工作流合理性验证中的事件平衡分析*

罗海滨, 范玉顺, 吴 澄

(清华大学 自动化系,北京 100084)

E-mail: luohb97@mails.tsinghua.edu.cn; fan@cims.tsinghua.edu.cn; wuc@tsinghua.edu.cn

http://www.simflow.net

摘要: 工作流合理性的验证是目前工作流研究领域尚未得到很好解决的一个问题,许多工作流管理系统都缺乏有效的方法和工具来保证工作流的合理性.从事件平衡的角度推导了合理工作流所具备的必要条件,提出工作流执行历史的事件平衡定理,并把事件平衡的计算引入工作流模型,用以分析模型中是否存在可能破坏工作流合理性的结构.这种分析方法不受具体工作流模型的限制,适用范围广,而且具有多项式时间的计算复杂度.

关 键 词: 工作流;工作流合理性;事件平衡

中图法分类号: TP311 文献标识码: A

作为实现企业经营过程自动化的一种有效手段,工作流技术自 20 世纪 80 年代产生以来,在各方面都已有了长足的进步和发展,包括工作流管理系统体系结构、工作流模型与定义语言、工作流实现技术、工作流仿真与分析、工作流事务特性、工作流的集成与互操作技术等在内的较为完整的工作流研究框架已经建立[1],而且可用的工作流管理系统已经超过 250 种[2].但是,目前这些工作流管理系统普遍存在的一个不足是缺乏有效的方法与工具来保证工作流的合理性.许多工作流管理系统允许建立存在错误控制逻辑的工作流模型,所建立的工作流模型在执行时是否会出现异常情况也无法保证.导致这种不足的原因主要有以下两个方面:

(1) 工作流本身缺乏坚实的理论基础,许多模型的建立是基于直观理解,没有严格的形式化定义与约束,因而在验证问题上也就没有成熟的理论可以依据;而且,由于不同的工作流系统之间可能存在很大差别,所以,很难有一种统一的验证方法可以适用于所有类型的工作流模型.

(2) 工作流本身的复杂性使得对于它的验证问题变得难以解决.在有关工作流性质验证的可计算性与复杂性的研究中[2~4],人们得出的一个普遍认同的结论是:如果不对模型结构加以限制,则对于任意工作流的性质验证问题将会是 NP 完全甚至是不可判定的.文献[3]指出,在工作流模型的表达能力与该工作流模型合理性验证问题的复杂性之间存在着一种平衡(trade-off),模型的表达能力越强,相应的合理性验证问题就越复杂,一些验证问题是难解的,甚至是不可确定的.

对于已有的一些模型,研究人员给出了相应的验证方法,如文献[2,5]分别提出了基于 Petri 网与 CTR(concurrent transaction logic)的分析方法,文献[3]则通过一些限制提出了一种可在多项式时间内进行终止验证的模型结构.这些方法都是基于具体的工作流模型,而且这些模型都不同程度地带有表达上的局限性,通常是对实际流程的一种简化,因而适用的场合很有限.本文将提出一种与模型无关的工作流合理性的分析方法——事件平衡分析方法,这种方法从工作流执行历史的角度出发,基于工作流对象发生的前因与后果,提出工作流合理性的必要条件——事件平衡定理,并给出证明.进一步地,将事件平衡分析方法引入工作流模型,用于

* 收稿日期: 2001-03-26; 修改日期: 2001-07-30

基金项目: 国家 863 高科技发展计划资助项目(863-511-944-002)

作者简介: 罗海滨(1974－),男,黑龙江佳木斯人,博士生,主要研究领域为工作流技术,企业建模;范玉顺(1962－),男,江苏扬州人,博士,教授,博士生导师,主要研究领域为工作流技术,企业建模,集成平台,Petri 网,车间管理与控制技术;吴澄(1940－),男,浙江桐乡人,教授,博士生导师,中国工程院院士,主要研究领域复杂生产系统的调度与可靠性研究,供应链建模,系统集成.

对模型结构进行判断与分析.

