

# 支持工作流动态变化的过程元模型\*

孙瑞志, 史美林

(清华大学 计算机科学与技术系, 北京 100084)

## A Process Meta-Model Supporting Dynamic Change of Workflow

SUN Rui-Zhi<sup>+</sup>, SHI Mei-Lin

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: 86-10-62785609, E-mail: srzhi@csnet4.cs.tsinghua.edu.cn

<http://www.tsinghua.edu.cn>

Received 2002-02-25; Accepted 2002-05-30

Sun RZ, Shi ML. A process meta-model supporting dynamic change of workflow. *Journal of Software*, 2003,14(1):62-67.

**Abstract:** Supporting dynamic changes and flexibility is required when workflow management systems are employed. Related research work are all focused on some special changing issues. A process meta-model is proposed to guide the design of workflow process model. The process meta-model supports dynamic changes of workflow process. Its dynamic characteristics of workflow process from the points of view of time and process levels are surveyed, and the workflow meta model proposed by workflow management coalition is extended. This process meta-model enables the dynamic characteristics to be described in build-time and enables the changes of a process to be handled with these pre-defined characteristics. Based on this meta-model, a workflow management system possesses the flexibility without losing the control of workflow process.

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

**摘要:** 支持动态可变并具备灵活性是人们在实际应用中对 workflow 管理系统提出的新要求,已有的相关研究工作都集中在对 workflow 中发生变化的某个具体问题上。提出了支持动态特性的 workflow 过程元模型,可以为动态过程模型的设计提供指导。从时间和 workflow 组成的过程级别两个角度分析了其动态特性的表现,扩展了 workflow 管理联盟的 workflow 元模型。扩展后的过程元模型在 workflow 的建立阶段力求将动态特性进行描述和定义,在 workflow 执行阶段可以根据已定义的动态属性,处理过程的变化情况。基于该元模型,workflow 管理系统既具有灵活性,又有利于变化控制操作。

**关键词:** workflow 管理系统; workflow; 元模型; 灵活性; 动态性

中图法分类号: TP311 文献标识码: A

workflow 技术发展到今天,随着实际应用的不断深入,人们对其提出了更多的功能要求,其中一个主要问题是

\* Supported by the National Natural Science Foundation of China under Grant No.60083011 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2001AA113150 (国家高技术研究发展计划)

第一作者简介: 孙瑞志(1964—),男,山东莱州人,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。

workflow 系统的灵活性和动态性.传统的工作流,其工作过程分成两个阶段,即建立阶段和运行阶段.建立阶段是对一个工作流程进行定义,运行阶段是 workflow 执行服务对 workflow 定义进行解释执行.但现实世界是个多变的世界,一个 workflow 在运行阶段不是一成不变的,因此,传统的工作流系统因缺乏柔性(rigid)和实用性(availability),受到了人们的一些批评.如何使 workflow 系统具备灵活的动态性和自适应能力已成为目前 workflow 管理系统研究的一个热点.本文在分析了 workflow 执行过程中动态特性的表现和实质以后,在 WfMC(workflow 管理联盟)定义的工作流元模型基础上,在 workflow 过程控制方面提出了支持动态特性的过程元模型.该元模型对过程的动态变化因素进行了描述,对设计一个具有动态自适应能力的工作流管理系统提供了指导.

## 1 相关研究

在动态自适应 workflow 的研究中,研究者们已经探索了几种不同的技术手段.文献[1]采用异常处理的方法,利用 ECA(event-condition-action)规则,针对出现变化要求时的不同情况采用不同的处理策略.文献[2]采用集成外部工具的方法,通过将外部的群件系统集成到 workflow 执行中,以增加 workflow 系统的灵活性、开放性和与外部交互的能力.文献[3,4]通过提供用户原语或命令操作,使人们可以方便地干预 workflow 的执行.workflow 系统通过原语得到这些操作时,经过一定的算法,努力保证修改后的操作与原来的 workflow 保持连贯.文献[5]中的 Reflection 方法采用 workflow 执行过程中人机不断交互来完成一个具有动态执行的 workflow.文献[6]提出了一种移交策略描述语言,在一个移交策略中定义了一些规则,以保证新旧 workflow 能够转换成功.

以上的各种研究方法从动态性、灵活性等角度出发,采用了各种具体的处理方法,试图使一个 workflow 系统具备动态自适应能力.另一方面,分析 workflow 系统的基本特性——业务逻辑与应用分离——可以看到,如果在 workflow 的过程定义中(业务逻辑)能够充分考虑动态特性,配合执行阶段(应用)的处理,可以从更基本的层次解决动态自适应性问题,这正是本文提出支持动态性的 workflow 过程元模型的出发点.

