

面向执行的工作流控制流建模技术研究

李伟刚

LI Wei-gang

西北工业大学 软件与微电子学院,西安 710075

College of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710075, China

E-mail: w_g_li@163.com

LI Wei-gang. Research on execution-oriented control flow modeling technique in workflow. Computer Engineering and Applications, 2008, 44(3): 98-100.

Abstract: Control flow modeling in workflow systems commonly focuses on describing the semantics of routing relationship between activities. It constructs process routing structures by composing split nodes and join nodes. But this approach isn't appropriate for workflow engine to execute, and results in complication of the workflow engine. To accord with the nature of control flow, the author investigates the relation between control and response of process and activities firstly, then the routing relationship between activities directly. A control flow modeling approach based on ECA rules is proposed, which can model the dynamic behaviors of workflow by allowing the process and activity respond the events outside the workflow system. Moreover, complicated routing semantics are modeled by links and control points. With this approach, the executing mechanism of control flow is unified. It simplifies the realization of workflow engine greatly.

Key words: workflow; process modeling; control flow; ECA

摘要: 工作流系统中的控制流建模一般注重描述活动间的路由语义,通过组合分支和汇合结点构造过程路由结构。这种描述方法不便于执行,导致工作流引擎的实现复杂。直接从过程和活动本身及相邻两个活动间的控制响应关系和路由关系入手,开发了一种基于ECA规则的控制流建模方法。使得过程和活动能响应外部事件,从而实现对工作流的动态行为建模。而且,用链接和控制点实现了活动间复杂路由语义的建模。统一了工作流引擎对控制流的执行方法,使得工作流引擎的实现机制大大简化。

关键词: 工作流; 过程建模; 控制流; ECA

文章编号: 1002-8331(2008)03-0098-03 **文献标识码:** A **中图分类号:** TP391

1 引言

从工作流执行的角度来看,工作流模型应能描述实际业务过程的功能方面、行为方面、信息方面、操作方面和组织方面^[1]。其中,定义过程中活动的执行依赖关系(控制流)的行为方面是工作流模型中最基本和最重要的方面,是描述其它方面的基础^[2],其描述能力的强弱是评价工作流模型表达能力和灵活性的重要指标。多数工作流模型在定义控制流时,将重点放在过程路由语义的定义上,如XOR分支、AND分支、AND汇合等^[3]。然而,由于控制流本质上描述了活动间的执行依赖关系,所以更自然的描述方法是直接从活动关系(路由关系和控制关系)入手来定义它。这样,工作流引擎在执行某活动时,可按照其与下个活动的链接关系,确定下一步执行路由及它们之间相互作用关系,效率更高。另外,过程路由语义节点难以描述活动间的控制关系,导致工作流引擎处理路由关系和控制关系的方式不统一,增加了其实现难度。本文从易于工作流引擎执行的角度出发,基于ECA规则,开发了一种细粒度、高度灵活的活动间路由关系和控制依赖关系的建模方法,对简化工作流系统的实现

具有指导意义。

2 工作流的控制流模型

2.1 过程与活动

研究工作流建模方法实质上是给出一种元模型,为过程模型创建提供方法论。过程模型是描述工作流的具体方法,其形式化定义如下。

定义1 过程模型。过程模型 P 定义为多元组 $(A, Q, L, V, M, BR, Ac, Sp, c)$,其中 A 是所有活动组成的集合; Q 是起止标记的集合; L 是活动间的执行依赖关系,称为链接; V 是 P 中定义的所有变量的集合; M 约定了 P 运行期间接收的消息格式契约; BR 为商业规则集合; Ac 是 P 上定义的动作集合; Sp 为 P 的实例可能达到的状态的集合; c 是补偿路径。 P 满足以下关系:

- (1) $A \cap Q = \emptyset, A \neq \emptyset$;
- (2) $Q = \{b, f\}$,其中 b 为起始标记, f 为结束标记;
- (3) $L \subset A \times A \cup \{b\} \times A \cup A \times \{f\}$;
- (4) 设 $dom(L) = \{i | \exists j: (i, j) \in L\}$, $ran(L) = \{j | \exists i: (i, j) \in L\}$,

