

一个支持动态变化的工作流元模型

孙瑞志, 史美林

(清华大学计算机科学与技术系, 北京 100084)

摘要: 支持动态可变和灵活的工作流系统是人们在实际应用中对工作流系统提出的新要求, 传统的工作流建模思想必须加以改进以适应这种可动态变化的工作流运行模式. 本文结合传统工作流的建模规律和动态变化的要求, 提出了一个动态变化的三维空间元模型, 将由过程模型、组织模型和信息模型组成的三维工作流模型从建模和运行变化的角度重新考虑, 增加了对过程、活动、活动属性和组织模型的动态特性描述和操作协议描述, 可以使在这个元模型之上的工作流模型具备人机交互能力和动态灵活性. 元模型在工作流的建立阶段力求将动态特性进行描述和定义, 在执行阶段可以在已定义的动态框架内变化, 既保证了系统的灵活性, 又使系统对动态变化具备可控操作.

关键词: 工作流; 工作流管理系统; 元模型; 灵活性; 动态性

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12A-2052-05

A Meta-Model Supporting Dynamic Changing Workflow

SUN Rui-zhi, SHI Mei-lin

(Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China)

Abstract: The practical applications of workflow management system (WFMS) demand that WFMS should support dynamic changing. To satisfy this request the designing model of traditional workflow must be improved. In this paper, a dynamic three dimensional meta-model is proposed. The meta-model is derived from the traditional laws of building model and the laws of changing. After re-surveying a three dimensional workflow model, which comprises process model, organization model and information model, the meta model describes the dynamic characteristics behaving in objects of process, activity, activity attributes and organization. A workflow model educed from this meta-model will possess the ability of interacting with human and the ability of handling dynamic changing. The meta-model works in the way that it defines the dynamic characteristics as many as possible at the design stage and the changes in run-time can be confined in the pre-defined schema. This approach will enable WFMS to be flexible without losing the process control.

Key words: workflow; workflow management system; meta-model; flexibility; dynamic

1 引言

工作流技术作为面向业务过程的信息处理方式, 适合于业务过程重构和结构化设计, 因而受到了人们的广泛关注. 但随着实际应用的不深入, 人们发现早期的工作流产品在处理业务过程中缺乏灵活性和自适应能力, 尤其是工作流在知识管理系统的应用^[1,2], 需要解决业务运行过程中的人机交互. 传统的工作流, 其工作方式被明确分成两个阶段: 即建立时和运行时. 在建立时, 人们通过不同的建模方法将一个过程定义好, 在运行时, 工作流引擎解释执行定义的过程. 现实生活中, 有的过程在工作流执行阶段才会被确定, 有的活动或过程会发生变化, 两个阶段完全分开的运行模式就不能适应了. 本文在工作流管理联盟 (WFMC) 提出的框架结构之上充分考虑动态变化的因素, 提出了一个动态可变的三维工作流元模型, 可以满足人们对工作流动态变化的需要.

2 相关研究

WFMC 提出了的工作流参考模型及相关的标准^[3,4], 参考模型中给出了一个基本的工作流元模型, 它描述了工作流模型中的基本实体及相互关系. 文献[5]将工作流元模型分成几个角度来看待; 即功能观点、行为观点、信息观点和组织观点. 文献[6]描述了一个三维工作流模型, 即控制流模型、资源模型和实例模型. 控制流模型关心的是任务的执行顺序, 资源模型说明角色和组织, 实例模型关心的是各个执行的实例.

在动态工作流研究方面上, 文献[7]利用外部集成工具, 将外部的群件系统结合到工作流执行中来增加工作流系统的灵活性、开放性和与外部交互的能力. 文献[8,9]通过提供用户原语或命令操作形式, 可以进行人工干预. 文献[10]的反演 (reflection) 方法, 采用工作流执行过程中人机不断交互方式来完成一个可以动态变化的工作流. 文献[11]通过一个包括用

户层、系统层和功能层的 3 层结构,通过各自层次的灵活的模型定义和外部工具,支持动态和自适应能力.文献[12]分析了工作流变化的范围和几种特性,并考虑了变化对时间限制的问题,提出了对修改过程的自动处理问题.