## 2 过程元模型

### 2.1 工作流过程模型动态特性的表现和实质

可以从时间和 workflow 组成的过程级别这两个方面来分析 workflow 的动态表现.在时间上,我们仍将 workflow 分为建立阶段和运行阶段.workflow 的过程级别是指,是从一个过程的整体流程控制去看问题,还是进一步到组成过程的一个个活动去看问题.如图 1 所示.建立阶段的动态性是指在 workflow 定义时能够定义一个过程的动态行为;运行阶段的动态性是指在一个 workflow 执行时,能够动态修改预先定义的流程或者一些不可预测的变化.过程一级的动态性是指影响整个过程控制的变化,也被称为 workflow 演变;活动一级的动态性是指对一个具体的活动属性进行修改.

Kammer 对动态自适应的 workflow 系统所要完成的功能进行了概括<sup>[7]</sup>,可分为 9 种情况:(1) 能动态地修改和编制 workflow 过程定义;(2) workflow 的执行模型可以配置:进行部分执行、指导性执行和强制执行;(3) 能够为活动、资源、agents 和 artifacts 分门别类地建立模型,保证在运行时可有选择性,并保证整个过程的一致性;(4) 具有自反性:workflow 过程在执行时能够访问自己,并对自己的原模型重新修改;(5) 根据工作实例对过程模型进行演变;(6) 能够逻辑分解一个过程模型;(7) 能够利用过程片断和组件库;(8) 能够访问工作历史,并与专家交互;(9) 为参与者提供相互通信的集成支持.

根据如图 1 所示的二维模型,就可以将上述功能中的(1)、(2)、(4)、(5)的划分为属于过程一级的动态特性,而(3)、(6)、(7)的划分为属于活动一级的动态特性.为了完成这些功能,必须由建立时和运行时相互配合才能完成.其余的几个功能主要涉及运行时进行具体的处理,不涉及 workflow 的动态性.

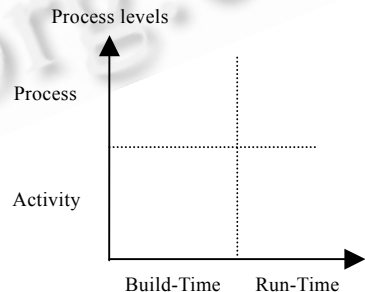


Fig.1 2 views of dynamic process

图 1 观察动态过程的二维空间

### 2.2 工作流过程元模型

工作流过程元模型是指对工作流过程定义中的基本元素和规则进行抽象,并加以一般性描述,用来指导工作

流管理系统的过程建模.过程模型是指如何将实际的运行业务结构化表示成一个工作流程.一个 workflow 系统的灵活性和自适应性在很大程度上取决于过程模型的动态支持能力,所以一套能够描述动态特性的元模型有利于过程模型对动态、灵活性问题的解决.如图 2 所示,为本文提出的一个过程元模型所包含的主要因素.

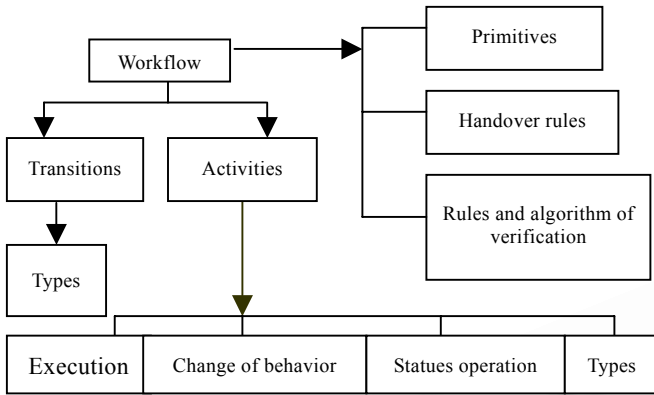


Fig.2 The workflow meta process model  
图 2 工作流元过程模型

支持动态特性的过程模型除了要描述并解决 workflow 中各种活动的具体语义和活动之间的相互依赖关系以外,还要体现动态特性的两个方面:(1) 能够表达在建立阶段有语义不确定的活动或子过程;(2) 允许在运行阶段

修改预先定义的流程.下面,我们将按照如图 1 所示的二维空间来说明过程元模型中的元素如何体现工作流的动态特性.

