

基于 Pi-Calculus 的跨组织 workflow 建模方法^{*}

潘晓华¹, 冯志林^{1,2}, 尹建伟¹, 郑正平³, 董金祥¹

(1. 浙江大学 计算机科学与工程系, 浙江 杭州 310027; 2. 浙江工业大学 之江学院, 浙江 杭州 310024; 3. 浙江省交通规划设计研究院, 浙江 杭州 310006)

摘要: 针对跨组织协同环境下的业务流程建模技术的不足, 提出了一个新的用于跨组织业务流程的形式化建模方法。首先根据 workflow 管理联盟 (WfMC) 的有关 workflow 过程定义接口规范, 利用 Pi-Calculus 技术对跨组织业务流程结构进行了形式化定义。此外, 利用 Pi-Calculus 特有机对并发流程及其之间的通信进行描述。提出的方法可以有效用于不同业务流程间的协同工作, 并且适合于对分布式协同环境下的 workflow 进行建模。

关键词: workflow 建模; Pi-Calculus; 跨组织流程

中图法分类号: TP301.6 文献标识码: A 文章编号: 1001-3695(2006)01-0063-03

Modeling Method for Inter-Organizational Workflow Based on Pi-Calculus

PAN Xiao-hua¹, FENG Zhi-lin^{1,2}, YIN Jian-wei¹, ZHENG Zheng-ping³, DONG Jin-xiang¹

(1. Dept. of Computer Science & Engineering, Zhejiang University, Hangzhou Zhejiang 310027, China; 2. College of Zhijiang, Zhejiang University of Technology, Hangzhou Zhejiang 310024, China; 3. Zhejiang Provincial Plan Design & Research Institute of Communications, Hangzhou Zhejiang 310006, China)

Abstract: Due to some defects on the existing modeling methods for supporting inter-organizational processes, a novel formal modeling method was proposed. According to the specifications of workflow process definition interface instituted by Workflow Management Coalition (WfMC), firstly defined a formal foundation for structure modeling of inter-organizational business processes, which is based on Pi-Calculus technique. Then make use of the special mechanism provided by the Pi-Calculus technology to describe parallel processes and their communication. The proposed method works well on the collaboration between different workflow processes and appropriates for the modeling of distributed cooperative workflow systems.

Key words: Workflow Modeling; Pi-Calculus; Inter-Organizational Process

随着经济的全球化发展, 现代企业为了应对日益激烈的市场竞争, 满足复杂多变的业务需求, 其自身的组织结构正发生着巨大的变化, 并逐步跨入协同商务 (Collaboration Business) 的阶段。通过实现多个合作组织之间的协作, 可以将企业与不同的客户、供应商及其合作者紧密联系在一起, 从而构成一个完整的统一业务实体。在这种大的背景环境下, 各个业务组织边界的区分日趋模糊, 且正不断进行演化发展。跨组织流程日益成为组织生存和发展的基础, 针对该流程模型特点的工作流建模技术也就成为人们竞相关注的焦点。跨组织流程建模技术要求各业务伙伴之间进行相互合作, 同时合理调配供应链企业的资源和集成企业内外的各种应用, 以便更有利地实现企业间的并行运作, 从而有效提高跨组织流程对客户需求的快速响应能力。现有的 workflow 技术虽然能够较好地支持简单协同工作模型, 但在处理较为复杂的跨组织环境下的业务流程时, 还存在着很多的问题^[1], 这些缺陷直接限制了跨组织 workflow 技术在现代企业中的应用。

主要研究跨组织环境下的 workflow 形式化建模方法。跨组织 workflow 是由多个独立商业组织共同参与协同协作的 workflow

程, 这些商业组织之间的关系经常改变, 并且组织内部结构也会发生变化。针对跨组织 workflow 协同工作的特点, 我们引入 Pi-Calculus^[2,3] 技术, 并在其基础上提出一个新的形式化建模方法, 实现跨组织环境下的 workflow 动态建模。由于 Pi-Calculus 具备描述和解释并发流程互通信行为的能力, 因此这项技术非常适合用于跨组织 workflow 建模。

