基于状态空间的工作流模型验证

赵　磊　钱乐秋　赵文耘
(复旦大学计算机与信息技术系,上海　200433)
E-mail:lingyun1979@263.net

摘　要　工作流模型的正确性和可靠性是工作流管理系统健壮性的基础。该文提出一种基于状态空间的工作流模型验证方法,从而确保在构造期间产生有效的工作流模型。

关键词　工作流模式　依赖　断言　状态空间　验证算法

文章编号　1002-8331-(2004)10-0220-03　**文献标识码**　A　**中图分类号**　TP311

State-space Based Verification of Workflow Model

Zhao Lei　Qian Leqiu　Zhao Wenyun
(Dept. of Computer and Information Technology, Fudan Univ., Shanghai 200433)

Abstract: The correctness and reliability of workflow model is the basis of the robustness of the workflow management system.This article indicates a state-space based verification method of workflow model, to ensure producing the effective workflow model in the building time.

Keywords: workflow scheme, dependency, predicate, state-space, verification algorithm

1　引言

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,在各方面都已经有了长足的进步和发展。在各种工作流相关技术中,工作流的有效建模是开发健壮的、合理的大型工作流系统的关键。

W3C组织提出的XPDL语言是描述工作流模型的良好方法。但是由于XML语言没有坚实的数学基础和逻辑基础,使得由其描述的模型的一致性无法得到保证。

现有的工作流建模通常关注于工作流的图形化表示,而很少关心工作流模型的计算机验证的可能性。对于小型的、简单的工作流系统,开发人员可以比较容易地分析工作流活动之间的逻辑相关性,从而确保工作流模型的有效性;但是对于大型的、包含数十个甚至数百个活动的工作流系统,单凭依靠开发人员的人力判断,无法保证工作流是否完备、是否无二义性、是否无死锁等等。

该文提出一种基于状态空间的工作流模型的形式化验证方法。这样,在工作流的构造阶段,就可以对工作流模型进行自动化验证。

2　工作流验证方法现状

由于工作流模型的复杂性,目前还没有有效的算法可以对工作流模型的正确性进行分析。文献[1]提出一种基于图形化简的方法进行工作流模型验证,并归纳了几种化简规则,但这种方法只对工作流模型结构中存在的特殊问题进行了分析与验证,而且这种方法不适用于存在循环结构的工作流模型。

文献[9]中提出了用Petri网进行工作流建模的工作流网,将工作流模型的正确性归纳为工作流网的完整性,并提出了一种基于Petri网的图形化简方法辅助模型验证,但是其提出的化简步骤由于不具备完备性并不能完全验证工作流模型的正确性;而且Petri网对于大多数用户来说显得过于复杂,不容易掌握。

综合上面的方法,现有工作流建模技术,无论是以UML活动图描述,抑或是以Petri网描述,通常是将工作流建模为活动的迁移。这种方式是比较直观的,但在进行工作流验证的时候,会由于缺少行之有效的形式化表示而难以进行。

该文提出的基于状态空间搜索的工作流验证方法,将工作流建模为一种类似于状态图的形式。其基本思想是将工作流看作状态空间中的序列,这样,产生有效的工作流模式的问题就转化为根据条件和约束搜索状态空间的问题。该文后面会给出将工作流模式由传统的活动序列转化为状态序列,并对其加以形式化表示,使其支持机器自动验证的全过程。

3　工作流模式

3.1　工作流模式定义

工作流管理系统(WfMS)从办公自动化工具演化而来。一个工作流过程定义(或者工作流模式)是业务过程的一个形式化模型。它通常由联合起来实现某一业务目标的复合任务和原始活动组合而成。这些任务通常被组织成有向图,弧表示活动之间的控制流。工作流被建模成过程图$G(N,E)$,N是任务集合,E是转换条件的集合。任务被分配给代理(人或软件或两者)用于顺序或并发执行。一个工作流模式被初始化多次,多个实例可以在工作流执行引擎中同时运行。一个工作流模式必须