1 基于执行历史的工作流合理性定义

工作流执行历史是指一个工作流实例从开始到结束所经历的轨迹.无论工作流是采用怎样的模型来进行描述,工作流执行历史都具有相同的结构,即一系列对等工作流对象的偏序集合.“对等”是指集合中的元素在概念上属于同一类型;“偏序”是指集合中的不同元素按照一定的顺序进行排列,在这里是基于时间的先后;“工作流对象”则取决于具体的工作流模型,它们可能是事件、任务、操作等,但无论是何种对象,都不影响我们在下文中分析所得的结果,因此,我们以“事件”为代表来进行讨论.

我们称一个工作流是合理的,当且仅当它所有可能的执行历史 H 都满足:

(1) 在 H 中仅有一个工作流的初始事件和一个工作流的结束事件;

(2) 工作流的初始事件最先发生,即 $e \in IE_{wf}, e \notin H(t) \Rightarrow \forall t' \le t, h(t') = \varnothing$;

(3) 工作流的结束事件最后发生,即 $e \in TE_{wf}, e \in H(t) \Rightarrow \forall t' > t, H(t') = H(t)$.

直观地理解,合理的工作流要求它的每一次执行都是以整个工作流的一个初始事件为开始,以整个工作流的一个结束事件为终止.在这个初始事件发生前,任何其他事件都不允许发生,即所有的活动、任务或者操作都处于未执行状态;在这个结束事件发生后,任何事件都不可能发生,即不存在正处于进行中的活动、任务或者操作.

与其他文献中提出的有关工作流合理性方面的要求有所不同,我们所提出的工作流的合理性是从执行历史的角度来考虑的,而并未对模型本身的结构及组成有任何明确的限制.只要一个工作流模型的所有可能的执行历史都满足上述的要求,那么我们并不在意它具有怎样的模型结构,这给了模型结构以最大的自由度.从执行历史的角度来定义工作流的合理性要比从模型本身的角度来定义更为宽松一些.比如,工作流模型中可能存在有不可能发生的事件或任务,但只要它不影响工作流的正常开始与结束,我们仍称它是合理的,只不过该模型在结构上有些冗余,可以进行一些优化.

基于执行历史来定义工作流合理性的另一个优点就在于,工作流执行历史中工作流对象之间的关系具有简单、直接的特点,这种关系是一种按时间偏序的“因果”关系,即某一对象出现在执行历史中的原因就在于执行历史中其他对象在此之前的发生;而且偏序集合中的所有元素都是惟一的、不重复的,即使是在模型中被表示为反复执行的相同元素,在执行历史中也被认为是不同元素而区别对待;出现在执行历史中的元素表示它已经发生,这与模型中定义了某些可能因不满足条件而不一定总发生的元素的情形相比要更为简单,只要是出现在执行历史中的元素,则一定是满足条件而发生的.执行历史所具备的这些特点为我们在下文中推导事件平衡定理奠定了基础.

2 事件平衡定理

定义 **1**. 工作流执行历史中事件的关系是一种按时间偏序的“因果”关系,我们定义这种关系如下:

$\rightarrow: EV \times EV$,$EV$ 为工作流执行历史 H 中的事件集合.$e1 \rightarrow e2$ 表示如果 $e1$ 不出现在 H 中,则 $e2$ 也必然不会发生;换言之,$e1$ 是 $e2$ 发生的前提.

容易证明,$\rightarrow$ 关系本身是一个传递闭包.

如果 $e1 \rightarrow e2$,且不存在 $e' \in EV$ 有 $e1 \rightarrow e', e' \rightarrow e2$ 同时成立,我们则称 $e1$ 与 $e2$ 构成“直接因果”关系,$e1$ 为 $e2$ 的“前因事件”,$e2$ 为 $e1$ 的“后果事件”.在工作流的执行历史中,一个事件可能会有多个前因事件,即该事件需要在这些前因事件都发生之后才可以发生,比如模型中定义的事件同步;同样,一个事件也可以有多个后果事件,即该事件的发生将导致多个后果事件的发生,比如模型定义中的事件并发.

