

基于 π 演算的跨组织 workflow 建模研究

张 静 王海洋 崔立真

(山东大学计算机科学与技术学院 济南 250061)

(gshzhj@public.jn.sd.cn)

Research on Cross-Organizational Workflow Modeling Based on Pi-Calculus

Zhang Jing, Wang Haiyang, and Cui Lizhen

(Institute of Computer Science and Technology, Shandong University, Jinan 250061)

Abstract Cross-organizational workflow has some different characteristics compared with normal workflow, such as process-oriented, compositionality, abstraction, involving communication, and collaboration of several systems. Traditional workflow modeling methods cannot meet these new requirements because they don't have mechanisms to support abstraction, and there are no standards and concurrency operations to obtain bigger models by combining small ones in these modeling methods. Aimed at this problem, a cross-organizational workflow modeling method based on Pi-calculus is presented and the consistency and compositionality of the model are analyzed. Using concurrent operators Pi-calculus provided, a cross-organizational business process is modeled as a composition of a set of autonomy and concurrent intra-organizational sub-processes, and an intra-organizational sub-process is modeled as a composition of a local business process definition and control constraints between organizations as well. Based on weak bisimulation theory of Pi-calculus, external behavior equivalence of two cross-organizational sub-processes is defined, which can help building abstract of an internal private business process. Compared with the traditional methods, this new cross-organizational workflow modeling method builds a loosely coupled relationship among sub-processes, which makes this model adapt to dynamic cross-organizational environment. And this model is based on strict formal method, which facilitates the analysis and verification of business process models.

Key words cross-organizational workflow ; Pi-calculus ; business process modeling ; abstract ; composition

摘 要 传统的工作流建模方法主要用于描述组织内部流程,因而难以描述跨组织 workflow 面向流程、组合、抽象、涉及多个自治系统通信合作的新特点. 针对此问题,提出了一种基于 π 演算的跨组织 workflow 建模方法,利用 π 演算的并发计算操作符,将跨组织业务流程建模为一组自治且并发执行的组织内子流程的组合,子流程建模为组织内本地流程定义和组织间控制约束的组合. 基于 π 演算的弱互相似理论,验证了两个跨组织子流程外部行为的相等性,用于帮助组织内私有流程的外部抽象. 基于该方法建立的跨组织 workflow 模型在子流程间建立了一种松耦合的关系,适用于动态的跨组织环境,同时基于严格的形式化方法,便于分析和验证.

关键词 跨组织 workflow ; π 演算 ; 业务流程建模 ; 抽象 ; 组合

中图法分类号 TP311

随着计算机和网络技术的发展, workflow 技术应用于越来越多的领域,应用的范围也越来越广,逐渐

跨越组织的边界^[1-2]. 与传统的组织内 workflow 技术相比,跨组织 workflow 位于一种更加分布的环境,并涉

及多个动态且自治的实体以及大量的异构信息源, 这为工作流的建模带来了新的挑战: 首先, 跨组织工作流是由不同组织的多个子流程组合而成的一个集成流程, 它具有面向流程的组合特性; 其次, 每个子流程由不同工作流系统管理, 必须维护它们的自治性, 同时这些子流程又需要相互通信, 共同推进全局流程的演进, 即涉及多个自治系统的通信合作; 第3, 在跨组织环境下, 企业希望隐蔽私有流程的内部细节, 因而组成跨组织工作流的子流程往往是实际流程的一个抽象。

面对跨组织工作流呈现的面向流程、组合、抽象、涉及多个自治系统通信合作的新特点, 传统的用于描述组织内部简单、集中流程的工作流建模方法显然存在着不足。针对此问题, 本文以 Pi 演算为形式化基础, 提出一种新的跨组织工作流模型, 模型既描述了跨组织工作流面向流程、组合、涉及多个系统合作并发的特点, 同时可以基于 Pi 演算互模拟理论验证流程的相等性, 将组织内部私有流程抽象为外部公共流程, 保持各组织的私有自治性。

1 相关研究

流程建模是将实际业务流程转化为计算机可处理的形式化定义的过程, 因而一直是 workflow 技术的一个研究热点^[3-4], 目前已提出了多种建模方法, 如有向图、IDEF、Petri 网、UML、角色活动图等^[5-6]。但这些建模方法是针对传统的组织内业务流程提出的, 随着业务流程逐渐跨越组织的边界, 一些研究工作开始致力于扩展传统 workflow 建模方法以支持跨组织环境, 尤其是针对 Petri 网的研究^[7-8], 虽然 Petri 网在建模业务流程中有诸多的优势^[9], 但它不具备支持抽象的机制, 而且欠缺一种合并多个子 Petri 网成为一个大的 Petri 网的标准并发操作^[10], 因而 Petri 网对于跨组织工作流的描述有一定的局限性。显然, 其他的流程建模方法也存在着与 Petri 网相同的上述局限性。

Pi 演算是一种描述并发系统的计算模型, 作为过程代数的一个衍生体, 它不仅具有严格的形式化语义, 而且提供了利用并发操作符, 通过组合子系统构造系统的方法^[11], 非常适用描述越来越复杂的业务流程。目前已有一些相关的研究工作, Förster 在文献^[12]中提出了将 Pi 演算做为流程形式化的一种方法, 解释了它的理论概念以及成为一种形式化业务流程建模语言基础需要满足的条件; Smith 等人在分析了工作流的根本和局限性之后, 提出了工

作流就是 Pi 过程的观点^[13]; 通过建模活动为一个与其他进程通过端口的交互, 文献^[14]提出了利用 Pi 演算建模 workflow 的一种新方法, 该模型可利用 Pi 演算的标记转移语义(LTS)描述 workflow 的动态行为。上述研究工作的重点仍是关注于企业内部的传统 workflow, 并未涉及对跨组织 workflow 的研究。

2 跨组织工作流模型

一个跨组织工作流由位于多个组织内的子流程以及这些子流程之间的约束关系组成。如图 1 所示的订单处理业务流程就是一个跨组织工作流, 它由分属不同组织的子流程 price, ship, account 和 invoice 组成。本文将以此流程为例说明跨组织工作流的建模方法。