基金项目:国家863高技术研究发展计划项目基金资助

作者简介:赵磊,男,复旦大学计算机系硕士研究生。研究方向:软件工程,工作流技术。钱乐秋,男,复旦大学计算机系教授,博士生导师。研究方向:面向对象技术,软件复用技术,软件过程工程。赵文耘,男,复旦大学计算机系教授,博士生导师。研究方向:软件工程,构件技术,软件重用技术。

有足够的表达能力来表达业务过程的结构。

一个任务可以是原子的,也可以是复合的。复合任务促进了工作流模式定义的重用。

图 1 展示了复旦大学办公自动化系统中,一个简化的公文流转工作流模式实例。

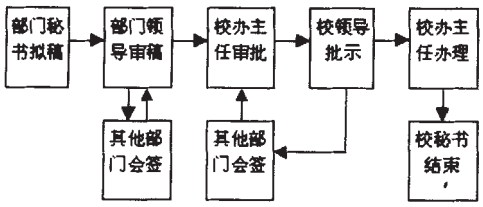

图 1

3.2 依赖与断言

一个工作流应用可以表示为一系列任务(tasks)以及它们之间的依赖关系。一个任务是一个特定的活动。依赖指相连任务之间的关系。不同任务之间可能有两种依赖关系:

(1)通知依赖

通知依赖指前一任务完成后,发送一个通知给后一任务,从而激发后一任务。

(2)数据流依赖

数据流依赖指一个任务需要从另一个任务那里获得某些输入数据。

对这两种依赖分析后,可以看出:对于通知依赖,可以用事件是否发生进行描述;对于数据流依赖,可以用数据是否存在或其是否满足一定条件描述。这样,就可以用统一的断言集对这两种依赖进行描述。一个断言是一个逻辑表达式,用来表示某事为真或为假;一个断言集是一系列逻辑表达式,从而可以用来表示系统中所有事件的真假。这样一个任务可以描述为三元式 T(<I>,a,<O>)。其中,<I>为其输入断言集的集合,表示任务 T 所有可能的输入断言集;同样的,<O>为其输出断言集的集合,表示任务 T 所有可能的输出断言集;a 为活动。其中,一个输入断言集的断言既可以由一个前序任务的某一个输出断言集满足,也可以由多个前序任务的多个输出断言集共同满足。任务的执行由一个输入集的可用性激活。只有第一个可用的输入集会激活任务。

图 2(a)描述了工作流的输入输出断言集表示。从图中可以看出,每个任务可以有多个输入断言集和多个输出断言集;一个输入断言集既可以由单个输出断言集满足(如 $T1/O1$->$Ta/I1$),也可以由多个任务的多个输出断言集满足(如 $T2/O1$,$T3/O1$->$Ta/I2$)。当 $I1$ 或 $I2$ 中的一个满足时,Ta 任务就被激活。

3.3 模型重构

根据分析,在构建工作流模式中,任务中需要关注的是其输入和输出断言集,而具体的活动(或操作)只有在实现的过程中才需要考虑。同时,对于连续的工作流任务,其前驱任务的输出断言与其后续任务的输入断言必然具有某种相关性,即任务之间根据断言集进行匹配。因此,可以考虑将断言集表示成状态的形式,对工作流模型进行重构,即将工作流表示为状态之间的活动转换。

这样,系统中出现的各种断言,即系统中各种变量、依赖等的逻辑表达式,就构成了系统的状态。

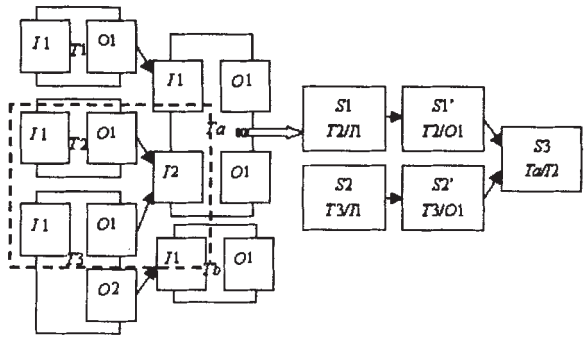

图 2

图 2(b)描述了对图 2(a)虚线框中部分的状态重构:当 $T2$ 任务的 $I1$ 输入断言集满足时,执行 $T2$ 任务,执行完毕转入 $S1'$ 状态;同样的,当 $T3$ 任务的 $I1$ 输入断言集满足时,执行 $T3$ 任务;这里 $T3$ 任务根据执行的结果可能转入 $S2'$ 状态。其后,由 $S1'$,$S2'$ 状态到 $S3$ 状态进行一次迁移为合并的转化,这种合并可以被看作一种伪活动。这样,就完成了一次活动到状态的重构。更复杂的重构方式将在今后的研究过程中详细考察。