对于一个合理的工作流,由前面的定义可知:工作流的初始事件没有前因事件,而工作流的结束事件也没有后果事件.除初始事件外,其他任意事件至少具有一个前因事件;除结束事件外,其他任意事件都可能具有后果事件;当结束事件发生后,则所有事件都不可能存在后果事件.

对于执行历史中的一个事件而言,它的前因事件数与后果事件数之间没有确定的规律可循,前因事件数小于、等于或者大于后果事件数的情况都可能会发生,但对于执行历史中所有事件而言,这两个数字之间则存在着一定的规律,这就是“事件平衡定理”.

定理 **1**(事件平衡定理). 对于一个合理的工作流,在它的执行历史中,所有事件的前因事件数之和必然等于所有事件的后果事件数之和.也就是说,对于合理的工作流执行历史中的每个事件 ei,i=1,2,…,n,如果它的前因事件数为 N_i,后果事件数为 M_i,则称$(-N_i+M_i)$为 ei 的“事件平衡数”,记为 B_i,则有

$$\sum_{i=1}^{n} B_i = 0.$$

证明:我们由工作流执行历史构造一个有向图 $G(V,E)$,其中 V 为所有节点的集合,E 为连接于节点间的有向弧.构造方法如下:

(1) 把工作流执行历史中的事件按照发生的先后顺序由左向右依次排列,如果是同时发生,则在同一位置上下排列,使任意左侧的事件都早于右侧事件发生.排列完毕后,把每个事件映射成有向图 G 中的节点 v,从而构成节点集合 V.

(2) 在有“直接因果”关系的两个节点(事件)之间,连接一个有向弧,方向是由左至右,也就是由前因事件到后果事件.这条连接弧直观地表示出左侧先发生的事件是右侧后发生事件的“前因”,而右侧后发生的事件是左侧先发生事件的“后果”.如果某一事件有多个后果事件,则会有多条有向弧从这一节点出发(即输出弧),分别指向各个后果事件;如果某一事件有多个前因事件,则会有多条有向弧指向这个节点(即输入弧),这些有向弧分别来自那些需要先发生的前因事件.

(3) 由于存在着工作流执行历史中事件的惟一性、有序性,则按照前面两个步骤构造的有向图 G 是一个无自环、无平行边的简单图,而且无回路.所有的有向弧都是连接于 G 的不同节点间,我们用$|E|$表示图 G 中有向弧的数量.

令 $N=\sum_{i=1}^{n} N_i$,$M=\sum_{i=1}^{n} M_i$,只需证明 $N=M$.

因为 N_i 对应于有向图 G 是图中每一节点的输入弧的数量,所以 N 就是所有节点的输入弧的总和,由上面的构造过程,有 $N=|E|$.

同理,因为 M_i 对应于有向图 G 是图中每一节点的输出弧的数量,M 就是所有节点的输出弧的总和,由上面的构造过程,有 $M=|E|$;亦即 $N=M$. □

图 1 给出了一个有向图 G 的示例,由图 1 可知,$e1$ 为工作流的初始事件,$e7$ 为工作流的结束事件.

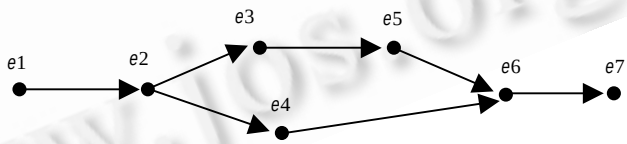

Fig.1 An example of directed graph G

图 1 一个有向图 G 的示例

3 工作流模型的事件平衡分析

事件平衡定理可以看成是工作流合理性的一个必要条件,对于一个合理的工作流而言,它所有可能的执行历史必然满足事件平衡定理;如果一个工作流的执行历史不满足事件平衡定理,则该工作流必然为不合理的.从本质上讲,工作流的执行历史是由工作流模型中的控制流所决定的,一个工作流模型的控制流部分就是所有可能发生的该工作流执行历史的集合,因此,我们把相应的事件平衡数的计算与分析引入到工作流模型中,从而在执行前得出该工作流在事件平衡方面的有关结论,从模型层上保证合理工作流的事件平衡特性.