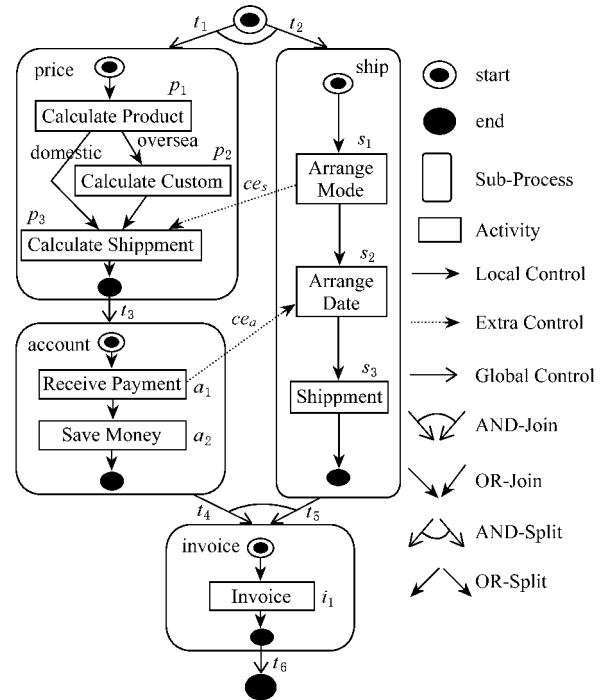


Fig. 1 Order processing business process.

图 1 订单处理业务流程

2.1 模型概述

跨组织工作流中各元素间的关系形成了如图 2 所示的跨组织工作流模型: 组织内部的本地 workflow 活动和流程控制结构组成了一个本地业务流程, 加上建立在其上的外部流程控制规则就构成了一个跨组织子流程。跨组织子流程是构成跨组织工作流的基本单元, 由各自的流程管理系统自治管理, 并由子流程间的输入/输出转移同步。子流程加上它应响应的跨组织输入转移以及输入转移的组合类型、应产生的跨组织输出转移以及输出转移的组合类型就形

成了一个跨组织 workflow 活动. 跨组织 workflow 是多个 跨组织 workflow 活动的一个组合.

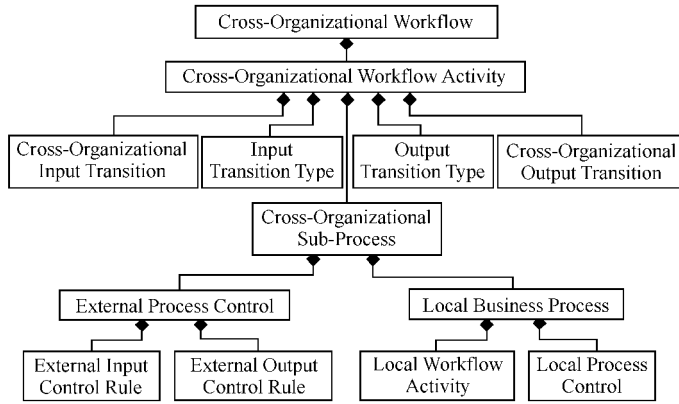


Fig. 2 Elements of cross-organizational workflow model.

图 2 跨组织 workflow 模型元素

2.2 本地业务流程

本地业务流程是形成跨组织 workflow 的基础,它是一个组织内部的本地定义,并由这个组织自治管理. 一个本地业务流程由在本地执行的工作流活动和建立在其上的本地控制结构组成.

定义 1. 本地 workflow 活动 LWA 是一个五元组 (ls, es, le, ee, eq) , 其中 ls, es 分别为接收本地和外部启动消息的通道, le 和 ee 分别为向本地和外部发送结束消息的通道, eq 为发送外部启动请求的通道. 以 ls, es, le, ee, eq 为参数, LWA 的行为定义为:

$$LWA(ls, es, le, ee, eq) \stackrel{\text{def}}{=} (ls. \overline{eq}. es + es. ls). \tau. \overline{le}. \overline{ee}. \quad (1)$$

在跨组织环境下,在组织本地执行的工作流活动必须在收到本地和外部启动消息后方可启动. 活动启动后执行处理任务的操作,这些操作仅仅涉及活动的内部通信和内部状态,因而是一个内部行为 τ . 活动结束后通过通道 le 和 ee 分别通知本地流程控制和外部流程控制本活动的结束.

定义 2. 本地流程控制结构 LPC 定义了 workflow 活动的执行顺序及依赖关系,它是一个嵌套的结构,一个 LPC 由多个子控制结构以及建立在它们之上的控制关系构成. 根据流程活动之间的控制依赖,区分为 4 种本地流程控制结构:顺序、选择、并发、循环.

定义 3. 顺序控制结构 SC 由 n 个顺序执行的本地流程控制结构组成,以启动命令通道 s 、结束消息通道 e 、SC 中包含的所有本地 workflow 活动的外部启动命令通道 ES 、外部结束消息通道 EE 、外部启动请求通道 EQ 为参数, SC 形式化为

$$SC(s, e, ES, EE, EQ) \stackrel{\text{def}}{=} (\bigvee s_2 \dots s_n) \cdot (\bigvee e_1 \dots e_{n-1} \overline{e_1} \cdot \overline{s_2} \cdot e_2 \dots \overline{s_{n-1}} \cdot e_{n-1} \cdot \overline{s_n})$$

$$LPC_1(s, e, ES_1, EE_1, EQ_1) | LPC_2(s_2, e_2, ES_2, EE_2, EQ_2) | \dots | LPC_n(s_n, e, ES_n, EE_n, EQ_n). \quad (2)$$

如果顺序结构 SC 中只包含一个本地 workflow 活动 LWA, 则称这种特殊类型的顺序结构为原子控制结构 AC, 显然它依照本地 workflow 活动的定义行为. 如果顺序结构 SC 中子控制结构的个数为 0, 那么这种结构被称为空结构 NC:

$$AC(s, e, es, ee, eq) \stackrel{\text{def}}{=} LWA(s, es, e, ee, eq), \quad (3)$$

$$NC(s, e) \stackrel{\text{def}}{=} s. \overline{e}. 0. \quad (4)$$

如图 3 中的最外层虚线所包括的结构是一个顺序结构 SC_{price} , 它由 3 个子结构:原子结构 AC_1 、选择结构 CC_{price} 和原子 AC_3 组成, 定义为

$$SC_{\text{price}}(s, e, es_1, es_2, es_3, ee_1, ee_2, ee_3, eq_1, eq_2, eq_3) \stackrel{\text{def}}{=} (\bigvee s_2, s_3 \overline{e_1} \cdot e_2 \overline{e_1} \cdot \overline{s_3} | AC_1(s, e, es_1, ee_1, eq_1) | CC_{\text{price}}(s_2, e_2, es_2, ee_2, eq_2) | AC_3(s_3, e, es_3, ee_3, eq_3)). \quad (5)$$

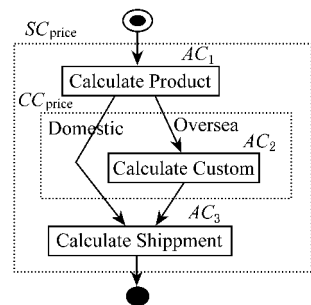


Fig. 3 An example of local process control.