基金项目: 陕西省科技公关计划(the Key Technologies R&D Program of Shaanxi(Province), China under Grant No.2004k05-G45); 西安市信息技术专项(the Special Fund on Information Technology of Xi'an City of China under Grant No.ZX06033)。

作者简介: 李伟刚(1972-),男,博士,主要研究领域为柔性工作流、协同设计、CIMS等。

则 $dom(L) \cup ran(L) = A \cup Q$ 。

过程包含一个起始标记,一个结束标记和若干活动,它们由链接相互关联,并且不存在孤立的活动和起止标记。变量集合 V 描述了过程模型的信息方面,变量是过程与外部数据交换,内部活动之间数据流转的媒介。过程在运行时可以接收其它 workflow 发来的,或由其它外部应用系统发来的消息,由 M 、 BR 和 Ac 组成的 ECA 规则使 workflow 具有响应消息的能力,便于实现分布式过程协调机制,支持跨组织的动态业务流程。

定义 2 活动。活动 $a_i \in A$ 可以定义为一个多元组 $(I, O, Ag, R, M, BR, Ac, S_A, c)$ 。其中 I 和 O 分别是输入和输出数据的集合; Ag 定义了执行 a_i 的代理集合; R 描述了资源及其使用方法; M 约定了 a_i 运行期间接收的消息格式契约; BR 为商业规则集合; Ac 是 a_i 上定义的动作集合; S_A 为活动 a_i 的实例可能达到的状态的集合; c 是补偿路径。

在过程模型 P 中,存在关系: $\bigcup_{i=1}^n (I(a_i) \cup O(a_i)) \subseteq V$, 其中 $n \in N$ 为 P 中的活动数。同样, M, BR, Ac 组合描述了活动对外部事件的响应行为,能实现活动之间或活动与外部应用间细粒度的通信和协调控制。运行时, workflow 引擎根据消息格式 M , 按照 BR 获取消息体中的数据, 顺次执行 Ac 定义的动作。 $Ac \subseteq AC, AC$ 为组件库, 供 workflow 引擎在运行时调用。随着 AC 不断丰富, 过程模型表达能力越来越强, 系统可扩展性越来越高。 $S_A = \{OPEN, CLOSED\}$, OPEN 状态的子状态集合为 $\{NOT_STARTED, SUSPENDED, ASSIGNED, REJECTD, RUNNING\}$; CLOSED 状态的子状态集合为 $\{COMPLETED, ABORTED, TERMINATED, COMPENSATED\}$ 。在定义过程和活动, 可以设定一条补偿路径 c , 它的作用是: 当取消执行中的过程或活动后, 抵消由于不完整执行而带来的数据冲突。另外, 也可以主动地要求消除执行完的过程或活动对数据的影响。

2.2 控制流模型

工作流的控制流描述了 workflow 引擎执行活动的先序后序关系, 即路由关系; 同时, 还应描述活动执行期间相互控制和影响的关系, 即控制关系。通过链接来描述控制流。

定义 3 链接。链接 $l_i \in L$ 表示活动间的执行依赖关系, 定义为六元组 $(ha, ta, e_s, CN_i, Ac, S_L)$, 其中 ha, ta 分别表示 l_i 的头部和尾部, 且 $ha, ta \in A \cup Q$, e_s 是 ha 的一个状态转变事件, 记作 $e(ha.state=s)$, 其中 $s \in S_A$ 为 ha 的状态。 $CN_i \subseteq CN$ 是 l_i 上控制点组成的位置有序的集合, 表示为拟序集 $\langle CN_i, < \rangle$ 。 $Ac \in AC$ 表示当 ha 的状态变成 s 时, 若 CN_i 确定的控制依赖约束规则满足, 在尾部 ta 上进行的元操作(动作)集合。 S_L 为链接能够达到的状态集合, $S_L = \{PASSING, PASSED\}$, 其中, $PASSED = \{TRUE_SIGNALLED, FALSE_SIGNALLED\}$ 。 l_i 记作: $ha \xrightarrow{e_s, CN_i, Ac} ta$ 。称满足 CN_i 确定的条件的链接是通的(passable), l_i 是一个通路。