但是,模型毕竟不能等同于执行历史.由于条件的设置,使得模型中引入了不确定性因素,条件是否满足完全取决于执行时的具体情况.也正是因为有这种不确定性因素的存在,才使模型集合了所有可能发生的执行历

史.因此,在模型层对事件的“前因”、“后果”的计算分析要比已经处于执行历史中的事件的分析困难得多.本节将给出一种基于条件发生概率来计算事件平衡数的方法.首先我们假设模型中不存在任何条件,即在完全确定的情况下分析事件平衡数;然后再引入条件,分析不确定状态下事件平衡数的计算.

(1) 模型中不出现任何条件

在这种情况下,一个事件的后果事件数即为模型中它的后继事件的总和,后继事件包括由它直接产生的事件以及在与其他事件同步后而产生的事件.

一个事件的前因事件数的计算要根据它与前趋事件的关系而定:如果任何一个前趋事件的发生都将直接产生该事件,我们称这种关系为“异步激活”,则前因事件数只计为 1;如果需要所有前趋事件同步后该事件才可发生,我们称这种关系为“同步激活”,则前因事件数为所有需要同步的前趋事件数之和 n.如果一个事件的前趋事件中既有异步关系又有同步关系,我们则可以通过引入中间的辅助事件来把这种复杂关系等价转化为所有事件仅包含一种关系的情形,进而再计算事件平衡数.

(2) 模型中出现条件

在这种情况下,一个事件的后果事件数即为模型中它可能产生的所有后继事件之和,不论后继事件的发生是否有条件,我们都把这些后继事件计入该事件的后果事件数之中,这与在模型中不出现任何条件的情况下计算后果事件数的方法是完全相同的.

对于前因事件数的计算,也需要按照与前趋事件的“同步”与“异步”关系而有所区别,我们先假设该事件有 n 个前趋事件$\{e1,e2,\ldots,en\}$,每个前趋事件与该事件之间的条件为 $C_i,i=1,2,\ldots,n;C_i$ 发生的概率为 $P(C_i)$.如果某一前趋事件 ei 与该事件间没有条件设置,则可以认为 C_i=TRUE,$P(C_i)=1$.

如果该事件与前趋事件之间是“异步”关系,则前因事件数$=1+\sum_{i=1}^{n}P(\neg C_i)$;

如果该事件与前趋事件之间是“同步”关系,则前因事件数$=n+P(C\bigcup_{i=1}^{n}\neg C_i)$;

$\neg C_i$ 表示条件 C_i 的互斥条件,$P(C_i\cap\neg C_i)=0,P(C_i\cup\neg C_i)=1$.

我们可以这样理解上文的两个计算公式:对于“异步”关系来说,如果一个前趋事件的条件不满足时,则 $P(\neg C_i)=1$,表示这个前趋事件的发生不会产生该事件,对于整个工作流的执行而言,相当于多消耗了一个事件,因此前因事件数应该在原有基础上加上 $P(\neg C_i)$;而对于“同步”关系来说,只要有一个前趋事件的条件不满足,$P(\cup\neg C_i)=1$,则该事件必然不能发生,对于整个工作流的执行而言,也相当于多消耗了一个事件,因此前因事件数应该在原有基础上加上 $P(\cup\neg C_i)$.当条件全部满足时,则不论是“异步”关系还是“同步”关系,$P(\neg C_i)=0,P(\cup\neg C_i)=0$,所得的前因事件数与无条件情况下的结果完全相同.

实际上,孤立地计算一个事件的事件平衡数并没有太大的意义,事件平衡定理是在多个事件的事件平衡数之间建立起一种相对的平衡关系,只要满足这种关系,我们并不需要关心每一个具体的事件平衡数的含义.正如在模型中有条件设置的情况下,前因事件数的计算结果可能由于概率的出现而不是整数,但这并不影响事件平衡关系的建立.下面我们将给出两个例子,分别计算在互斥条件与非互斥条件的设置下所得到的事件平衡结果,从而说明条件设置得是否合理将对工作流的合理性产生影响.

