写并提交申请;
- t_2 ——预审;
- t_3 ——环保审批,若超时最多可重复 3 次;
- t_4 ——规划审批;
- $t_5 \sim t_8$ ——综合审批投票,4 选 3 投票,2 票以上赞成通过;
- t_9 ——发证;

t_{10} ——拒绝通知;

t_{11} ——归档.

路由变迁 $r_1 \sim r_2$ 都是自动化活动,各变迁的状态规则和路由的转移规则不再描述,可见文献[2,3]用 ECA 形式描述.图中时间单位为 1/10 工作日.政府部门承诺:接到申请后,30 个~45 个工作日完成审批.

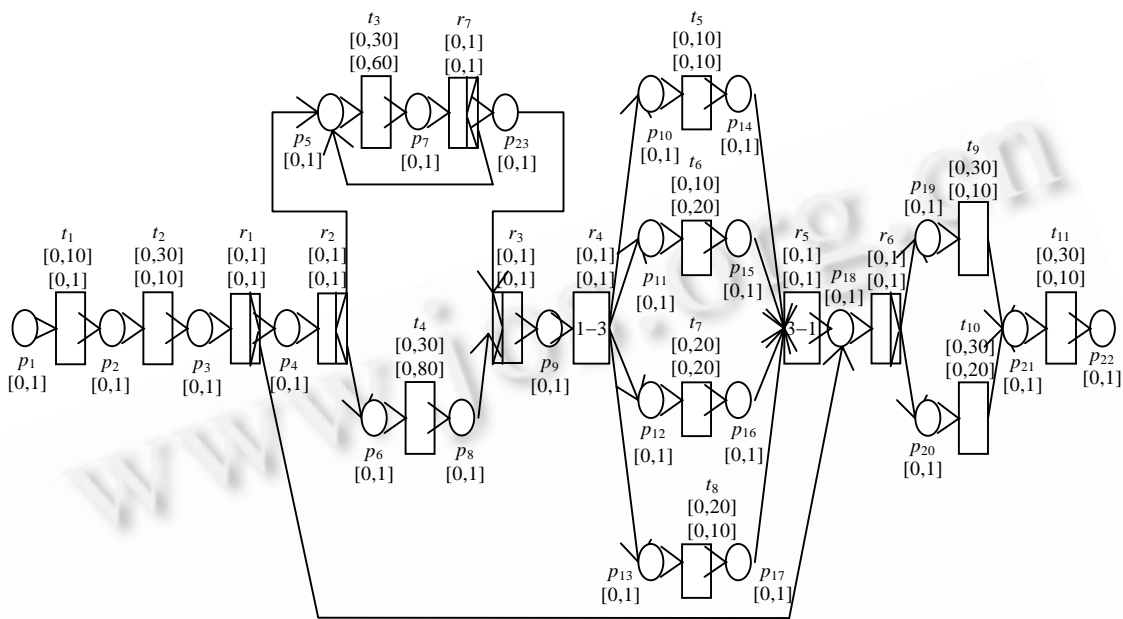


Fig.7 An example of TPWF-net model

图 7 一个 TPWF-net 建模实例

模型验证的过程如图 8 所示.图 8(a)是图 7 用标准块替换 1 号、7 号、5 号、3 号基本结构所得;图 8(b)是图 8(a)用标准块替换 4 号、1 号基本结构所得;图 8(c)是图 8(b)替换掉 1 号结构所得;图 8(d)是图 8(c)替换掉 3 号结构所得.注意,这里替换的是 3 号结构的一个特例,即一个选择分支上没有变迁和库所;图 8(e)是图 8(d)连续两次替换 1 号结构所得.

由于化简结果是一个标准构造块,因此该工厂报建的 TPWF-net 模型是过程合理的。

$$T_{\max}=[0+0+0,1+389+1]=[0,391],$$

$$T_{\min}=[0+0+0,1+140+1]=[0,142].$$

过程模型绝对合理的条件是 $V_{\max} \geq 391$; 过程模型相对合理的条件是 $142 \leq V_{\max} < 391$.

根据该案例的时间约束,若申请单的有效期限是 30 天,则过程模型只能达到相对合理性,相对可调整时间 $\delta_{\max}=300-142=158$.

若申请单的有效期是 45 天,则过程模型达到绝对合理性,绝对可调整时间 $\delta_{\min}=450-391=59$.

$\delta_{\max}, \delta_{\min}$ 可用于处理时间异常.