链接定义的活动间的执行依赖关系隐含路由关系和控制关系。当 $e_s \equiv e(ha.state=COMPLETED) \wedge Ac = \{start\}$ 时, 链接描述了顺次执行的路由关系。链接还可以对活动动态行为的相关关系进行描述: 将一个活动运行状态的变化, 与对其它活动行为的控制关联起来, 这样就提供了一种细粒度、高度灵活的活动间控制依赖关系的建模方法, 模型表达能力提高。 e_s, CN_i 和 Ac 仍然构成 ECA 机制, 由 workflow 引擎统一处理。可见, 过程、活动和链接在关于控制流方面的定义方法是一致的, 这便于 workflow 引擎采用统一的方法执行它们, 同时使得 workflow 引擎的实现难

度大大降低。

链接将分支路由结构语义隐含在控制点集合 CN_i 中, 下面给出控制点的形式化定义。

定义 4 控制点。控制点 cn_i 是链接 l_i 上的一个条件判断点, cn_i 根据语义不同可分为顺序控制点、选择控制点和同步控制点:

(1) 顺序控制点 $cn_i = ev(cn_i)$;

(2) 选择控制点 $cn_i = (ev(cn_i), SEL(p, L_s)), p \in \{1, 2, \dots, |L_s|\} \wedge l_i \in L_s, L_s \in L$;

(3) 同步控制点 $cn_i = (ev(cn_i), SYN(q, L_y)), q \in \{1, 2, \dots, |L_y|\} \wedge l_i \in L_y, L_y \in L$ 。

其中, $ev(cn_i)$ 表示控制点 cn_i 在链接 l_i 上定义的条件表达式, 称为条件分量; $SEL(p, L_s)$ 表示链接集合 L_s 中的任一链接均受 cn_i 的选择语义约束, 即 L_s 中的链接在 cn_i 上有不多于 p 条链接的通路规则为真, $SEL(p, L_s)$ 称为选择控制点 cn_i 在链接 l_i 上的语义分量; $SYN(q, L_y)$ 表示链接集合 L_y 中的任一链接均受 cn_i 的同步语义约束, 即 L_y 中的链接在 cn_i 上有且只有 q 条链接的通路规则同时为真, $SYN(q, L_y)$ 称为同步控制点 cn_i 在链接 l_i 上的语义分量。

由定义 4 可得如下引理:

引理 1 假定 $l_i \in L_s, \exists l_i$ 上的控制点 $cn_i = (ev(cn_i), SEL(p, L_s))$, 则 $\forall l_j \in L_s, j=1, \dots, p, j \neq i, l_j$ 上必然存在选择控制点, 且其语义分量为 $SEL(p, L_s)$; 假定 $l_i \in L_y, \exists l_i$ 上的控制点 $cn_i = (ev(cn_i), SYN(q, L_y))$, 则 $\forall l_j \in L_y, j=1, \dots, q, j \neq i, l_j$ 上必然存在同步控制点, 且其语义分量为 $SYN(q, L_y)$ 。

执行依赖关系的确定关键是要评价控制点所确定的通路规则的值。假设 $CN_i = \{cn_{i1}, cn_{i2}, \dots, cn_{im}\}$, 而 cn_{ij} 定义的通路规则为 $r(cn_{ij})$, 则 $ha \xrightarrow{e_s, CN_i, Ac} ta$ 为通路的充要条件是 $r(cn_{i1}) \wedge r(cn_{i2}) \wedge \dots \wedge r(cn_{im}) = true$ 。

控制点的语义和通路规则参见表 1。

表 1 控制点的语义和通路条件

控制点 cn_i	语义	通路规则 $r(cn_i) = true$
顺序控制点	根据 $ev(cn_i)$ 的值判断 cn_i 在 l_i 上的通路规则是否为真	$ev(cn_i) = true$
选择控制点	根据 $ev(cn_i)$ 的值和 L_s 中具有相同语义分量 $SEL(p, L_s)$ 的选择控制点的条件分量为真的数目 m 判断 cn_i 在 l_i 上的通路规则是否为真	$ev(cn_i) = true \wedge m \leq p$
同步控制点	根据 $ev(cn_i)$ 的值和 L_y 中具有相同语义分量 $SYN(q, L_y)$ 的同步控制点的条件分量为真的数目 m 判断 cn_i 在 l_i 上的通路规则是否为真	$ev(cn_i) = true \wedge m = q$