图 3 本地流程控制结构举例

定义 4. 选择控制结构 CC 由 n 个子控制结构组成, 并选择一个子控制结构执行, 形式化为:

$$CC(s, e, ES, EE, EQ) \stackrel{\text{def}}{=} LPC_1(s, e, ES_1, EE_1, EQ_1) + \dots + LPC_n(s, e, ES_n, EE_n, EQ_n). \quad (6)$$

如图 3 中内层虚线所包括的结构是一个选择结构 CC_{price} , 它包括两个顺序结构: 原子结构 AC_2 和空结构, 定义为:

$$CC_{\text{price}}(s, e, es, ee, eq) \stackrel{\text{def}}{=} AC_2(s, e, es, ee, eq) + NC(s, e). \quad (7)$$

定义 5. 并发控制结构 PC 由 n 个子分支控制结构组成, 且并行执行这 n 个子控制结构, 形式化为:

$$PC(s, e, ES, EE, EQ) \stackrel{\text{def}}{=} (\bigvee s_1 \dots s_n) (\bigvee e_1 \dots e_n) \bar{x} s. (\bar{s}_1.e_1 \mid \dots \mid \bar{s}_n.e_n). \bar{e} \mid LPC_1(s_1, e_1, ES_1, EE_1, EQ_1) \mid \dots \mid LPC_n(s_n, e_n, ES_n, EE_n, EQ_n). \quad (8)$$

定义 6. 循环控制结构 LC 循环执行一个流程控制结构, 直至循环条件 c 不再被满足, 形式化为

$$LC(s, e, ES, EE, EQ) \stackrel{\text{def}}{=} (\bigvee s_l, e_l) \bar{x} s. \bar{s}_l.e_l. ([c = \text{true}] \bar{s} \mid LC(s, e, ES, EE, EQ)) + ([c = \text{false}] \bar{e}) \mid LPC(s_l, e_l, ES, EE, EQ). \quad (9)$$

定义 7. 本地业务流程 LBP 是建立在本地 workflow 活动集合 $WAS = \{LWA_1, \dots, LWA_n\}$ 上的一个本地流程控制结构 LPC . 以流程启动和结束命令通道 ps 和 pe , WAS 中所有活动的外部启动和结束消息通道 $ES = \{es_1, \dots, es_n\}$, $EE = \{ee_1, \dots, ee_n\}$ 以及外部启动请求通道 $EQ = \{eq_1, \dots, eq_n\}$ 为参数, LBP 的行为可由如下表达式定义

$$LBP(ps, pe, ES, EE, EQ) \stackrel{\text{def}}{=} LPC(ps, pe, ES, EE, EQ). \quad (10)$$

如图 3 中的本地业务流程 price 是建立在本地 workflow 活动集 $WAS_{\text{price}} = \{LWA_{p1}, LWA_{p2}, LWA_{p3}\}$ 上的一个顺序控制结构, 定义为

$$LBP_{\text{price}}(ps, pe, es_1, es_2, es_3, ee_1, ee_2, ee_3, eq_1, eq_2, eq_3) \stackrel{\text{def}}{=} SC_{\text{price}}(ps, pe, es_1, es_2, es_3, ee_1, ee_2, ee_3, eq_1, eq_2, eq_3). \quad (11)$$

令 $LBPS$ 为所有本地业务流程的集合, 定义本地 workflow 活动 LWA 上的函数 $\text{process}: LWA \rightarrow LBP$, $LBP \in LBPS$ 返回 workflow 活动的所属的本地业务流程.

2.3 外部流程控制

在跨组织 workflow 环境下, 跨组织子流程除了接受本地流程管理的控制外, 还受到其他流程中活动执行情况的约束, 即外部流程控制, 分为输入控制和输出控制两种. 它们分别定义了一个 workflow 活动的启动和结束与其他流程中的活动间的依赖关系.

定义 8. 外部流程输入控制规则 $EPIC$ 是一个二元组 $(pre_cond, post_wa)$:

1) $pre_cond = \{wa_{e1}, \dots, wa_{ep}\}$ 为控制规则的前提活动, 且 $pre_cond \neq \emptyset$;

2) $post_wa$ 为控制规则约束的跨组织 workflow 活动, 且 $\forall wa_{ei} \in pre_cond \Rightarrow process(wa_{ei}) \neq process(post_wa)$.

以 pre_cond 中的所有活动的跨组织结束消息通道 $CE = \{ce_1, \dots, ce_p\}$, $post_wa$ 的外部启动消息通道 es 为参数, $EPIC$ 的行为定义如下表达式:

$$EPIC(CE, es) \stackrel{\text{def}}{=} (ce_1 \mid \dots \mid ce_p). \bar{es}. \quad (12)$$

定义 9. 外部流程输出控制规则 $EPOC$ 是一个三元组 $(pre_wa, post_cond, post_type)$:

1) pre_wa 为输出控制规则约束的本地 workflow 活动;

2) $post_cond = \{wa_{e1}, c_1, \dots, wa_{eq}, c_q\}$ 为控制规则的外部后继活动及条件, $post_cond \neq \emptyset$ 且 $\forall wa_{ei}, c_i \in post_cond \Rightarrow process(wa_{ei}) \neq process(pre_wa)$.

3) $post_type \in \{\text{AND}, \text{OR}\}$ 约束了 q 个外部后继活动的并行关系.