### 2.3 过程建立阶段对动态性的支持

#### 2.3.1 过程一级的动态支持

建立阶段对过程一级的动态支持表现在对流控变化的描述和组成内容不确定的描述方面.在 WfMC 参考模型中,一个 workflow 由活动(activity)和转换(transition)组成.转换又分成 3 种控制类型,即顺序执行(无条件转换);并行执行(AND);选择执行(OR)<sup>[8]</sup>.在流程的图示中,我们将活动称为一般活动节点,将条件转换称为控制活动节点.在我们的元模型中,将转换扩充成 4 种控制类型,即除了上述 3 种以外,又加上了循环条件控制(do-while, until).循环条件控制的增加,其思想来源于高级程序语言的控制语句类型,因为 workflow 中上一个活动与下一个活动之间的执行关系与高级程序语言的语句之间的执行关系是非常相似的.高级语言中的顺序执行与 workflow 中的顺序执行意义是相同的;workflow 并行执行可以在高级程序语言的并行处理机制中找到原型;选择执行可以对应高级程序语言的条件跳转;循环条件控制可以对应高级程序语言中的循环语句.转换控制类型通过扩充,能够灵活地表达 workflow 中的各种执行顺序,这一点可以通过高级程序语言的控制能力得到间接证明.在图 2 中,转换的分类就用于表示这些控制类型.

workflow 定义的另一类对象——一般活动节点——表示了不同行为的语义.WfMC 定义了以下几类活动节点:起始、结束、原子活动、子过程和块活动等.每一类活动节点的活动语义在建立阶段都是明确定义好的,而 workflow 的动态性则恰恰相反.动态性反映的是结构的不确定性和语义的随机性.为了解决 workflow 的结构化特性和实际要求的动态可变性之间的冲突,我们采用层次化的方法,即在较高的层次上维护其结构性,在具体行为上允许有随机性.对于一个流程来讲,虽然有不确定因素,但整个过程的结构化特性是已知的(否则就不能称作 workflow 了).为了表示不确定的因素,我们的元模型也对 WfMC 参考模型的一般活动节点进行了扩充,定义一个新的节点类型,称为“黑盒”,用来标识一个语义不确定的节点,如图 3 所示.

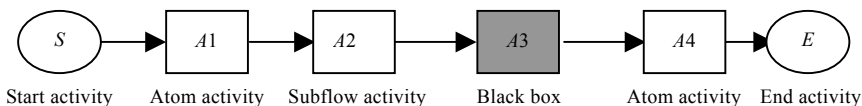


Fig.3 Define a black-box activity in process build-time

图 3 工作流建模时黑盒定义

将“黑盒”节点的语义明确,一般是在流程执行中才能确定,即在 workflow 执行中一些结果逐渐明朗以后,才能

决定黑盒内部的具体意义.这时候,黑盒变成了白盒.白盒的具体意义可以有如下几种方式:空节点、原子活动、子过程、控制节点加活动节点、控制节点加活动节点加黑盒,如图 4 所示.

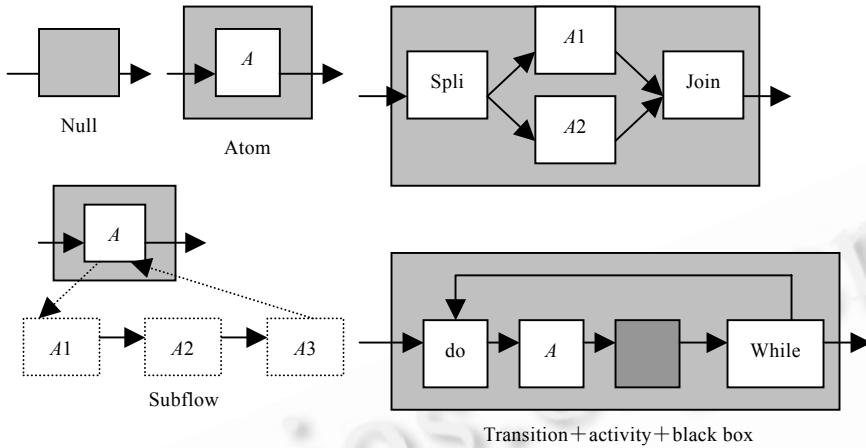


Fig.4 The cases of activities a black box activity denotes  
图 4 黑盒表示的几种活动节点组合

这样,过程元模型在活动节点类型上的扩展与 WfMC 的元模型相比,在工作流定义阶段提高了过程对动态变化的描述能力,为工作流的执行提供了变化的依据.