4 基于状态空间的工作流表示

该文介绍的验证方法采用了[5]中介绍的设计(Planning)理论的思想,即根据给定的输入条件和一系列运算规则,检测是否可以获得所需的设计。

一个设计需要至少三种输入,也被称作领域描述:

(1)初始状态的描述;

(2)目标即最终状态的描述;

(3)可以被执行的动作的描述。

设计的输出是一个动作序列。这些动作在满足初始状态的条件下执行时,最终可以达到目标。这样,一个设计领域被定义为不同可能状态、可用的动作及由动作执行引起的状态转换的语义模型描述。

这里用 4 元组 P(F,S,A,R)来描述一个语义模型。其中,F 是前面提到的断言的集合,一个断言描述了领域的一个状态。由于变量是有限的,依赖是有限的,因此断言集合必然是有限的。S 是 F 的幂集,它表示状态空间。这样,S 中的元素就是工作流模式某一时刻所处的状态。A 是活动的有限集,它不仅包含前面描述的三元式 T(<I>,a,<O>)中的活动 a,也包括重构过程中需要增加的一些伪活动;R:$S*A$->S 是转换方法。当 R(s,a)不等于空时,动作 a 是可执行的,这里 $a\in A$,$s\in S$。

这样,一个工作流模式生成问题需要考察一个三元组(P,I,T),P 是 4 元组 P(F,S,A,R),I={$s0$}是初始状态,$T\subseteq S$,是目标或终止状态集。对一个问题 WS 的一个设计 plan 被定义成 $plan$={<s,a>:$a\in A$,$s\in S$,<s,a>$\in R$},这里<s,a>被称作状态-活动对。

在基于设计的模型检查方法中,对一个设计的搜索被形式化为状态空间的从终止状态到初始状态的完全搜索。当初始节点通过评估状态标号被定位时,搜索终止。如果初始节点能够到达,一个模式存在,并且该模式等价于从初始节点到终止节点的状态-活动集组成的序列。如果初始节点无法到达,则不存在相应的模式。更进一步的,在一个给定的状态空间中可以存在多个模式。第 5 节将详细描述验证算法。

5 工作流验证

根据前面描述的表示法,可以开发出根据工作流形式化表示进行自动验证的工具。对于给定了断言集的工作流模型,就获得了相应的状态空间。这样,对工作流模型进行验证就转变为根据给定条件求解以获得工作流模式的过程。

为了演示设计算法,以图 1 的公文流转工作流模型的简化版本作为范例。集中于 4 个关键命题来描述状态:P 表示真,NP 表示假,从而产生 2 的 4 次方 =16 个状态的状态空间。同时,由于某些状态之间有显式的顺序约束,实际的状态数可能远小于 16 个,例如对于本范例,只有 6 个状态。

遍历状态空间的设计算法描述如下:

```
function MAKEPLAN (I)
States=T; States´=Φ; Plan=Φ;
While I!⊆State or State!=States´ do
Begin
    States´=States;
    Plan=Plan∪RemovePairs (GetPairs (States),State);
    States=States∪GetStates (GetPairs (States));
end
If I⊆States
then return Plan
else return Fail
```

其中:

GetPairs (*states*)函数表示返回所有可能指向 *states* 状态集的状态-活动对<*s*,*a*>;

RemovePairs{*r*,*states*}函数表示从 *r* 中除去其状态已经在状态集 *states* 中出现的状态-活动对<*s*,*a*>;