以 pre_wa 的外部结束消息通道 ee , 所有外部活动的跨组织结束消息通道 $CE = \{ce_1, \dots, ce_q\}$ 为参数, $post_type$ 为 AND 和 OR 的 $EPOC$ 的行为分别由表达式(13)和(14)定义:

$$EPOC_{\text{AND}}(ee, CE) \stackrel{\text{def}}{=} ee. ([c_1 = \text{true}] \bar{ce}_1 \mid \dots \mid [c_q = \text{true}] \bar{ce}_q), \quad (13)$$

$$EPOC_{\text{OR}}(ee, CE) \stackrel{\text{def}}{=} ee. ([c_1 = \text{true}] \bar{ce}_1 + \dots + [c_q = \text{true}] \bar{ce}_q). \quad (14)$$

定义 10. 本地业务流程的外部流程控制 EPC 是建立在本地业务流程的 workflow 活动集 WAS 上的 u 个外部流程输入控制规则 $EPICS = \{eqic_1, \dots, eqic_u\}$ 和 v 个输出控制规则 $EPOCS = \{eqoc_1, \dots, eqoc_v\}$ 的集合. 以 WAS 中所有活动的外部启动和结束消息通道 $ES = \{es_1, \dots, es_n\}$, $EE = \{ee_1, \dots, ee_n\}$ 以及外部启动请求通道 $EQ = \{eq_1, \dots, eq_n\}$, $EPICS$ 中所有跨组织结束消息通道 $CE_i = \{ce_{i1}, \dots,$

$ce_{is}\}$, EPOCS 中所有跨组织启动消息通道 $CE_o = \{ce_{o1} \dots, ce_{ot}\}$ 为参数, 外部流程控制 EPC 的行为由表达式 (15)~(19) 定义:

$$EPC(ES, EE, EQ, CE_o, CE_i) \stackrel{\text{def}}{=} EPICS(CE_i, ES_y) | EPOCS(EE_y, CE_o) | ESQC(EQ_y, EQ_n, ES_n) | EENC(EE_n); \quad (15)$$

$$EPICS(CE_i, ES_y) \stackrel{\text{def}}{=} EPIC_1(CE_{i1}, es_{y1}) | \dots | EPIC_u(CE_{iu}, es_{yu}); \quad (16)$$

$$EPOCS(EE_y, CE_o) \stackrel{\text{def}}{=} EPOC_1(ee_{y1}, CE_{o1}) | \dots | EPOC_v(ee_{yv}, CE_{ov}); \quad (17)$$

$$ESQC(EQ_y, EQ_n, ES_n) \stackrel{\text{def}}{=} eq_{y1} | \dots | eq_{yv} | eq_{n1} \cdot \overline{es_{n1}} | \dots | eq_{na} \cdot \overline{es_{na}}; \quad (18)$$

$$EENC(EE_n) \stackrel{\text{def}}{=} ee_{n1} | \dots | ee_{nb}, \quad (19)$$

其中:

$$\begin{aligned} ES_y \cap ES_n &= \emptyset \& ES_y \cup ES_n = ES, \\ EE_y \cap EE_n &= \emptyset \& EE_y \cup EE_n = EE, \\ EQ_y \cap EQ_n &= \emptyset \& EQ_y \cup EQ_n = EQ, \end{aligned}$$

$$\bigcup_{j=0}^u CE_{ij} = CE_j, \quad \bigcup_{j=0}^v CE_{oj} = CE_o.$$

ESQC 定义了本地流程的外部启动请求控制, EENC 定义了对所有无外部输出控制规则活动的结束消息的空操作响应。

如图 1 中的 ship 子流程有一个外部流程输入控制约束和一个外部流程输出控制约束:

$$EPIC_{\text{ship}}(ce_s, es_2) \stackrel{\text{def}}{=} ce_s \cdot \overline{es_2}; \quad (20)$$

$$EPOC_{\text{ship}}(ee_1, ce_a) \stackrel{\text{def}}{=} ee_1 \cdot [cc_1 = \text{true}] \overline{ce_a}. \quad (21)$$

它的外部启动请求控制和外部结束空操作控制分别为:

$$ESQC_{\text{ship}}(eq_2, eq_1, eq_3, es_1, es_3) \stackrel{\text{def}}{=} eq_2 | eq_1 \cdot \overline{es_1} | eq_3 \cdot \overline{es_3}; \quad (22)$$

$$EENC_{\text{ship}}(ee_2, ee_3) \stackrel{\text{def}}{=} ee_2 | ee_3. \quad (23)$$

本地流程 ship 的外部流程控制定义如下:

$$\begin{aligned} EPC_{\text{ship}}(es_1, es_2, es_3, ee_1, ee_2, ee_3, eq_1, eq_2, eq_3, ce_a, ce_s) \stackrel{\text{def}}{=} & EPIC_{\text{ship}}(ce_s, es_2) | \\ & EPOC_{\text{ship}}(ee_1, ce_a) | ESQC_{\text{ship}}(eq_2, eq_1, eq_3, es_1, es_3) | \\ & EENC_{\text{ship}}(ee_2, ee_3). \end{aligned} \quad (24)$$

2.4 跨组织子流程

定义 11. 跨组织子流程 CSP 是一个二元组 (LBP, EPC) :

- 1) LBP 是一个本地业务流程;
- 2) EPC 是建立在 LBP 上的外部流程控制。

以跨组织子流程启动和结束消息通道 s 和 e 、外部流程输入控制规则集合中所有外部活动的跨组织结束消息通道 $CE_i = \{ce_{i1}, \dots, ce_{is}\}$ 外部输出控制规则集合中所有外部活动跨组织启动消息通道 $CE_o = \{ce_{o1}, \dots, ce_{ot}\}$ 为参数, 跨组织子流程的行为如式 (25) 定义:

$$\begin{aligned} CSP(s, e, CE_o, CE_i) \stackrel{\text{def}}{=} & (\bigvee es_1 \dots es_n) \\ & (\bigvee ee_1 \dots ee_n) \wedge (\bigvee eq_1 \dots eq_n) \wedge LBP(ps, pe, \\ & ES, EE, EQ) | EPC(ES, EE, EQ, CE_o, CE_i). \end{aligned} \quad (25)$$

跨组织子流程是本地业务流程及建立之上的外部流程控制的一个封装, 跨组织子流程的启动即标志着本地业务流程的启动, 本地流程的结束则标志着跨组织子流程的结束. 本地流程启动后同时受到本地流程控制和外部流程控制的约束。

图 1 中的跨组织子流程 CSP_{price} 就是由第 2.2 节中定义的本地业务流程 LBP_{price} 和一个外部流程输入控制 $EPIC_{\text{price}}$ 的组合:

$$\begin{aligned} CSP_{\text{price}}(s, e, ce_s) \stackrel{\text{def}}{=} & (\bigvee es_1, es_2, es_3) \wedge (\bigvee ee_1, \\ & ee_2, ee_3) \wedge (\bigvee eq_1, eq_2, eq_3) \wedge LBP_{\text{price}}(ps, pe, \\ & ES, EE, EQ) | EPC_{\text{price}}(ES, EE, EQ, ce_s); \end{aligned} \quad (26)$$

$$\begin{aligned} EPC_{\text{price}}(ES, EE, EQ, ce) \stackrel{\text{def}}{=} & ce \cdot \overline{es_3} | eq_3 | \\ & eq_1 \cdot \overline{es_1} | eq_2 \cdot \overline{es_2} | ee_1 | ee_2 | ee_3, \end{aligned} \quad (27)$$

其中: $ES = \{es_1, es_2, es_3\}$, $EE = \{ee_1, ee_2, ee_3\}$, $EQ = \{eq_1, eq_2, eq_3\}$.