### 2.3.2 活动一级的动态支持

一个活动的动态性要从它的属性定义来体现.WfMC 虽然对一般活动的属性进行了基本的定义,如 ID,Name,Performer 等,但要支持动态性,就有必要对其进行扩充.在传统的工作流中,一个活动出现在一个流程中意味着当流程到达该活动时,活动就要执行,活动之间分不出哪些是关键的,哪些是可以变通的.为了使整个流程具有灵活性,我们采用以下做法:(1) 先将活动分为如下几个类别:必须的、可选的、可重复的.这一点与文献[1]是相同的.这样,一方面在出现异常情况时,可以根据活动的属性作相应的处理,另一方面使得一个流程的执行不再是死板地执行所定义的每个活动.(2) 通过规则灵活地定义活动的行为,将活动的执行行为划分为维护原来的行为和可以改变原来的行为两类.维护行为又细分为必须执行、可重复执行、可等待执行.改变行为包括可被替代、可被忽略、可以取消、需要人工干预操作等.这种对活动灵活性的描述为一个活动在执行时的动态变化提供了具体的指导.这就是如图 2 所示的活动所具有的“执行属性”和“行为可改变属性”的含义.

## 2.4 过程执行阶段对动态修改的支持

### 2.4.1 workflow运行阶段的动态表现

过程一级和活动一级在工作流运行阶段的动态特性是指,在工作流发生变化以后,是对工作流的整个或部分过程产生影响,还是只对其中一个活动产生影响.我们将工作流执行时的动态处理分为 3 类.(1) 处理在过程建立阶段已定义的有动态行为的活动.(2) 修改原来定义的过程实例,并根据新的过程实例形成新的过程定义.(3) 只修改活动实例而不改变过程定义,这种情况适合于临时改变流程的情况.在这种情况下,一个运行实例执行完以后,不影响原来的过程定义.

因此,运行阶段工作流的动态处理可以划分为两种情况.一种是可预测的处理.例如,前面提到的过程定义中“黑盒”的处理.在此情景下,工作流引擎从过程定义中已知活动的意义是什么,只是具体的操作要由人工干预或由不同的规则处理.另一种是不可预测的操作.例如,一个活动正在执行时或已执行完毕,因为其他活动修改或执行失败导致该活动的重新执行或撤销等.再如,人工干预在何时进行、操作原语何时提交给工作流引擎也是不可预测的.

### 2.4.2 执行阶段动态修改的处理方法与过程

在工作流执行阶段,当需要进行修改时,需要有一系列的操作来完成.例如,插入一个新活动,当这个新的活

动插入以后,为了保证修改以后的 workflow 与已执行的 workflow 实例的连贯性,需要有一个正确性保证机制.再如,“黑盒”节点的具体化,在人工完成具体的活动定义以后,也需要有一个正确性保证机制.这种保证机制需要一定的验证算法.我们把 workflow 的修改过程归纳为 5 个阶段:暂停执行、人工修改、修改后移交、连贯性验证、继续执行.图 5 表示插入一个活动“A3”的操作过程.

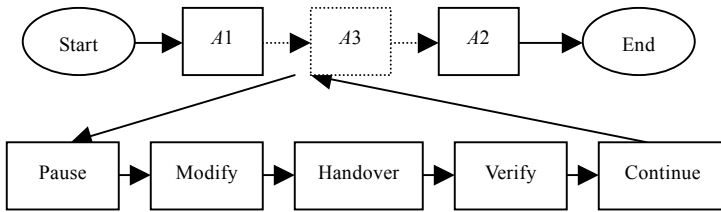


Fig.5 The procedures of a process modification

图 5 流程修改时的处理过程

处理,这就是图 2 中活动节点所指的“状态操作”属性.

修改过程一般是需要人工来操作的.比如,当 workflow 执行到一个黑盒活动或接到一条“插入活动”命令时,workflow 引擎要提供给用户必要的提示,用户可以通过人机交互接口描述一个新活动及其属性,就像建立阶段的过程定义一样,人工可以通过必要的操作命令完成新活动的定义.