GetStates ()函数表示对于给定的一系列状态-活动对,返回相应的状态集。

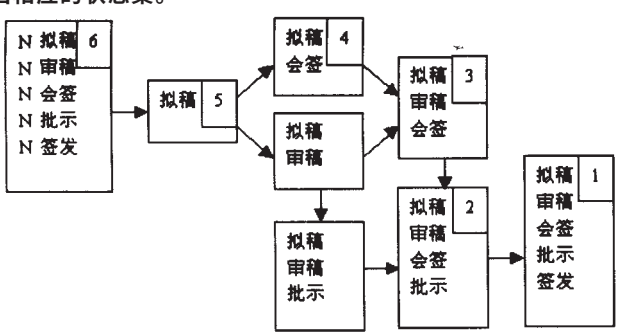

图 3

这里,设计算法从目标状态(即终止状态)开始遍历,直到到达初始节点。各种约束或者属性可以作为断言加入到状态描述中。进一步的,可以使用各种方法对算法进行优化。例如当某个活动序列被禁止时,可以明确剪除不可到达的状态,从而最小化状态空间。此外,这种方法可以被扩展到层次化状态模型。这里任务由子任务组成,状态层次用递归方法遍历。通过层次化的扩展,可以较大地提高状态空间遍历效率。由于状态是有限的,设计总会终止。即如果一个有效的模式存在,它就会被生成。

图 3 描述了复旦大学公文流转工作流模型的一个简化的状态空间。右上角的标号是根据状态空间搜索得到的一个工作流模式。转换弧上的活动未标注。

6 结论

工作流的有效建模是产生健壮的工作流系统的基础。该文提出了一种基于状态空间的工作流的形式化设计方法,同时,给出了对该形式化表示的自动验证方法。这样,可以在工作流设计时期,考察其可用性,从而避免实现期修改模型带来的巨大损失。

今后,研究的重点将放在集成这种形式化表示与验证方法到图形化的工作流建模工具中,通过在构造阶段对活动图输入相应的断言,将活动图自动转化为状态图,并对其进行自动验证,以产生有效的工作流模式;同时,对于大规模的工作流模型,现有的验证算法可能无法予以有效支持。因此,进一步的研究工作也将集中于采用新的验证算法,以保证工作流验证的可行性和高效性。(收稿日期:2004 年 2 月)

参考文献

1.Workflow Management Coalition.The Workflow Reference Model.Web:http://wfmc.org

2.范玉顺.工作流管理技术基础[M].清华大学出版社,2001

3.W M P van der Aslst.The Application of Petri Nets to Workflow Management[J].The Journal of Circuits,Systems and Computers,1998;8(1):21~66

4.C Karamanolis,D Giannakopoulou.Formal Verification of Workflow Schemas.Dept of Computing,Imperial College of Science,Technology and Medicine,2000

5.Frank Leymann,Dieter Roller.Production Workflow[M].Prentice Hall,2000

6.Alessandro Cimatti,Marco Roveri.Confromant planning via symbolic model checking[J].Journal of Artificial Intelligence Research,2002;13

7.Edmund M Clarke Jr,Orna Grumberg,Doron A Peled.Model Checking[M].MIT Press,2000

8.Therani Madhusudan.A Framework for workflow Process Design[R].Technical report,MIS Dept,University of Arizona,Tucson,AZ,2001

9.W M P van der Aslst,Arthur H M Ter Hosfstede.Verification of Workflow Task Structure:A Petri-Net-Based Approach[J].Information Systems,2000,25(1):43~69

(上接 216 页)

多资源约束条件下的任务调度技术在项目管理中具有十分重要的现实意义。该文提出的优化方法和算法,能较好地解决工程项目各任务之间各类资源竞争与工期的矛盾,不仅适合于车辆调度、资源安排、文件流程等,对于生产过程复杂、各项工作连续紧密和一些跨部门、跨企业、跨地区的大型工程项目(如大型建设项目、设备大修、开发新产品等)的进度安排,更具有指导意义。(收稿日期:2003 年 11 月)

参考文献

1.应飚,楼伟进.软件组件技术与知识发现系统[J].计算机工程与设计,2000,21(6)

2.袁小玲,吴业福.组件技术-企业管理信息系统开发的新曙光[J].计算机工程与应用,1999,35(6):56~57

3.徐敏,周定康.组件技术在软件开发中的应用[J].计算机与现代化,2002(1)

4.冯允成.活动网络分析[M].北京航空航天大学出版社

5.陈森发.网络模型及其优化[M].东南大学出版社