2.5 跨组织业务流程

在跨组织环境下, 由于构成跨组织工作流的子流程分属不同的组织, 集中管理的方式会破坏各组织的自治性, 因而我们将跨组织工作流活动之间的控制依赖建模为活动间的控制转移行为, 活动则建模为主动的自治实体, 每个活动自主地响应指定的输入转移行为, 并触发跨组织子流程的执行, 并在执行结束后产生指定的输出转移。

定义 12. 跨组织工作流活动 CWA 是一个五元组 $(CSP, ITS, join_type, OTS, split_type)$:

- 1) CSP 是跨组织工作流活动应执行的跨组织子流程;
- 2) $ITS = \{t_{i1}, \dots, t_{ip}\}$ 是活动应响应的输入转移行为的集合;
- 3) $join_type \in \{AND, OR\}$ 为所有输入转移的合并类型, 如果 $join_type = AND$, 那么只有当所有

的输入转移均到达后,方可启动 CSP;如果 $join_type = OR$,那么只要有一个输入转移到达后,即可启动 CSP;

4) $OTS = \{ \bar{t}_{o1}, c_1, \dots, \bar{t}_{oq}, c_q \}$ 是活动应产生的输出转移行为以及产生条件的集合;

5) $split_type \in \{AND, OR\}$ 为所有输出转移的分支类型,如果 $split_type = AND$,那么当 CSP 结束后,对于每个输出转移,只要条件满足就产生这个转移,如果 $split_type = OR$,则只产生一个条件满足的输出转移。

以 CWA 接收输入转移的通道 $IT = \{t_{i1}, \dots, t_{ip}\}$,发送输出转移的通道 $OT = \{t_{o1}, \dots, t_{oq}\}$,CSP 的外部流程输入和输出控制规则集合中的所有跨组织结束消息通道 $CE_i = \{ce_{i1}, \dots, ce_{is}\}$ 和 $CE_o = \{ce_{o1}, \dots, ce_{ot}\}$ 为参数,根据跨合并类型和分支类型的不同,跨组织 workflow 活动的行为如以下表达式定义:

$$CWA_{AA}(IT, OT, CE_o, CE_i) \stackrel{\text{def}}{=} (\bigvee s, e) \\ (((t_{i1} | \dots | t_{ip}).\bar{s}.e.([c_1 = \text{true}]\bar{t}_{o1} | \dots | \\ [c_q = \text{true}]\bar{t}_{oq})) | CSP(s, e, CE_o, CE_i)) \quad (28)$$

$$CWA_{AO}(IT, OT, CE_o, CE_i) \stackrel{\text{def}}{=} (\bigvee s, e) \\ (((t_{i1} | \dots | t_{ip}).\bar{s}.e.([c_1 = \text{true}]\bar{t}_{o1} + \dots + \\ [c_q = \text{true}]\bar{t}_{oq})) | CSP(s, e, CE_o, CE_i)); \quad (29)$$

$$CWA_{OA}(IT, OT, CE_o, CE_i) \stackrel{\text{def}}{=} (\bigvee s, e) \\ (((t_{i1} + \dots + t_{ip}).\bar{s}.e.([c_1 = \text{true}]\bar{t}_{o1} | \dots | \\ [c_q = \text{true}]\bar{t}_{oq})) | CSP(s, e, CE_o, CE_i)); \quad (30)$$

$$CWA_{OO}(IT, OT, CE_o, CE_i) \stackrel{\text{def}}{=} ((\bigvee s, e) \\ (((t_{i1} + \dots + t_{ip}).\bar{s}.e.([c_1 = \text{true}]\bar{t}_{o1} + \dots + \\ [c_q = \text{true}]\bar{t}_{oq})) | CSP(s, e, CE_o, CE_i)). \quad (31)$$

有两种特殊类型的 CWA 来标记一个跨组织业务流程的开始和结束,它们是跨组织业务流程启动(CSA)和结束活动(CEA),它们的行为由表达式(32)~(36)定义,其中 s 和 f 分别是跨组织 workflow 启动命令和结束消息通道。

$$CSA_{AND}(OT, s) \stackrel{\text{def}}{=} s.([c_1 = \text{true}]\bar{t}_{o1} | \dots | \\ [c_q = \text{true}]\bar{t}_{oq}); \quad (32)$$

$$CSA_{OR}(OT, s) \stackrel{\text{def}}{=} s.([c_1 = \text{true}]\bar{t}_{o1} + \dots + \\ [c_q = \text{true}]\bar{t}_{oq}); \quad (33)$$

$$CEA_{AND}(IT, f) \stackrel{\text{def}}{=} (t_{i1} | \dots | t_{ip}).\bar{f}; \quad (34)$$

$$CEA_{OR}(IT, f) \stackrel{\text{def}}{=} (t_{i1} + \dots + t_{ip}).\bar{f}. \quad (35)$$

定义 13. 跨组织业务流程 CBP 由一个跨组织流程启动活动 CSA、一个跨组织流程结束活动 CEA 和 n 个跨组织 workflow 活动 $CWAS = \{CWA_1, \dots, CWA_n\}$ 组成,其行为由如下表达式定义:

$$CBP(s, f) \stackrel{\text{def}}{=} (\bigvee t_1, \dots, t_z) \bigwedge (\bigvee ce_1, \dots, ce_z). \\ (CSA(OT_s, s) | CEA(IT_e, f) | (CWA_1(IT_1, \\ OT_1, CE_{o1}, CE_{i1}) | \dots | (CWA_n(IT_n, \\ OT_n, CE_{on}, CE_{in}))), \quad (36)$$

且:

1) $\forall t_i \in \bigcup_{i=1}^n CWA_i.ITS \cup CEA.ITS \Rightarrow$ 存在且只存在 $t_o \in \bigcup_{i=1}^n CWA_i.OTS \cup CSA.OTS$,使得 $t_i = t_o$;