文献[13]提出了一个元模型来进行异常处理,根据 ECA 规则,提出了元类层 (MetaClass)、类层 (Class) 和实例层 (Instance) 来对异常事件实施不同的处理措施.

WFMC 提出的元模型只是说明了构成整个系统的元素以及元素被使用的时机 (建立时和运行时),三维工作流模型^[6]虽然将这些元素的性质进一步分类,但也没有说明如何能够支持动态工作流.文献[13]提出的元模型只用来根据 ECA 规则来实施异常处理.其他文献在分析了动态变化的特性后,提出各种方法,对解决动态变化的工作流都有一定的贡献.但对一个工作流管理系统的设计,需要从体系结构和建模策略等多方面来综合考虑,需要一种更能概括表示动态自适应能力的工作流元模型.

3 动态工作流元模型

一般讲来,工作流在建立阶段被定义,运行阶段被执行.但实际应用中,这两个阶段可能要交替进行.如,一个活动实例被完成时,可能需要动态的选择执行下一个活动的执行者;过程实例化后,可能需要对该过程实例进行修改,插入或删除其中的一个活动实例;过程建模阶段有的活动语义不能明确定义,要等到其他活动有了执行结果才能决定,等等.所以一个过程的执行不能只靠在建立时一次定义就能完成,工作流模型的建立或完善应贯穿在整个工作流系统从设计到运行结束的全过程中.

从工作流的静态构成要素看,是这样一个三维模型,即过程模型、组织模型和信息模型.过程模型就是 WFMC 已经给出的标准模型^[4],组织模型描述了一个组织机构的关系图,信息模型描述了工作流执行阶段的行为以及所访问的数据对象.

工作流系统的动态性是指运行时的三维模型的数据与建立时相比,有的定义发生了变化.这个动态模型如图 1 所示:

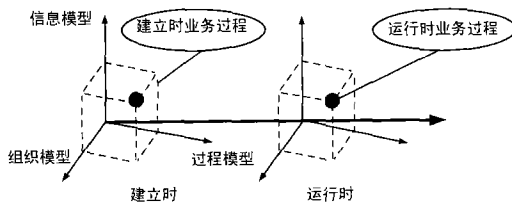


图 1 具有动态特性的工作流三维空间模型

组织模型的动态性主要表现在当工作流处在运行状态时,(1)执行角色发生变化.(2)过程模型的动态性引起组织模型的变化,如修改了的一个工作项要求一个新的具备某些能力的角色或修改某些角色的权限等.

对于工作流中的信息,文献[6]中,作者认为控制数据不能被外部所访问.这样就限制了工作流的动态性,相反,如果

这些在过程定义时被指定的数据,如果在运行时可以被外部访问和修改,工作流管理系统就会表现出灵活性.

过程模型动态性是指实际的业务逻辑可能要变化,或者在运行开始时,还可能有一些不确定的活动存在.

工作流模型定义时需要一种方式来表达这些不确定的因素,并为这些因素的最后确定提供操作工具.人们习惯用元模型来表示从实际应用中抽象出来的形式化表示的对象,所以,动态工作流的元模型可形式化成一个 3 元组 $\langle O, I, P \rangle$,如图 2 所示.

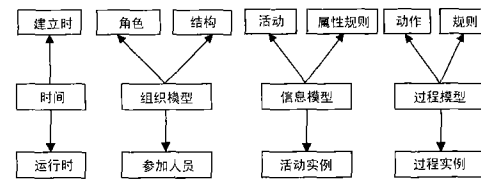


图 2 动态工作流元模型

其中, O 是一个有序集合 $\{(R, S), U\}$,表示组织模型在不同时间的元素,其中 R 是角色的集合, S 是组织结构关系图, U 是运行阶段确定的最终用户的集合;

I 是一个有序集合 $\{(A, C), AI\}$,表示信息模型在不同时间的元素,其中 A 是活动的集合, C 是活动的变化属性集合, AI 是运行阶段确定的活动实例的集合;

P 是一个有序集合 $\{(OP, Ru), PI\}$,表示过程模型在不同时间的元素,其中 OP 是动作的集合, Ru 是操作规则集合, PI 是运行阶段确定的过程实例的集合.