1 跨组织 workflow 建模相关研究

现代企业间的竞争日趋激烈化, 如今的竞争, 不再是简单的企业与企业之间的竞争, 而是供应链与供应链, 甚至是价值链与价值链之间的竞争, 任何一个企业依靠自身的力量都很难获得长足的发展。为了有效地整合企业外部资源, 抓住有限的市场机会, 跨组织协同协作成为企业发展的必然趋势。为了满足这种新的需求, workflow 管理系统必须能够提供对分布式异构环境下协同工作的支持^[4,5]。然而, 由于自身体系结构的缺陷, 现有 workflow 管理系统无法实现复杂的跨组织业务集成, 只能应用于简单的协同工作模型。

跨组织的工作流涉及分布在不同地域的资源, 需要通过协调多个参与者组织的活动和计算机应用来协同完成。实现组织之间、组织内部成员间的协调工作以达到业务的整体目标是跨组织 workflow 管理系统的出发点。针对跨组织业务集成, workflow 管理联盟 (WfMC) 定义了四个协同工作模型^[6]: 链式

(Chained), 子流程嵌套式 (Nested Subprocesses), 对等式 (Peer-to-Peer), 并行同步式 (Parallel-Synchronised), 有效实现了简单业务流程间的协同工作。WfMC 规定所有工作流协同工作模型必须实现以下两点: 首先是实现对流程定义的公共解释; 其次要实现引擎之间的控制信息相互转换和工作流相关数据和应用程序数据的传递。

近年来, 国内外学者提出了许多有关工作流建模的方法。Hommes 在文献 [7] 中指出, 到目前为止大约有 350 多种支持 BPR 的流程建模技术和工具被提出和实现, 这些建模技术各有特色, 但是在进行跨组织业务流程建模时都存在一定的缺陷^[8]: 数据流程图的可理解性强, 具有较好的计算机化能力, 但是缺乏抽象机制; IDEF 语言的表达清晰, 且能够进行分层次的图形化过程描述, 但是缺乏动态分析能力; UML 虽能有效反映客户需求, 但是缺乏严密有效的验证和分析方法; Petri 网是目前使用最广, 比较成熟的建模方法。由于 Petri 网模型强调人员和活动之间的协调性, 因此非常适合对一些较为复杂业务流程的形式化建模。但是, Petri 网模型本身存在着一些缺陷, 首先, Petri 网系统是一个封闭的网系统, 对外没有明确的输入和输出, 其动态行为是由内部状态的各种可能变化以及这些变化间的关系刻画的; 此外当网系统节点过多的时候, 状态空间的状态数将随着节点的增加而呈指数关系增长, 使得对复杂系统的分析非常困难。这些缺陷使得 Petri 技术不适合用于复杂的跨组织工作流建模。

综上所述, 目前的工作流技术不能很好地用于跨组织建模, 有必要引入新的建模方法, 基于这种情况提出了 Pi-Calculus 工作流建模技术。

2 Pi-Calculus 语法

Pi-Calculus 是一门用来描述并发和分布式系统的语言, 发明至今已有十余年, 它在并发和分布式理论研究中占据支配性地位。Pi-Calculus 包含两个实体概念^[9]: 流程 (Process) 和信道 (Channel)。流程独立地进行并发计算, 通过相互通信来交换信息, 而信道则是这些流程进行通信的渠道。名字 (Name) 是通过信道传输的变量, 它是通信信道的唯一标志, 两个流程通过共享一个特定的名字来实现交互。Pi-Calculus 非常适合于对并发和同步过程进行描述^[10], 在描述并发和同步过程时, 使用 Pi-Calculus 技术要比使用 Fork、数据共享和条件变量技术更方便, 比使用远程过程调用 (RPC) 技术更灵活。