2) $\forall t_o \in \bigcup_{i=1}^n CWA_i.OTS \cup CSA.OTS \Rightarrow$ 存在且只存在 $t_i \in \bigcup_{i=1}^n CWA_i.ITS \cup CEA.ITS$,使得 $t_i = t_o$;

$$3) \bigcup_{j=1}^n CE_{oj} \equiv \bigcup_{j=1}^n CE_{ij} \equiv \{ce_1, ce_2, \dots, ce_z\}.$$

如图 1 所示的跨组织流程由 4 个跨组织 workflow 活动:price, account, ship, invoice 和启动活动 start 及结束活动 end 组成,每个活动的输入和输出转移如图 1 所示,它们的定义为:

$$CWA_{price}(t_1, t_3, ce_s) \stackrel{\text{def}}{=} (\bigvee s, e) \bigwedge t_1.\bar{s}.e. \\ [c_3 = \text{true}]\bar{t}_3 | CSP_{price}(s, e, ce_s)); \quad (37)$$

$$CWA_{ship}(t_2, t_5, ce_s, ce_a) \stackrel{\text{def}}{=} (\bigvee s, e) \bigwedge t_2.\bar{s}.e. \\ [c_5 = \text{true}]\bar{t}_5 | CSP_{ship}(s, e, ce_s, ce_a)); \quad (38)$$

$$CWA_{invoice}(t_4, t_5, t_6) \stackrel{\text{def}}{=} (\bigvee s, e) \bigwedge (t_4 | t_5).\bar{s}.e. \\ [c_6 = \text{true}]\bar{t}_6 | CSP_{invoice}(s, e)); \quad (39)$$

$$CWA_{account}(t_3, t_4, ce_a) \stackrel{\text{def}}{=} (\bigvee s, e) \bigwedge t_3.\bar{s}.e. \\ [c_4 = \text{true}]\bar{t}_4 | CSP_{account}(s, e, ce_a)); \quad (40)$$

$$CSA_{start}(t_1, t_2, s) \stackrel{\text{def}}{=} s.([c_1 = \text{true}]\bar{t}_1 | \\ [c_2 = \text{true}]\bar{t}_2); \quad (41)$$

$$CEA_{end}(t_6, f) \stackrel{\text{def}}{=} t_6.f. \quad (42)$$

跨组织 workflow order 的定义为

$$CBP_{order}(s, f) \stackrel{\text{def}}{=} (\bigvee t_1, t_2, t_3, t_4, t_5, t_6) \\ (\bigvee ce_s, ce_a) \bigwedge (CSA_{start}(t_1, t_2, s) | CEA_{end}(t_6, f) | \\ CWA_{price}(t_1, t_3, ce_s) | CWA_{ship}(t_1, t_3, ce_a, ce_s) | \\ CWA_{account}(t_3, t_4, ce_a) | CWA_{invoice}(t_4, t_5, t_6)). \quad (43)$$

3 模型分析

3.1 一致性

Pi 演算定义了一种观察相等性,即不管两个过程的内部结构如何,从外部行为的角度观察,它们是相同的.观察相等性通过弱互模拟的关系描述.

定义 14^[15].弱互模拟是满足下述要求的过程上的一个对称的二元关系 $\mathcal{R}:P\mathcal{R}Q$,且若有 $P \xrightarrow{\alpha} P'$,则:

如果 $\alpha = a(x)$,则

$$\begin{aligned} &\exists Q'' : Q \Rightarrow^{a(x)} Q'' \wedge \forall u, \\ &\exists Q' : Q''\{u/x\} \Rightarrow Q' \wedge P'\{u/x\} \mathcal{R} Q'; \end{aligned}$$

如果 α 不是一个输入,则 $\exists Q' : Q \xrightarrow{\alpha} Q' \wedge P' \mathcal{R} Q'$;

如果 P 和 Q 可由一个弱互模拟相联,那么 P 和 Q 是弱互相似,记为 $P \approx Q$.

定义 15.如果两个跨组织子流程 CSP_A 和 CSP_B 弱互相似,那么这两个子流程等价,记为 $CSP_A = CSP_B$.

根据弱互模拟理论,可以帮助判断建立的业务流程是否满足既定的要求,或者帮助实现内部私有流程的外部抽象.例如图 1 中的“price”子流程,提供此流程的组织期望将计算货物价格的内部过程隐藏,对外只提供如图 4 示的一个抽象流程 CSP_{price_a} ,这个抽象流程的定义如下:

$$\begin{aligned} CSP_{price_a}(s, e, ce_s) \stackrel{def}{=} &(\bigvee es_1, es_2 \ \& \ \bigvee ee_1, ee_2) \\ &(\bigvee eq_1, eq_2 \ \& \ LBP_{price_a}(ps, pe, ES, EE, EQ) | \\ &EPC_{price_a}(ES, EE, EQ, ce_s)); \end{aligned} \quad (44)$$

$$\begin{aligned} LBP_{price_a}(ps, pe, ES, EE, EQ) \stackrel{def}{=} & \\ &(\bigvee s_2, e_1 \ \& \ AC_{1_a}(ps, e_1, es_1, ee_1, eq_1) | \\ &AC_{2_a}(s_2, pe, es_2, ee_2, eq_2)); \end{aligned} \quad (45)$$

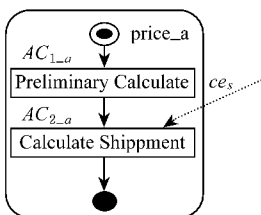


Fig. 4 Abstract process of calculating price.
图 4 价格计算抽象流程

$$EPC_{price_a}(ES, EE, EQ, ce) \stackrel{def}{=}$$

$$ce. \overline{es_2} | eq_2 | eq_1. \overline{es_1} | ee_1 | ee_2, \quad (46)$$

其中, $ES = \{es_1, es_2, es_3\}$, $EE = \{ee_1, ee_2, ee_3\}$, $EQ = \{eq_1, eq_2, eq_3\}$.

通过 Pi 演算的弱互模拟理论,我们可以判断这个抽象流程是否与实际流程(在第 2.4 节中定义)的外部行为一致.

命题 1. CSP_{price} 与 CSP_{price_a} 是等价的两个跨组织子流程.

证明. 因为 $CSP_{price} \approx CSP_{price_a}$ (证明略),所以 $CSP_{price} = CSP_{price_a}$. 证毕.