4 工作流运行时的动态交互特性

4.1 工作流系统的知识学习

工作流模型的建立从用户角度来讲希望能提供很大的灵活性^[14],而从技术角度来讲却要求能够有统一的形式化的描述.因此,只有通过人机交互机制和规则的约束,才能满足这两方面的要求.可以将这个交互过程用知识学习的方式来表达,学习的过程可以分成两种类型:(1)知识更新,对原有系统的修改.(2)知识学习:对原有系统的细化.可以表示为: $k^i @ accord(ru) @ action(op) \Rightarrow k^{i+1}$,其中 k^i 表示原有知识,@表示一个实施动作, ru 表示学习规则, $accord(r)$,表示根据 r 来执行, op 是某一动作, $action(op)$ 表示 op 动作的执行, k^{i+1} 表示产生的新知识.

4.2 过程空间的动态元模型

过程模型描述的是活动之间的相互依赖关系和结构,过程是由一般的活动节点 (activity) 和控制节点 (transition) 组成的,工作流系统需要必要的修改原语或操作^[8,9]以便去动态增加或删除活动节点和控制节点或改变过程的运行状态.

考察 3 元组 $\langle O, I, P \rangle$ 中集合 P 的元素 $\{(OP, Ru), PI\}$ 对动态变化的作用,令 $OP = \{insert, delete, set\}$,令所有定义的过程模板构成初始过程定义集合 P_S ,则

过程: $w^i @ action\{insert(f, n, e)\} \Rightarrow w^{i+1}$,表示在过程模板中插入一个对象.其中 $f, n, e \in I$ 是操作的参数,分别表示插

入位置的前一个活动,要插入的活动和之后的活动. $w^i, w^{i+1} \in \{P_S \cup PI\}$,表示对过程模板或者过程实例的修改.同样,可以表示出删除操作和修改操作.

在 4.1 节中,我们看到, workflow 系统中知识的学习由三个要素组成,下面的例子就会看到,以上的操作过程并不能保证 workflow 的正确运行,还需要施加 $accord(ru)$ 操作,用规则来保证正确性.

图 3(a) 是一个简单的过程实例, A2 是我们插入的一个新活动,如果流程已经执行到 A3,则插入后的流程(b)并不能执行 A2.

因此,对于插入操作, workflow 引擎要检查实例当前的运行状态,在上述例子中,当发现插入的操作不在将要执行的路径上时,必须提示用户,是否停止 A3,修改运行状态.所以就可以有如下的规则:

Rule 1: If ($op = \text{"insert"}$) and ($f.state = \text{"finished"}$) then 发提示信息给用户.

对于删除操作,假设在图 3 中,要将过程实例(b)改为(a),则除要判断 A2 是否被执行过外,还要判断 A2 是否正在执行.因此就可以有如下的规则:

```
Rule 2: If ( $op = \text{"delete"}$ )
  Switch  $n.state()$  {
    Case: "finished" 删除并通知用户
    Case: "activated" 发出警告信息并拒绝执行
  }
```

如果操作被拒绝,用户可以通过改变其属性,强制中止活动实例.相应的规则不再赘述.

每一个具体的 workflow 可以有不同的约束规则,我们将 workflow 的变化过程归纳为 5 个子阶段,可以分别定义约束规则.它们是:暂停执行、人工修改、修改后提交、连贯性验证和继续执行.

暂停执行是在 workflow 引擎接到一个操作原语后,做必要的操作以保持一种状态.可以定义规则:

```
Rule 3: Switch  $op$  {
  Case: "insert": 暂停活动  $f$  和  $e$  的实例执行,保存状态;
  Case: "delete": 停止活动  $n$  的实例,保存状态;
  Case: "set": 保存状态;
}
```

修改过程就是系统得到操作原语的输入参数并执行操作命令.可以定义规则来检查输入参数的正确性,保证操作的合法性,例如:

```
Rule 4: Switch  $op$  {
  Case: "insert":  $f$  不能为结束节点,  $e$  不能为开始节点;
  Case: "delete": 活动  $n$  要存在;
  Case: "set": 空操作
```