3 工作流引擎执行方法

本文提出的控制流建模方法是基于 ECA 规则的: 过程上定义的 ECA 规则可实现分布式过程协调, 支持跨组织的动态业务流程; 活动上定义的 ECA 规则可实现活动之间或活动与外部应用间细粒度的通信和协调控制; 链接上定义的 ECA 规则可实现活动间动态行为关联。本文的方法已成功应用于商用 workflow 中间件 Adaptoflow 中, 其中 ECA 的处理机制可用图 1 来示意, 分 3 步进行。

步骤 1 JMS 消息传递到 JMS 服务器的消息队列, JMS 接收者侦听消息到来后, 按消息优先级和过滤规则获取消息并解析消息属性和消息体, 获取数据为引擎所用;

表 2 链接定义

链接的定义	链接上的控制点序列	说明
$l_1=(b, a_1, e(W.state=RUNNING), r(cn_1), init)$	$cn_1="true"$	cn_1 为顺序控制点
$l_2=(a_1, a_2, e(a_1.state=COMPLETED), r(cn_2), init)$	$cn_2="quantity*price>100"$	cn_2 为顺序控制点
$l_3=(a_1, a_3, e(a_1.state=COMPLETED), r(cn_3), init)$	$cn_3="quantity*price\leq 100"$	cn_3 为顺序控制点
$l_4=(a_1, a_5, e(a_1.state=COMPLETED), r(cn_4), init)$	$cn_4="true"$	cn_4 为顺序控制点
$l_5=(a_2, a_4, e(a_2.state=COMPLETED), r(cn_5), init)$	$cn_5="true"$	cn_5 为顺序控制点
$l_6=(a_3, a_4, e(a_3.state=COMPLETED), r(cn_6), init)$	$cn_6="true"$	cn_6 为顺序控制点
$l_7=(a_4, a_6, e(a_4.state=COMPLETED), r(cn_7), init)$	$cn_7=("true", SYN(2, \{l_7, l_8\}))$	cn_7 为同步控制点
$l_8=(a_5, a_6, e(a_5.state=COMPLETED), r(cn_8), init)$	$cn_8=("true", SYN(2, \{l_7, l_8\}))$	cn_8 为同步控制点
$l_9=(a_6, f, e(a_6.state=COMPLETED), r(cn_9), complete)$	$cn_9="true"$	cn_9 为顺序控制点

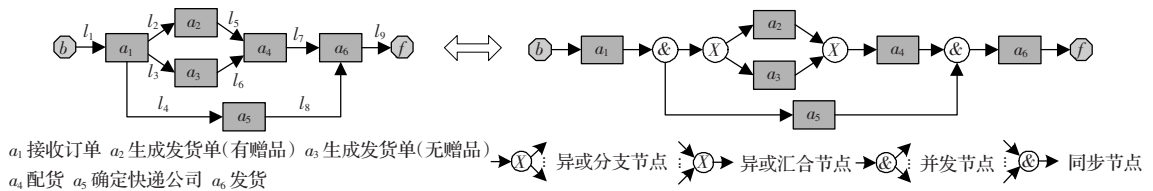


图 2 电子商务订单处理流程

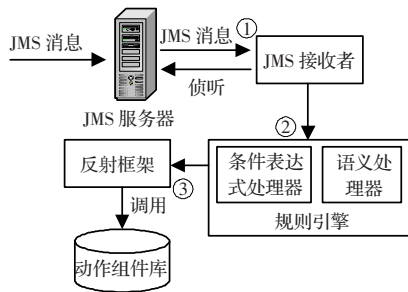


图 1 ECA 处理机制

步骤2 workflow引擎调用规则引擎评价规则和条件,规则引擎可以处理条件表达式和分支路由结构语义。若满足则到步骤3,若不满足则退出;