例 1:条件选择,如图 2(a)所示.$e0$ 发生后,如果满足条件 C_1,则产生 $e1$;如果满足条件 C_2,则产生 $e2$.$e1,e2$ 同后继事件 $e3$ 之间是异步关系,即只要 $e1,e2$ 有一个发生,则 $e3$ 发生.这是一个工作流的局部片段,$e0$ 到 $e3$ 的事件平衡数分别计算如下:

$$B_0=(-1)+2=1;B_1=-(1+P(\neg C_1))+1;B_2=-(1+P(\neg C_2))+1;B_3=-1+1=0;$$

$$B=B_0+B_1+B_2+B_3=1-(P(\neg C_1)+P(\neg C_2)).$$

如果 C_1,C_2 是互斥条件,则 $P(\neg C_1)+P(\neg C_2)=1$,所以有 $B=0$,事件平衡.

如果 C_1,C_2 是非互斥条件,则 $P(\neg C_1)+P(\neg C_2)$不确定,所以有 B 不一定为 0,事件平衡有可能被破坏,从而导致工作流不合理.直观地理解,非互斥条件下,C_1,C_2 有可能都不满足,则 $e0$ 无法产生后继事件;反之,若 C_1,C_2 都得到满足,则 $e1,e2$ 将分别产生两个 $e3$ 的实例,这也可能导致工作流运行出错.

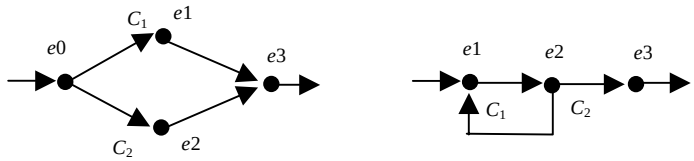

Fig.2 Examples of conditional selection and iteration

图 2 条件选择与反复的工作流示例

例 2:反复(即自环),如图 2(b)所示.当 $e2$ 发生后,如果条件 C_1 满足,则 $e1$ 被再次激活、反复执行;如果条件 C_2 满足,则 $e3$ 被激活,自环结束.$e1$ 到 $e3$ 的事件平衡数计算如下:

$$B_1= -(1+P(\neg C_1))+1;B_2=-1+2=1;B_3= -(1+P(\neg C_2))+1;$$

$$B=B_1+B_2+B_3=1-(P(\neg C_1)+P(\neg C_2)).$$

与例 1 类似,如果 C_1,C_2 是互斥条件,则满足事件平衡;否则,事件平衡可能被破坏.正常情况下,对于反复(自环)的模型结构都是设置互斥条件的,以保证自环能够顺利结束并激活后继事件.

4 结 论

工作流性质的验证问题至今还没有很有效的解决方法.本文从一个比较独特的角度推导了合理工作流的必要条件,提出工作流执行历史的事件平衡定理,并把事件平衡的计算引入工作流模型,用以分析模型中是否存在可能破坏工作流合理性的结构.事件平衡分析的方法具有如下优点:

(1) 适用范围广,它不受某一种具体工作流模型的限制.虽然在本文中是以事件为基础进行讨论的,但是,当模型是由活动或者操作等元素组成时,也同样可以进行活动平衡数、操作平衡数的分析.

(2) 算法是可以在多项式时间内完成的,它的计算复杂度与模型中所包含的事件总数呈线性关系.

(3) 既可对模型整体进行分析,也同样可应用于某一个局部的工作流片段,灵活性大.

事件平衡分析的方法也存在着一些缺点,比如事件平衡定理只是工作流合理性的必要条件.在工作流模型中,由于条件设置的多样性,使得进行事件平衡数的计算比较复杂,当条件不是成对地以互斥的形式出现时,我们所给出的计算公式还有一定的局限性,这也是需要进一步解决的问题.