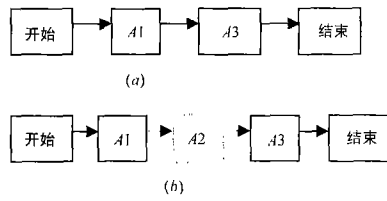


图 3 插入一个新活动

移交,当人工修改完一个活动实例后,就要按照新的过程定义来执行,即将原来的过程升级到新的 workflow 中,这时要考虑:(1)与原 workflow 中的活动实例发生什么依赖关系;(2)修改的活动何时生效.例如,当一个新的活动提交时,可能要撤销已执行的部分活动实例、回退到某个活动节点、跳转到另一个活动开始执行.而修改活动后变成新 workflow 生效的时机可以是立即生效、迟缓生效和可选择,可以定义规则:

Rule 5: If $n.handover_policy = \text{"迟缓生效"}$ then 继续原过程实例;

If $n.handover_policy = \text{"立即生效"}$ then

{ if $n.undo$ then 撤销与活动 n 有关的部分活动实例执行结果;

if $n.compensative$ then 启动与活动 n 有关的补偿操作;

验证,在动态修改时,验证主要指修改后的过程结构和活动行为的正确性.验证规则与验证算法相关,这里不再举例.

可以看到,有序集合 $\{(OP, Ru), PI\}$ 中, OP 和 Ru 中的元素可以动态决定运行阶段的过程实例 PI .

4.3 组织空间的动态元模型

在过程执行时,任务最终要分配给具体的执行者去完成,执行者是通过组织模型来确定的.组织模型的动态性表现在,对一个活动执行者的指定可以(1)指定到角色或指定到用户;(2)实时改变角色或改变用户;(3)限定前后同一活动或两个不同活动的执行者之间的关系;(4)执行者可以进行委托.因此,为达到灵活的用户管理,必须有一定的结构来描述以下几个关系:

隶属关系(belong): 设 $r1, r2 \in R$ 是集合 R 中的 2 个角色,若 $r2 \subset r1$,则称 $r2$ 隶属于 $r1$.同时,若 $\exists u \in U \wedge u \in r1$,则称用户 u 隶属于 $r1$.

同等关系(peer): 指在同一过程或活动中的排他关系和约束关系. 设 $u1, u2 \in U$, 若 $\exists t \in AI \{assign(t, u1) = true\} \Rightarrow \{assign(t, u2) = false\}$, 称 $u1, u2$ 在活动实例 t 上是排他关系. 若 $\exists t \in AI \{assign(t, u1) = true\} \Rightarrow \{assign(after(t), u2) = true\}$, 称 $u1, u2$ 具有约束关系, $assign(t, u)$ 表示任务 t 分配给用户 u 执行, $after(t)$ 表示 t 的后继任务.

委托关系(entrust): 执行者之间的委托关系包括:实时委托,某个任务分配给一个用户后,他可以委托给他人,即 $\exists t \in AI \{assign(t, u1) = true\} \Rightarrow \{assign(t, u2) = true\}$; 全权委托,即所有分配给某用户的任务都委托给他人,即 $\forall t \in AI \{assign(t, u1) = true\} \Rightarrow \{assign(t, u2) = true\}$; 就事委托,将某类任务委托给他人,即 $\{t \in AI \{assign(t, u1) = true\} \Rightarrow \{assign(t, u2) = true\}\}$, $AI' \subset AI$; 时间委托: 在某个时间范围内,委托关系生效.

考察 3 元组 $\langle O, I, P \rangle$ 中的集合 $O = \{(R, S), U\}$ 中元素的操作. 令 R 为所有角色的集合, $S = \{belong, peer, entrust\}$. 则集合 O 的动态性表现在以下两个方面:

(1) 知识学习: 集合 R 和 S 可以被随时修改. 即 $r' @ action$

$(op) \Rightarrow r^{i+1}, s^i @ action(op) \Rightarrow s^{i+1}$, 其中 $r^i, r^{i+1} \in R, s^i, s^{i+1} \in S$.