Pi-Calculus 的基本语法定义如下:

$$\begin{aligned}
 P &::= \bar{a}[b_1 \dots b_n].P \\
 &a(x_1 \dots x_n).P \\
 &P|Q \\
 &(vx:T)P
 \end{aligned}$$

其中, $\bar{a}[b_1 \dots b_n].P$ 表示沿着信道 a 发送名字 $b_1 \dots b_n$, 发送完成后继续运行过程 P ; $a(x_1 \dots x_n).P$ 表示沿着信道 a 接收名字 $x_1 \dots x_n$, 接收完成后继续运行过程 P ; 操作符 $|$ 表示过程并发, $P|Q$ 中的过程 P 和过程 Q 可以并发地独立运行; $(vx:T)P$ 产生一个类型为 T 的新信道, 并在过程 P 中调用名字 x ; $\bar{\quad}$ 代表什么也不做的空过程。此外, $!u(y).Q$ 表示运行在信道 u 的一个服务, 该服务在每次被调用的时候, 产生一个新的过程 Q 的拷

贝, 而约束操作符 $(x)P$ 描述一个属于过程 P 的名字 x 。

基于上述的定义, 可以很方便地使用 Pi-Calculus 技术进行业务流程描述。例如 $\bar{u}x.P|u(y).Q$, 它包含子流程 $\bar{u}x.p$, 这个流程利用信道 u 发送数据 x , 数据发送完成后继续运行流程 P ; 与此同时, 流程 $u(y).Q$ 并发执行, 它利用信道 u 接收数据 y , 数据接收完成后继续运行流程 Q 。名字 y 是属于流程 Q 的一个形式化参数, 这两个流程的交互方式可以表述为 $\bar{u}x.P|u(y).Q \equiv P|Q\{x/y\}$ 。

3 Pi-Calculus 工作流建模

在利用 Pi-Calculus 技术对跨组织工作流进行建模时, 各个不同组织下的流程被分解成许多不同类型的活动, 这些活动依据它们之间的依赖关系进行组合。根据 WfMC 有关工作流过程定义接口规范描述, 我们对跨组织环境下的业务流程进行如下形式化定义:

工作流活动定义用来说明构成工作流过程的每一个基本活动, 常用的基本类型包括自动活动、手工活动和路由活动^[11]。

定义 1 如果活动 A 是自动活动, 那么它在流程中相应表示为

$$\begin{aligned}
 \text{def} & \text{ } \\
 A &= \text{request_resource}[resource_id].\text{START}. \\
 & \quad \text{assigned_resource}(resource_id).\text{ACTION}. \\
 & \quad \text{release_resource}[resource_id].\text{FINISH}
 \end{aligned}$$

其中, 大写名字 (如 START, ACTION, 和 FINISH) 表示 Pi-Calculus 中的过程, 定义的小写名字表示信道 (如 request_resource, assigned_resource 和 release_resource), 或是表示与相邻活动进行通信的端口 (如 resource_id)。由定义可知, 自动 (Automatic) 活动完全由工作流引擎自动控制, 一旦满足开始条件, 工作流引擎自动执行该活动, 活动完成后, 自动运行其离开动作。

定义 2 如果活动 A 是手动活动, 那么它在流程中的相应表示为

$$\begin{aligned}
 \text{def} & \text{ } \\
 A &= \text{request_resource}[resource_id].\text{START}. \\
 & \quad \text{assigned_resource}(resource_id).\text{wait_user}[role_id].\text{ACTION}. \\
 & \quad \text{release_resource}[resource_id].\text{FINISH}
 \end{aligned}$$

由定义可知, 手动 (Manual) 活动通过与用户交互来触发活动的开始或结束。在这样的系统中, 活动的开始或者完成都是用户交互的结果。如定义所示, 手动活动在分配资源以后, 使用信道 wait_user 和端口 role_id 等待用户完成任务, 只有在用户完成操作以后, 才会继续运行过程 ACTION。