References:

[1] Luo, Hai-bin, Fan, Yu-shun, Wu, Cheng. The overview of workflow technology. Journal of Software, 2000,11(7):899~907 (in Chinese).

[2] van der Aalst, W.M.P. Verification of workflow nets. In: Azéma, P., ed. Application and Theory of Petri Nets 1997. Lecture Notes in Computer Science 1248, Berlin: Springer-Verlag, 1997. 407~426.

[3] Hofstede, A.H.M., Orlowska, M.E., Rajapakse, J. Verification problems in conceptual workflow specifications. In: Thalheim, B., ed. Proceedings of the 15th International Conference on Conceptual Modeling (ER'96). Lecture Notes in Computer Science 1157, Cottbus: Springer-Verlag, 1996. 73~88.

[4] Bonner, A.J. Workflow, transactions and datalog. In: ACM Symposium on Principles of Database Systems (PODS). 1999. 294~305.

[5] Davulcu, H., Kifer, M., Ramakrishnan, C.R., *et al*. Logic based modeling and analysis of workflows. In: ACM Symposium on Principles of Database Systems (PODS). 1998. 25~33.

附中文参考文献:

[1] 罗海滨,范玉顺,吴澄.工作流技术综述.软件学报,2000,11(7):899~907.

Analysis of Event Balance in the Verification of Workflow Soundness*

LUO Hai-bin, FAN Yu-shun, WU Cheng

(*Department of Automation, Tsinghua University, Beijing* 100084, *China*)

E-mail: luohb97@mails.tsinghua.edu.cn; fan@cims.tsinghua.edu.cn; wuc@tsinghua.edu.cn

http://www.simflow.net

Abstract: Verification of workflow soundness is a problem that has not been solved well in the workflow research. Many workflow management systems lack of the effective tolls which can guarantee the correctness of workflow. In this paper, a necessary condition of sound workflow based on the event balance is put forward. Theorem of event balance in the workflow execution history is proved. Also the calculation of event balance is introduced in the workflow model in order to analyze the model structure which may destroy the soundness of workflow. This method is not bound to any specific workflow model so that it can be used in a wide range of models. Also this method can be finished in polynomial time.

Key words: workflow; workflow soundness; event balance

* Received March 26, 2001; accepted July 30, 2001

Supported by the National High Technology Development 863 Program of China under Grant No.863-511-944-002

全国第 4 次程序设计语言发展与教学学术会议

征 文 通 知

全国第 4 次程序设计语言发展与教学学术会议定于 2003 年春季在江苏扬州召开。本次会议由东南大学承办，扬州大学、南京大学、武汉大学等院校协办，现将有关事项通知如下:

一、征文范围

程序设计语言历史、现状与发展,面向对象语言及相关技术,各类建模语言及其设计、实现与应用,面向网络应用的程序设计语言(XML、HTML、PERL 等),其他各种新型程序设计语言(包括逻辑型语言、函数型语言等),程序设计语言分析、评价与比较,程序设计语言语法、语义与语用以及形式化描述技术与方法,并发、并行与实时程序设计语言,软件开发过程中各类描述语言(包括软件体系结构描述语言等),第四代语言与数据库语言,程序设计语言教学、教材与课件,各类写作语言与工具,其他.

二、征文要求

来稿一般不得超过 6000 字，并且未被其它会议、期刊录用或发表。为了便于正式出版论文集，来稿必须附中英文摘要、关键词与主要参考文献，注明作者姓名、工作单位、详细通讯地址（包括电子邮件地址与电话）与作者简介。欢迎电子投稿，来稿不退，请自留底稿。

三、来稿地址

南京 东南大学 计算机科学与工程系 徐宝文 邮编：210096

电话：(025)3793977； E-mail：bwxu@seu.edu.cn

四、重要日期

征文截止日期： 2002 年 10 月 15 日

录用通知发出日期：2002 年 11 月 15 日

修改稿截止日期： 2002 年 12 月 15 日