移交过程.当人工修改完一个活动或实例以后,就要按照新的流程定义来执行,即将原来的流程升级到新的流程上.修改之后的活动在移交给 workflow 引擎的时候要考虑两方面的问题.(1) 与原 workflow 中的活动实例发生什么依赖关系.当一个新活动移交时,影响原来活动实例的结果可能有:撤销已执行的部分活动实例(undo)、回退到某个活动节点(也许还要补偿)、跳转到另一个活动开始执行<sup>[5]</sup>.(2) 变化之后的活动何时可以生效.活动变化以后就等于生成了新的 workflow,新 workflow 生效的时机可以是立即生效(workflow 引擎立即执行)、迟缓生效(原 workflow 的过程实例执行完毕以后)和选择(根据不同的情况作具体选择)<sup>[9]</sup>.所以,对于每一个移交动作,必须有一个规则来约定.

验证过程.workflow 过程定义的正确性验证是指对过程的结构和语义的正确性进行验证<sup>[6]</sup>,以避免出现死锁、不可达、同步丢失<sup>[10]</sup>及语义不连贯等问题.对于 workflow 的动态修改,验证主要是指修改以后的 workflow 在结构和语义上是否正确.这种验证需要靠一组规则来支持.由于篇幅所限,我们将另文讨论规则的表达.

因此,workflow 在执行阶段进行修改时,必须有一组规则和相应的算法来保证变化后的 workflow 的正确性.这些规则包括移交规则、验证规则等.

### 3 结 论

本文提出的支持动态特性的 workflow 过程元模型,在 WFMC 的元模型基础上,对组成 workflow 的控制活动节点和一般活动节点的类型进行了扩充.在该元模型中,一般活动节点的动态属性从不同的角度去描述一个活动的动态支持能力,一个过程的修改原语提供给用户一种灵活的手段,各种规则是 workflow 在修改变化以后可以继续执行的保证.

一个灵活的支持动态修改的 workflow 系统,虽然在过程执行阶段的随机性和不可预测性使行为的控制变得复杂且不规范,但除了人工干预的操作不可控制以外,workflow 系统本身通过适当的对象描述,能够极大地限制这种随意性.本文提出的 workflow 过程元模型在不影响系统控制权的前提下,能使 workflow 系统具有很好的动态特性和易于实现的管理特性.

在此元模型的指导下,要使一个 workflow 管理系统得到实现,需要制定必要的控制规则,而规则与活动语义有着密切的联系.同时,验证算法的设计和验证能力是保证修改以后 workflow 能够继续执行下去的关键.这些都是我们要进一步解决的问题.

暂停执行是在 workflow 引擎到达一个语义不确定的“黑盒”节点或接到一个修改操作时的一种内部操作,它要做一些必要的操作,以保持一种状态.比如,一个正处在“挂起”状态的活动要被修改,就应有相应的操作来进行状态处理.这些操作必须在过程定义中明确地标明,以便系统可以进行

**References:**

- [1] Chiu DKW, Li Q, Karlapalem K. A meta modeling approach to workflow management system supporting exception handling. *Information System*, 1999,24(2):159~184.
- [2] Haake JM, Wang WG. Flexible support for business process: extending cooperative hypermedia with process support. *Information and Software Technology*, 1999,41(6):355~366.
- [3] Casati F, Ceri S, Pernici B, Pozzi G. Workflow evolution. *Data and Knowledge Engineering*, 1998,24(3):211~238.
- [4] Reichert M, Dadam P. ADEPTflex-Supporting dynamic changes of workflows without losing control. *Journal of Intelligent Information Systems*, 1998,10(2):93~129.
- [5] Edmond D, ter Hofstede AHM. A reflective infrastructure for workflow adaptability. *Data and Knowledge Engineering*, 2000,34(3): 271~304.
- [6] Liu CF, Orłowska ME, Li H. Automating handover in dynamic workflow environments. In: Pernici B, Thanos C, eds. *Advanced Information Systems Engineering*. New York: Springer, 1998. 159~172.
- [7] Kammer PJ. Techniques for supporting dynamic and adaptive workflow. *Computer Supported Cooperative Work*, 2000,9(3-4): 269~292.
- [8] Workflow Management Coalition. Workflow process definition interface-XML process definition language. WFMTC-TC-1025, 2001.
- [9] Ellis C, Keddara K. ML-DEWS: modeling language to support dynamic evolution within workflow system. *Computer Supported Cooperative Work*, 2000,9(3-4):293~333.
- [10] Li HC. On workflow-based CSCW platform [Ph.D. Thesis]. Beijing: Tsinghua University, 2001 (in Chinese with English abstract).

**附中文参考文献:**

- [10] 李红臣.基于工作流的 CSCW 平台研究[博士学位论文].北京:清华大学,2001.