定义 3 如果活动 A 是路由活动, 那么它在流程中的相应表示为

$$\begin{aligned}
 \text{def} & \text{ } \\
 A &= \text{request_resource}[resource_id].\text{START}. \\
 & \quad \text{assigned_resource}(resource_id).\text{ACTION}. \\
 & \quad \text{release_resource}[resource_id].\text{FINISH}
 \end{aligned}$$

由定义可知, 路由 (Route) 活动又称为虚活动, 能够描述层叠转移条件。路由活动 (虚活动) 即没有执行者, 也没有应用程序, 路径活动的运行对工作流相关数据或应用程序数据没有任何影响。它只进行路由判断选择执行路径, 形式化定义和自动活动一样。

此外, WfMC 用 Join 描述一个活动的多个进入转移, 用

Split 描述一个活动的多个转出转移。AND 和 OR 是转移的两种不同方式。AND-Split 定义了可能并发执行的、代表这个活动转出转移的流程。如果是有条件转移, 并发线程的数量由每个转移的转移条件决定, 这些转移条件的计算是并发进行的。OR-Split 定义了活动的转出转移列表, 其中转移都是根据转移条件来选择执行的。选择哪条转移路径, 需要根据每个转移的转移条件来决定, 按照转移在列表中的顺序来对转移条件进行逐个判断, 满足则执行, 不满足就判断下一个。OR-Join 只要活动的任一进入转移的转移条件得到满足, 活动就被初始化。AND Join 可以看作是活动的预处理集合点, 直到活动的所有进入转移的条件都满足, 活动才被初始化。

定义 4 如果活动 A 和活动 B 之间有 OR-Split 依赖, 并且它们在相应的流程中分别被表示为 A, B, 那么 OR-Split 依赖可以被定义为:

$$A \text{ OR-Split } B = (\nu p) (\overline{\text{start}} . [p] \mid ([p/\text{start}] A + [p/\text{start}] B))$$

定义 5 如果活动 A 和活动 B 之间有 OR-Join 依赖, 并且它们在相应的流程中分别被表示为 A, B, 那么 OR-Join 依赖可以被定义为

$$A \text{ OR-Join } B = (\nu p) (([p/\text{finish}] A + [p/\text{finish}] B) \mid \overline{\text{finish}} . [p])$$

定义 6 如果活动 A 和活动 B 之间有 AND-Split 依赖, 并且它们在相应的流程中分别被表示为 A, B, 那么 AND-Split 依赖可以被定义为

$$A \text{ AND-Split } B = (\nu p) (\overline{\text{start}} . [p] \mid ([p/\text{start}] A \mid [p/\text{start}] B))$$

定义 7 如果活动 A 和活动 B 之间有 AND-Join 依赖, 并且它们在相应的流程中分别被表示为 A, B, 那么 AND-Join 依赖可以被定义为

$$A \text{ AND-Join } B = (\nu p) (([p/\text{finish}] A \mid [p/\text{finish}] B) \mid \overline{\text{finish}} . [p])$$

转移信息描述活动间的转移, 也描述在工作流运行期间转移发生与否的条件。转移条件表达式以工作流相关数据为基础, 当条件满足时发生转移。如果没有标记转移条件, 那么就默认转移条件永远为真。

定义 8 如果条件表达式 的值为真, 执行活动 A, 否则执行活动 B。形式化定义如下:

$$\text{If } \overline{\text{decision_exp}} \text{ then } A \text{ else } B = (\nu a, b) (\overline{\text{decision_exp}} . \text{START} . ([= \text{TRUE}] a + [= \text{FALSE}] b) \mid ([a/\text{start}] A \mid [b/\text{start}] B))$$