步骤3 workflow引擎通过反射框架调用动作组件库中的组件,实现相应动作。

JMS 消息分成三部分: 消息头、消息属性和消息体,消息头被 JMS API 使用,与业务逻辑无关; 消息体是 XML 语法描述的业务信息,其格式在 $M(e, \text{是一种系统内部格式})$ 中约定,用 XML Schema 定义; 消息属性的定义十分重要,因为它指出了消息的生产者和消费者、消息接收到后进行的操作、发送给同一消费者的消息的唯一标识等信息。因此消息包含以下属性: $(source, action, targetType, targetID, messageID)$, 这里, $source$ 表示消息生产者。对于过程协调来说,它具有值“MOM”。 $action$ 属性取值“fireWait”、“ERA”或“startProcess”,用于表明消息收到后,激活一个等待活动,执行一个 ERA 规则或启动一个新的过程。 $targetType$ 和 $targetID$ 属性用于标明消息消费者,其值可以指定某个过程实例或活动实例。 $messageID$ 属性用于区分传递给同一个过程实例或活动实例的消息,它的值也是在建模时由通信双方预先约定的。

4 实例研究

通过一个实例说明本文提出的控制流建模方法。如图 2 所示,为一个典型的电子商务流程。过程开始后,人工活动 a_1 接收订单,在电子表单中填写订货信息; 确认订单后,一方面人工活动 a_5 联系快递公司,另一方面自动活动 a_2, a_3 生成发货单,并根据订单总额是否超过 100 元决定是否附送赠品; 然后人工活动 a_4 根据发货单配货, a_6 委托选定的快递公司发货。着重考察

链接的定义方法,详见表 2。

表 2 中,链接 l_7 和 l_8 的执行过程值得注意: 不失一般性,假设 l_7 率先执行,即事件 $e(a_4.state=COMPLETED)$ 先于事件 $e(a_5.state=COMPLETED)$ 发生,则 workflow 引擎评价 cn_7 的条件分量。由于值为真,则需考察同组的链接 l_8 的条件分量。因为此时 l_8 尚未执行,所以 l_7 上的 ECA 失效。直到 l_8 执行时,评价 cn_8 的条件分量为真,同时 cn_7 的条件分量已经为真,所以 l_8 接通,使得只有 l_8 上的 ECA 有效。这就实现了控制同步。

图 2 还给出了使用传统的路由语义节点定义的等价过程模型。可见,以链接为核心定义控制流,虽然描述分支和汇合关系不够直观,但是形式描述更简洁,更易于被 workflow 引擎执行。为了便于用户使用,提供的工作流建模工具仍使用传统路由语义节点的图形方式建模,但是执行时即转换成链接描述形式。

5 结束语

本文从提高控制流表达能力和简化 workflow 引擎的实现出发,以面向执行的观点研究了一种基于 ECA 规则的工作流控制流建模方法,并对 workflow 引擎的执行机制进行了研究。与其它基于 ECA 规则的工作流建模方法^[3-5]相比,本文方法既能描述活动间的执行先序后序关系,又能刻画过程、活动对 workflow 系统内外消息的响应机制,易于采用统一的实现方法。本文研究结果对实现动态 workflow 系统和简化 workflow 引擎具有应用价值。(收稿日期:2007 年 9 月)

参考文献:

- [1] Petkov S, Oren E, Haller A. Aspects in workflow management [R]. Galway: DERI, National University of Ireland, 2005.
- [2] van der Aalst W M P, ter Hofstede A H M, Kiepuszewski B, et al. Workflow patterns[J]. Distributed and Parallel Databases, 2003, 14(1): 5-51.
- [3] 胡锦涛, 张中生, 余新颖. 基于 ECA 规则和活动分解的工作流模型[J]. 软件学报, 2002, 13(4): 761-767.
- [4] Bae J, Bae H, Kang S H, et al. Automatic control of workflow processes using ECA rules[J]. IEEE Trans on Knowledge and Data Engineering, 2004, 16(8): 1010-1023.
- [5] 徐正权, 王治国. 基于 ECA 规则的工作流过程建模[J]. 计算机工程与科学, 2006, 28(5): 105-108.