(2) 实时地确定用户: $r^i @ action\{s\} \Rightarrow r^{i+1}$, 或者: $r^i @ action\{s\} \Rightarrow u, u \in U$. 可以根据角色和关系结构动态确定最终用户或用户组.

所以, 有序集合 $\{(R, S), U\}$ 中 R 和 S 可以动态变化, 变化后的结果决定了任务的最终执行者.

4.4 信息空间的动态元模型

信息空间指工作流运行过程中影响或决定活动行为的控制数据的集合, 这些控制数据在工作流管理系统中表现为活动的属性, 活动和活动属性决定了每个活动实例的执行行为. 因此, 可以从这两个层次来考察活动行为的动态性.

WFMC 定义了以下几种活动节点类型: 起始、结束、原子活动和子过程和块活动等^[2]. 为表达一个事先不能确定或不完整的活动, 必须再定义一种节点类型, 我们将这种类型称之为“黑盒”.

考察 3 元组 $\langle O, I, P \rangle$ 中集合 $I = \{(A, C), AI\}$ 中元素的操作. 令 $a \in A$ 代表一个语义不确定的活动, 则活动 a 可以经过知识学习进行动态演变: $a @ accord(ru) @ action(op) = > AI_s \cup \{b\}$, 其中 $AI_s \subset AI, b \in AI, AI_s$ 是语义明确的活动实例集合; b 是产生的另一个黑盒活动实例, b 可以继续上述演变, 直到该类活动实例不再存在. 图 4 表示了黑盒活动动态演变的一种情况:

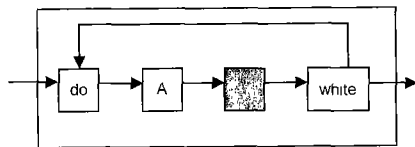


图 4 黑盒活动细化

为使过程中其他的活动不受黑盒内部不确定因素的影响, 可以要求黑盒与其后续活动间的执行次序是一种顺序关系, 图 5 就是不出现的一种情况. 因此, 可以定义规则:

设 $AI_s = \{a_1, a_2, \dots, a_n\}$, 是一个有序集合, a_n 为最后一元素(活动实例), 设 a_n 的后序节点集合 $BI = \{b_1, b_2, \dots\}, AI_s \cap B = \emptyset$ 则:

Rule6: 若 $(b_i \in BI, b_i = after(a_n))$, 则 b_i 是唯一的.

再来考察集合 $I = \{(A, C), AI\}$ 中活动的变化属性集合 C 的作用. 我们将一个活动的属性分为 2 类: 陈述属性和变化属性. 陈述属性指活动的 ID、Name、Performer^[2] 等, 它陈述了一些事实. 变化属性是为动态灵活性而设置的, 用于对活动行为的可改变性加以说明和限制. 可以将活动的执行行为在需要动态修改时划分为 2 类: 维护原来的行为和改变原来的行为. 维护行为包括: 必须执行 (compulsive)、重复 (iterative)、可忽略 (ignorable). 改变行为包括: 可替代 (alternative)、可撤销 (undo)、可取消 (cancelable)、补偿 (compensative) 等. 这种对活动变化时的行为描述为活动的执行提供了灵活的指导. 所以集合 C 可

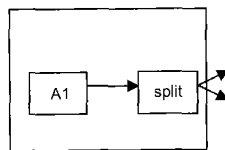


图 5 黑盒活动细化
不合法情况

以取值为 $\{compulsive, iterative, ignorable, alternative, undo, cancelable, compensative\}$.

集合 C 对工作流动态变化的贡献可以表示成 $a @ accord(a.attri) @ action(op) \Rightarrow a_i$, 意思是, 当一个变化操作 op 施加在 A 中的一个活动 a 时, 要根据 a 的变化属性 $a.attri$, 决定 a 的活动实例 a_i . 如 4.2 节中的“移交”操作, 变化属性 $undo, compensative$ 等决定了如 Rule5 中的内容和最后的变化.

所以, 有序集合 $\{(A, C), AI\}$ 中 A 可以表达某些活动的不确定性, A 和 C 决定了活动遇到变化时, 如何操作.