由于证明较繁琐,本文略去 $CSP_{price} \approx CSP_{price_a}$ 的证明过程,但给出通过验证工具得到的验证结果,如图 5 所示. Pi 演算经过 20 年的发展,不仅在形式化的理论上可以帮助理解移动并发的本质特征及系统的行为,而且还有许多成熟的验证工具,帮助建立正确的系统,如 Mobile Workbench (MWB) 就是一个可以操作和分析用 Pi 演算描述的移动并发系统的自动工具^[16].利用此工具,我们验证了 $CSP_{price} \approx CSP_{price_a}$ 的正确性.

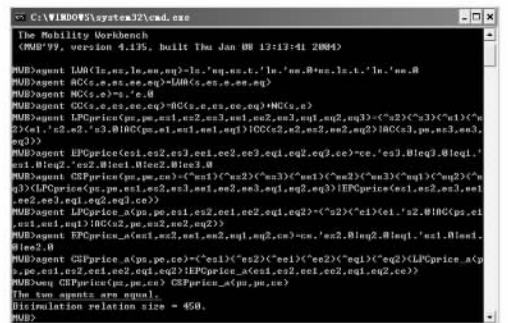


Fig. 5 Result of process equivalence verification from MWB.

图 5 流程等价性的 MWB 验证结果

3.2 组合性

由第 2 节的模型定义可以看出,基于 Pi 演算的代数特性,本文提出的跨组织工作流模型体现了跨组织业务流程的组合特性:一个跨组织业务流程是多个组织内的本地流程和这些流程间的全局控制的组合.这种组合流程的构造方式,不仅通过流程分解降低了流程结构的复杂性,便于跨组织流程的分析和验证,而且各子流程可以独立建模,互不影响,这不仅带来了流程的可重用性,而且一个子流程的变化不会影响其他的子流程,这种特点非常适合跨组织环境的动态性.

例如订单处理流程中增加一个海关关税征收子流程 impose, 而且 price 的“Calculate Custom”活动与 impose 中的“Impose Custom”活动之间存在一个外部控制依赖, 如图 6 所示:

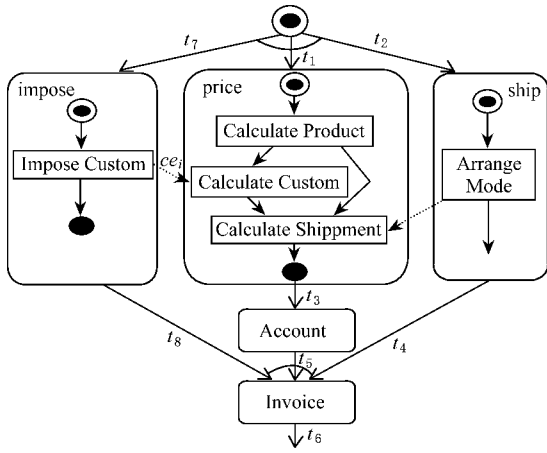


Fig. 6 The modified order processing business process.
图 6 修改后的订单处理业务流程

修改后的订单处理流程的定义由如下表达式定义:

$$CWA_{price}(t_1, t_3, ce_s, ce_i) \stackrel{\text{def}}{=} (\bigvee s, e \wedge t_1, \bar{s}, e, [c_3 = \text{true}] \bar{t}_3 | CSP_{price}(s, e, ce_s, ce_i)); \quad (47)$$

$$CWA_{ship}(t_2, t_5, ce_s, ce_a) \stackrel{\text{def}}{=} (\bigvee s, e \wedge t_2, \bar{s}, e, [c_5 = \text{true}] \bar{t}_5 | CSP_{ship}(s, e, ce_s, ce_a)); \quad (48)$$

$$CWA_{account}(t_3, t_4, ce_a) \stackrel{\text{def}}{=} (\bigvee s, e \wedge t_3, \bar{s}, e, [c_4 = \text{true}] \bar{t}_4 | CSP_{account}(s, e, ce_a)); \quad (49)$$

$$CWA_{invoice}(t_4, t_5, t_8, t_6) \stackrel{\text{def}}{=} (\bigvee s, e) ((t_4 | t_5 | t_8), \bar{s}, e, [c_6 = \text{true}] \bar{t}_6 | CSP_{invoice}(s, e)); \quad (50)$$

$$CWA_{impose}(t_7, t_8, ce_i) \stackrel{\text{def}}{=} (\bigvee s, e \wedge t_7, \bar{s}, e, [c_8 = \text{true}] \bar{t}_8 | CSP_{impose}(s, e, ce_i)); \quad (51)$$

$$CSA_{start}(t_1, t_2, t_7) \stackrel{\text{def}}{=} [c_1 = \text{true}] \bar{t}_1 | [c_2 = \text{true}] \bar{t}_2 | [c_7 = \text{true}] \bar{t}_7; \quad (52)$$

$$CEA_{end}(t_6) \stackrel{\text{def}}{=} t_6; \quad (53)$$

$$CSP_{price}(s, e, ce_s, ce_i) \stackrel{\text{def}}{=} (\bigvee es_1, es_2, es_3 \wedge \bigvee ee_1, ee_2, ee_3 \wedge \bigvee eq_1, eq_2, eq_3 \wedge LBP_{price}(ps, pe, ES, EE, EQ) | EPC_{price}(ES, EE, EQ, ce_s, ce_i)); \quad (54)$$

$$EPC_{price}(ES, EE, EQ, ce_s, ce_i) \stackrel{\text{def}}{=} ce_s, \bar{es}_3 | ce_i, \bar{es}_2 | eq_3 | eq_2 | eq_1, \bar{es}_1 | ee_1 | ee_2 | ee_3, \quad (55)$$

其中, $ES = \{es_1, es_2, es_3\}$, $EE = \{ee_1, ee_2, ee_3\}$, $EQ = \{eq_1, eq_2, eq_3\}$;

$$CBP_{order} \stackrel{\text{def}}{=} (\bigvee t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8 \wedge \bigvee ce_s, ce_i, ce_a \wedge CSA_{start}(t_1, t_2, t_7) | CEA_{end}(t_6) | CWA_{price}(t_1, t_3, ce_s, ce_i) | CWA_{ship}(t_2, t_5, ce_s, ce_a) | CWA_{account}(t_3, t_4, ce_a) | CWA_{invoice}(t_4, t_5, t_6) | CWA_{impose}(t_7, t_8, ce_i)); \quad (56)$$

由于本文提出的跨组织 workflow 模型的建模组合特性, 可以看出当增加一个新的子流程或者改变组织内的子流程之间的控制关系时, 并不会影响原有的定义, 比较上述例子的定义可以看出, 修改后的流程只需做很小的改动: 原有的本地业务流程均无需改动; 不受修改影响的跨组织子流程 CSP_{ship} , $CSP_{account}$, $CSP_{invoice}$ 也不需要修改, 只有 CSP_{price} 增加了一个外部输入控制; 在跨组织 workflow 活动级别上, 只需为 $CWA_{invoice}$ 增加一个输入转移行为, 为启动活动增加一个输出转移行为; 最后新增加一个可独立建模的跨组织 workflow 活动 CWA_{impose} .