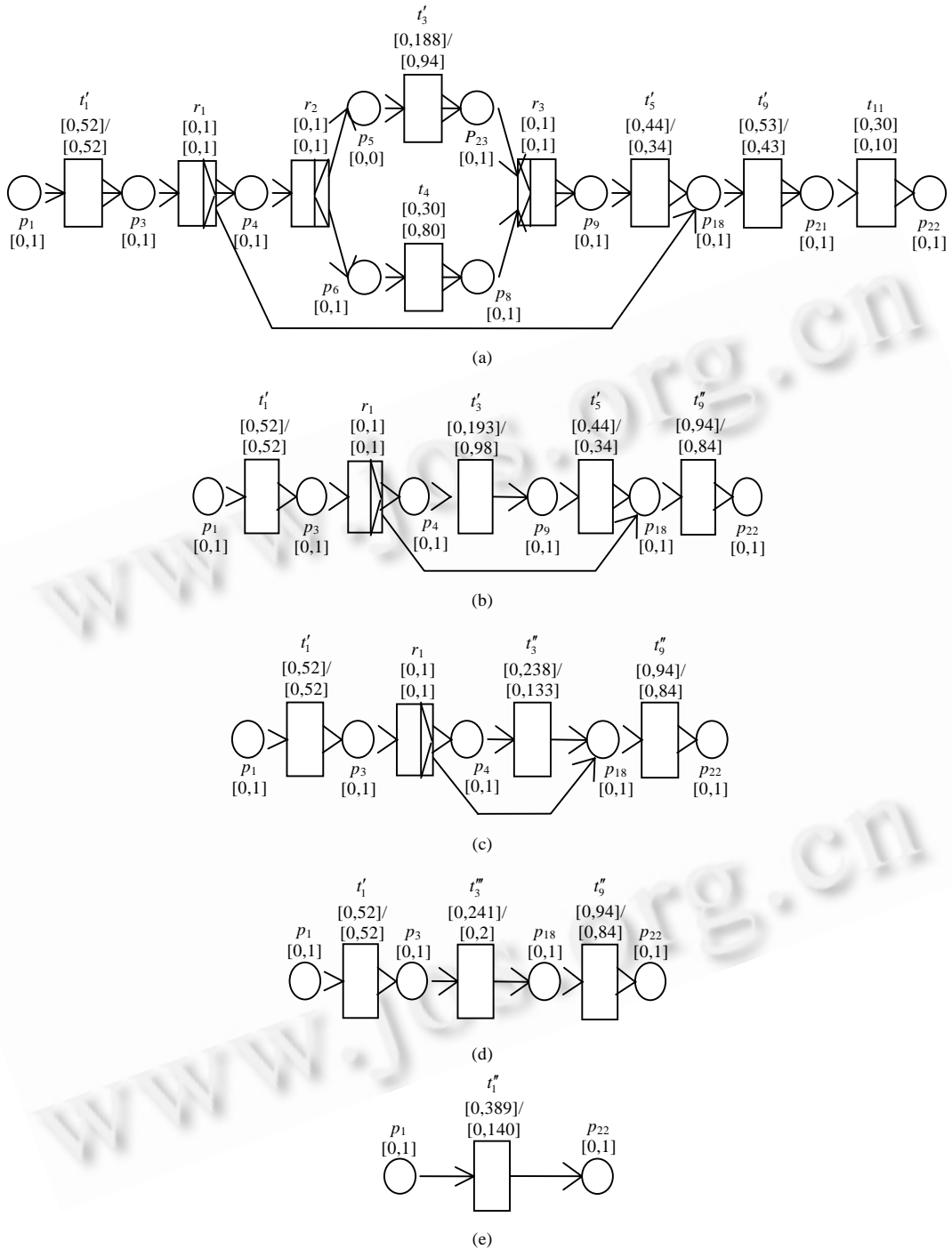


Fig.8 Simplifying process of a TPWF-net model

图 8 一个 TPWF-net 模型的化简过程

7 总结与展望

7.1 总 结

当前, workflow 领域对时间因素的研究主要基于为任务/活动赋予时间约束,通过工作流过程模型研究分析工作流的执行效率和能力,但对工作流其他维度元素的时态属性研究甚少,尤其是案例维。本文研究通过对现有工作流模型在时间建模方面的扩充和完善,试图为该问题的最终解决提供理论基础。

本文提出了时态工作流的概念,并基于对时态信息的表示及演算、时态工作流元模型的研究成果,对过程模型及其静态验证等基础问题进行了深入的研究。具体来讲,可分为如下两个方面:

- 1) 建立时态工作流过程模型。将工作流的基本概念映射为理论模型的基本元素,提出了能够综合描述过程、资源、案例和时间四维信息的时态工作流过程模型 TPWF-net,并给出了形式化的定义和模型的动态语义。
- 2) 研究过程模型合理性分析和验证的方法。证明了 TPWF-net 与 WF-net 的结构等价性、自由选择同步 TPWF-net 和良构 TPWF-net 的合理性可以在多项式时间内判定等结论。在此理论指导下,提出了结构化建模的思想,定义了一组基本过程结构和建模规则。最后,阐述了基于结构化简的 TPWF-net 模型验证方法,给出了化简步骤。

本文对时态工作流模型的初步研究为该理论体系的进一步发展奠定了基础,其研究成果不但使 workflow 中时间要素的描述和分析更加全面,而且有助于促进 workflow 相关问题(如柔性、事务处理等)的研究,并与时态数据库的研究互相促进。

我们基于现有研究成果和开源工作流产品 YAWL 初步开发了一个支持 TPWF-net 的时态工作流引擎。在广东省科技攻关项目的支持下,该技术已应用于广东省的免疫疾病信息采集。目前,在广东省科技攻关项目、广州市中小企业创新基金的支持下,该技术正在应用于突发卫生事件信息采集、IT 服务管理等对信息时效性要求较高的领域。时态工作流管理系统将成为电子政务、电子商务、企业信息化等领域的重要基础性平台,具有极大的应用价值。

7.2 进一步工作展望

综合分析本文的研究成果,值得进一步研究的问题包括:

- 1) 时态工作流模型验证的自动化。本文提出的基于结构化简的 TPWF-net 模型验证方法目前主要用于人工验证,如何实现自动化验证是一个有价值的研究方向。
- 2) 时态工作流模型的动态分析和时间规划。本文对时态工作流过程模型的分析仅限于静态的定性分析,其目的是验证模型的合理性。对时态工作流的动态分析将是一个充满挑战的研究领域,尤其是基于信息有效性的时间规划调整、活动时间的动态分配等问题的研究,将会充分发挥时态工作流对时间问题的处理能力。
- 3) 时态工作流的柔性 and 异常处理等相关问题。工作流的动态变化和异常处理与时间密切相关,如何利用时态工作流的时间信息处理这些问题值得进一步研究。
- 4) 分布式时态工作流的研究。本文的研究设定其应用背景是集中部署的 WFMS,分布式环境必考虑任务之间数据流动的延迟,它在时态工作流模型中如何表达、对其他问题有何影响,都是值得探讨的课题。
- 5) 时态工作流的实现技术。实现技术是时态工作流走向应用的关键一步,特别是以时态数据库技术为基础的实施方案研究将是非常有吸引力的。
- 6) 工作流结构化定理的研究。这个问题不仅仅是时态工作流的问题,而且是整个工作流理论的基础性问题。这个问题的存在,是 workflow 基础理论的一个重要缺陷。

References:

- [1] Hollingsworth D. The workflow reference model. Document Number, TC00-1003. Winchester: Workflow Management Coalition, 1995. 1–55. http://www.wfmc.org/index.php?option=com_docman&task=doc_download&gid=92&Itemid=72
- [2] Yu Y, Tang Y, Tang N, Ye XP, Liang L. A meta model of temporal workflow and its formalization. In: Jin H, *et al.*, eds. Proc. of the 3rd Int'l Conf. on Grid and Cooperative Computing. Heidelberg: Springer-Verlag, 2004. 987–992. <http://www.springerlink.com/content/56k64wwur4qm9q9b/>
- [3] Yu Y, Tang Y, Liang L, Pan Y, Tang N. Analysis of temporal extension of a workflow meta-model. Mini-Micro System, 2006,27(8): 1526–1530 (in Chinese with English abstract).
- [4] Zhuge H, Cheung TY, Pung HK. A timed workflow process model. The Journal of Systems and Software, 2001,55:231–243. [doi: 10.1016/S0164-1212(00)00073-X]
- [5] Du SZ, Tan JR, Lu GD. An extended time workflow model based on TWF-net and its application. Journal of Computer Research and Development, 2003,40(4):524–530 (in Chinese with English abstract).
- [6] Eder J, Panagos E, Pozewaunig H, Rabinovich M. Time management in workflow systems. In: Abramowicz W, Orlowska ME, eds. Proc. of the 3rd Int'l Conf. on Business Information Systems. Heidelberg, London, Berlin: Springer-Verlag, 1999. 265–280.
- [7] Ling S, Schmidt H. Time Petri nets for workflow modeling and analysis. In: Proc. of the IEEE Int'l Conf. on System, Man and Cybernetics. Nashville: IEEE Computer Society, 2000. 3039–3044. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00884464>
- [8] Li HF, Fan YS. Workflow model analysis based on time constraint Petri nets. Journal of Software, 2004,15(1):17–26 (in English with Chinese abstract). <http://www.jos.org.cn/1000-9825/15/17.htm>
- [9] Li W, Zeng GZ, Wang XL. A workflow model based on timed Petri net. Journal of Software, 2002,13(8):1666–1671 (in Chinese with English abstract). http://www.jos.org.cn/ch/readers/view_abstract.aspx?flag=1&file_no=20020845&journal_id=jos
- [10] Zhou JT, Shi ML, Ye XM. A method for semantic verification of workflow processes based on Petri net reduction technique. Journal of Software, 2005,16(7):1242–1251 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/1242.htm>
- [11] Zhao JL, Stohr EA. Temporal workflow management in a claim handling system. Software Engineering Notes, 1999,24(2): 187–195. [doi: 10.1145/295666.295687]
- [12] Li JQ, Fan YS. A method of workflow model performance analysis. Chinese Journal of Computers, 2003,26(5):513–523 (in Chinese with English abstract).
- [13] Li JQ, Fan YS, Zhou MC. Performance modeling and analysis of workflow. IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2004,34(2):229–242. [doi: 10.1109/TSMCA.2003.819490]
- [14] van der Aalst WMP. The application of Petri nets to workflow management. Journal of Circuits Systems and Computers, 1998, 8(1):21–66. [doi: 10.1142/S0218126698000043]
- [15] Hsu PY, Chen YL, Chang YB. A four dimensional Petri net approach for workflow management. In: Lee ML, Tan KL, Wuwongse V, eds. Proc. of the Database Systems for Advanced Applications. Berlin: Springer-Verlag, 2006. 617–627.
- [16] Combi C, Pozzi G. Towards temporal information in workflow systems. In: Olivé A, *et al.*, eds. Advanced Conceptual Modeling Techniques. Heidelberg: Springer-Verlag, 2003. 13–25.
- [17] Combi C, Pozzi G. Architectures for a temporal workflow management system. In: Haddad H, *et al.*, eds. Proc. of the ACM Symp. on Applied Computing. Nicosia: ACM, 2004. 659–666.
- [18] Combi C, Pozzi G. Task scheduling for a temporal workflow management system. In: Pustejovsky J, Revesz P, eds. Proc. of the 13th Int'l Symp. on Temporal Representation and Reasoning. Budapest: IEEE Computer Society, 2006. 61–68.
- [19] van der Aalst W, van Hee K. Workflow Management: Models, Methods and System. The MIT Press, 2002. 1–291.
- [20] Yuan CY. The Principle and Application of Petri-Net. Beijing: Electronics Industry Press, 2005. 225–258 (in Chinese).
- [21] Yu Y. Research on temporal workflow model [Ph.D. Thesis]. Guangzhou: Sun Yat-Sen University, 2007 (in Chinese).
- [22] Marjanovic O. Dynamic verification of temporal constraints in production workflows. In: Orlowska ME, ed. Proc. of the 11th Australian Database Conf. Canberra: IEEE Computer Society, 2000. 74–81.

附中文参考文献:

- [3] 余阳,汤庸,梁路,潘炎,汤娜.一个工作流元模型的分析及其时态扩展.小型微型计算机,2006,27(8):1526–1530.
- [5] 杜柱柱,谭建荣,陆国栋.一个基于 TWF-net 的扩展时间工作流模型及其应用.计算机研究与发展,2003,40(4):524–530.

- [9] 李炜,曾广周,王晓琳.一种基于时间 Petri 网的工作流模型.软件学报,2002,13(8):1666-1671. http://www.jos.org.cn/ch/reader/view_abstract.aspx?flag=1&file_no=20020845&journal_id=jos
- [10] 周建涛,史美林,叶新铭.一种基于 Petri 网化简的工作流过程语义验证方法.软件学报,2005,16(7):1242-1251. <http://www.jos.org.cn/1000-9825/16/1242.htm>
- [12] 李建强,范玉顺.一种工作流模型的性能分析方法.计算机学报,2003,26(5):513-523.
- [20] 袁崇义.Petri 网原理与应用.北京:电子工业出版社,2005.225-258.
- [21] 余阳.时态工作流模型研究[博士学位论文].广州:中山大学,2007.



余阳(1966 -),男,江西奉新人,博士,副教授,CCF 高级会员,主要研究领域为工作流,服务计算,软件工程.



郑婷婷(1984 -),女,硕士生,主要研究领域为时态工作流.



汤庸(1964 -),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库与协同软件.



麦健彬(1984 -),男,硕士生,主要研究领域为分布式工作流技术,SOA.



潘茂林(1965 -),男,高级讲师,主要研究领域为智能计算,工作流.