5 与其他相关工作比较及结论

动态工作流元模型既反映了 reflective 的思想, 又包括了文献[6]提出的三维模型, 因为这个三维模型中的实例模型从我们的模型看来就是另外二维模型在时间上的变化. 与其他的工作相比, 元模型更侧重于系统的分析动态变化的因素和处理过程, 而许多相关的研究都集中在从具体的某个方法去解决某些问题^[7,11,12]. 本文研究的贡献在于通过元模型对工作流系统中涉及到的元素的潜在变化, 变化的依据和限制以形式化的方式归纳总结出来, 指导建模过程中的元素定义, 并指导工作流具体实现的条件判断和算法设计.

工作流系统的元模型是开发设计一个工作流管理系统的基础, 对于动态工作流系统, 模型的建立要从变化的角度去重新审视. 本文结合了传统工作流的建模思想和变化的要求, 提出了动态的三维空间元模型, 增加了对元对象(过程、活动、活动属性)动态特性的描述, 通过交互可以做到(1)运行时的系统信息可以被用户访问(2)模型具备动态性, 用户可以修改模型以达到自己的要求.

参考文献:

- [1] Abecker A, Bernardi A, Ntoudis S, et al. The DECOR toolbox for workflow-embedded organizational memory access [A]. Proceedings of 3rd Int. Conference on Enterprise Information Systems [C]. Setbal, Portugal: Kluwer, 2001, 225 - 232.
- [2] Abecker Andreas, Mentzas Gregoris, Legal Maria, et al. Business-process oriented delivery of knowledge through domain ontologies [A]. Proceedings of Second International Workshop on Theory and Applications of Knowledge Management [C]. Munich, 2001. 3 - 7.
- [3] WfMC The Workflow Reference Model TC00-1003, Workflow Management Coalition [S].
- [4] WfMC Process Definition Interchange-Process Model TC-1016-P, Workflow Management Coalition [S].
- [5] Rob van Kaathoven, Manfred A Jeusfeld, Martin Staudt, et al. Organizational memory supported workflow management [A]. Electronic Business Engineering [C]. Heidelberg: Physica-Verlag, 1999. 543 - 563.
- [6] Aalst W M P. van der. Workflow verification: finding control-flow errors using petri-net-based techniques [A]. Business Process Management, Lecture Notes in Computer Science 1806 [C]. USA: 2000. 161 - 183.
- [7] Haake Jorg M, Wang Weigang. Flexible support for business process: extending cooperative hypermedia with process support [J]. Information and Software Technology. USA, 1999. 41: 355 - 366.

- [8] Casati F, Ceri S, Pernici B, Pozzi G. Workflow evolution[J]. Data and Knowledge Engineering, 1998, 24(3): 211 - 238.
- [9] Reichert Manfred, Dadam Peter. ADEPTflex-supporting dynamic changes of workflows without losing control[R]. German: Technical Report No. 1997 - 07, University of Ulm, Germany.
- [10] Edmond David, Hofstede Arthur H M ter. A reflective infrastructure for workflow adaptability[J]. Data & Knowledge Engineering, 2000, 34(3): 271 - 304.
- [11] Kammer Peter J, Bolcer Gerory Alan, Taylor Richard N, et al. Techniques for supporting dynamic and adaptive workflow[J]. Computer supported cooperative work, 2000, 3 - 4: 269 - 292.
- [12] Sadiq Shazia W, Marjanovic Olivera, Orlowska Maria E. Managing change and time in dynamic workflow processes [J]. International Journal of Cooperative Information Systems, 2000, 9(1 - 2): 93 - 116.
- [13] Chiu D K W, Li Qing, Karlapalem kamalakur. A meta modeling approach to workflow management system supporting exception handling [J]. Information system, 1999, 24(2): 159 - 184.
- [14] Jorgensen Havard D, Krogstie John. Active models for dynamic networked organizations[DB/OL]. <http://citeseer.nj.nec.com/488302.html>, 2001.

作者简介:



孙瑞志 男, 1964 年生于山东莱州, 博士生, 副教授, 主要研究领域为计算机网络, 计算机支持的协同工作.



史美林 男, 1938 年生于浙江余姚, 教授, 博士生导师, 主要研究领域为计算机网络, 计算机支持的协同工作.