4 结 论

本文结合 Pi 演算的面向流程、可利用并发操作符组合构造系统的优点, 提出了一种跨组织 workflow 模型, 在该模型中, 一个跨组织业务流程由分属不同组织的子流程组合而成, 每个子流程的内部执行过程由所属的组织自治管理, 子流程之间的执行约束则由它们之间的并发通信描述, 因而子流程内部的变化不会影响跨组织业务流程的建模, 同时子流程间跨组织控制关系的变化则不会影响子流程的内部建模.

另外, 在跨组织环境下, 往往需要隐藏组织内流程的执行细节, 因此关注的是组织内流程与其他子流程交互的外部行为, Pi 演算的观察相等性理论从外部行为角度定义了流程的相等性. 本文基于 Pi 演算的弱互模拟关系, 给出了跨组织子流程的相等性定义, 提供了判断两个流程是否具有相同外部行为的形式化方法.

跨组织 workflow 管理是一个动态的环境, 由于各子流程分属不同的 workflow 管理域, 变化和出现异常的情况时有发生, 对跨组织业务流程的事务管理和异常处理是进一步要解决的问题.

参 考 文 献

- [1] B Medjahed, B Benatallah, A Bouguettaya, *et al.* Business-to-business interactions : Issues and enabling technologies[J]. The VLDB Journal, 2003, 12(1): 59-85
- [2] T Amin, P H Keng. Inter-organizational workflow management in virtual healthcare enterprise [C]. The 4th Int 'l Conf on Enterprise Information Systems, Spain, 2002
- [3] Zhou Jiantao, Shi Meilin, Ye Xinming. Formal verification techniques in workflow process modeling [J]. Journal of Computer Research and Development, 2005, 42(1): 1-9 (in Chinese)
(周建涛, 史美林, 叶新铭. 工作流过程建模中的形式化验证技术[J]. 计算机研究与发展, 2005, 42(1): 1-9)
- [4] R M Bastos, D D A Ruiz. Extending UML activity diagram for workflow modeling in production systems[C]. In : Proc of the 35th Hawaii Int 'l Conf on System Sciences. Los Alamitos : IEEE Computer Society Press, 2002
- [5] W Aalst, K Hee. Workflow Management Models, Methods and Systems[M]. Cambridge : MIT Press, 2002
- [6] G M Giaglis. A taxonomy of business process modelling and information systems modelling techniques [J]. International Journal of Flexible Manufacturing Systems, 2001, 13(2): 209-228
- [7] W Aalst. Modeling and analyzing interorganizational workflows [C]. In : Proc of the 1998 Int 'l Conf on Application of Concurrency to System Design. Los Alamitos : IEEE Computer Society Press, 1998. 262-272
- [8] K Lenz, A Oberweis. Inter-organizational business process management with XML nets[C]. In : The 34th Hawaii Int 'l Conf on System Sciences. Los Alamitos : IEEE Computer Society Press, 2001. 243-263
- [9] Li Hongchen, Shi Meilin. The application of Petri nets in business process modeling[J]. Mini-Micro Computer Systems, 2001, 22(1): 29-32 (in Chinese)
(李红臣, 史美林. Petri 网在业务过程建模中的应用[J]. 小型微型计算机系统, 2001, 22(1): 29-32)
- [10] Liu Shuzhou, Angela Goh Eck Soong. A formal framework to support workflow adaptation [J]. International Journal of Software Engineering and Knowledge Engineering, 2002, 12(3): 245-267
- [11] R Milner. Communicating and Mobile Systems : The π -Calculus [M]. Cambridge : Cambridge University Press, 1999
- [12] M Förster. Theory of business process modelling : The Pi-calculus [C]. Seminar Process-Oriented Information Systems, Potsdam, Germany, 2003
- [13] H Smith, P Fingar. Workflow is just a Pi process [OL]. <http://www.bpm3.com/picalculus>, 2003-12-08
- [14] Y Dong, Z Shen-sheng. Approach for workflow modeling using π -calculus[J]. Journal of Zhejiang University Science, 2003, 4(6): 643-650
- [15] J Parrow. An introduction to the π -calculus[G]. In : Handbook of Process Algebra. Amsterdam : Elsevier Press, 2001. 479-543
- [16] B Victor, F Moller. The mobile workbench—A tool for the π -calculus [G]. In : Proc of the Computer-Aided Verification 1994, LNCS 818. Berlin : Springer, 1994. 428-440



Zhang Jing, born in 1973. Ph. D. candidate in computer software and theory from Shandong University, Jinan, China. Her main research interests include workflow and business process management.

张 静, 1973 年生, 博士研究生, 主要研究方向为工作流和业务流程管理。



Wang Haiyang, born in 1965. Professor and Ph. D. supervisor of Shandong University. His main research interests are software and data engineering, computer supported cooperative work and business process management.

王海洋, 1965 年生, 教授, 博士生导师, 主要研究方向为软件与数据工程、计算机协同工作和业务流程管理。



Cui Lizhen, born in 1976. Received his Ph. D. degree in computer software and theory in 2005. He has been lecturer of Shandong University since 2006. His current research interests include workflow and database application.

崔立真, 1976 年生, 博士, 讲师, 主要研究方向为工作流技术和数据库应用。

Research Background

Nowadays, workflow's application scope is crossing over the boundary of organization. Traditional workflow modeling methods can not meet the requirement of cross-organizational business processes. Pi-calculus is a kind of computing model for representing concurrent systems, which has strict formal semantics and also provides an approach to construct systems by composing subsystems. In this paper, we present a cross-organizational workflow modeling method based on Pi-calculus, which uses the concurrency operators and weak bismulation theory of Pi-calculus to model cross-organizational workflow, abstract internal private business process and verify the equivalence of cross-organizational sub-processes. A cross-organizational workflow model defined by this way builds a looselycoupled relationship among sub-processes, which makes this model adapt to dynamic cross-organizational environment. Our work is supported by the National Natural Science Foundation of China (60673130).