其中, 信道 decision_exp 对条件表达式 进行判断, 根据表达式判断结果, 由私有端口 a 和 b 指明将要执行的是哪一个活动 (A 或是 B)。

4 应用实例

为了解释如何利用 Pi-Calculus 技术进行跨组织环境下的 workflow 建模, 构建了一个跨组织 workflow 模型, 该模型包括四个局部组织下的 workflow: 消费者流程, 经销商流程, 生产商流程和供应商流程。消费者流程通过给网上商城发送订单来订购某件商品。若网上商城发现没有存货, 则需要向经销商订货。同样, 经销商查询库存, 若发现没有该商品库存, 则需要向生产商进货。生产商流程查询产品库存, 根据库存来决定是从仓库发货、或是紧急生产、或是向其他产品供应商外购。在执行上

述流程之后, 根据反馈信息确认消费者订单, 计算出发货日期并通知用户, 确认用户订单。图 1 给出了利用图形化工具所构建出的上述跨组织 workflow 模型。

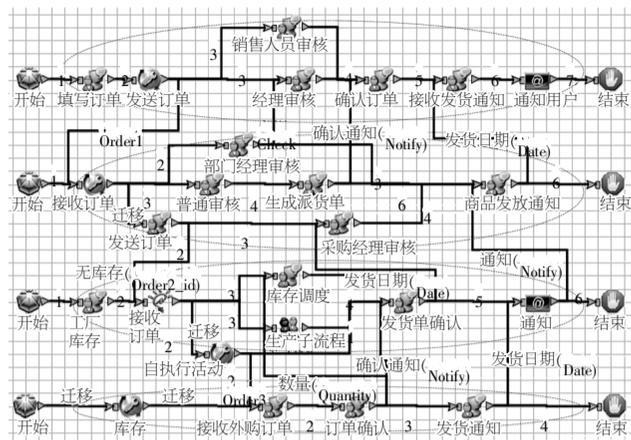


图 1 跨组织环境下的 workflow 模型组成

图 1 包含了四个不同局部组织环境下的业务流程, 每个流程隶属于不同的业务组织结构。在整个流程运行期间, 各个流程间需要进行相互通信, 并根据通信传输的信息进行路由判断, 协同完成整个跨组织流程的运行。

“消费者流程”中的发送订单 (send_order1) 活动属于自动活动, 其定义如下:

$$\text{Activity}(\text{send_order1}) = \overline{\text{request_resource}}[\text{resource_id}] . \text{START} . \overline{\text{assigned_resource}}(\text{resource_id}) . \text{send_order1} . \overline{\text{release_resource}}[\text{resource_id}] . \text{FINISH}$$

“消费者流程”中的经理审核 (Mgr_Audit) 是属于活动, 其定义如下:

$$\text{Activity}(\text{Mgr_Audit}) = \overline{\text{request_resource}}[\text{resource_id}] . \text{START} . \overline{\text{assigned_resource}}(\text{resource_id}) . \text{wait_user}[\text{role_id}] . \text{Mgr_Audit} . \overline{\text{release_resource}}[\text{resource_id}] . \text{FINISH}$$

“生产商流程”中的接收订单 2 (receive_order2) 属于虚活动, 其定义如下:

$$\text{Activity}(\text{receive_order2}) = \overline{\text{request_resource}}[\text{resource_id}] . \text{START} . \overline{\text{assigned_resource}}(\text{resource_id}) . \text{receive_order2} . \overline{\text{release_resource}}[\text{resource_id}] . \text{FINISH}$$

下面再利用 Pi-Calculus 技术对图 1 中不同组织流程间的典型通信过程进行描述, 由图 1 所示, “消费者流程”和“经销商流程”之间的通信描述如下:

(1) 如果 Activity(send_order1) 通过信道 Order1 发送订单给 Activity(receive_order1), 通信过程可以这样来描述

$$(\nu \text{order1}) (\overline{\text{start}} . [\text{order1}] \mid ([\text{order1}/\text{start}] \text{Activity}(\text{send_order1}) \mid [\text{order1}/\text{start}] \text{Activity}(\text{receive_order1})))$$

(2) 如果活动“商品发放通知”Activity(Notify), 通过信道 Notify 发送发货通知给活动“接收发货通知”Activity(receive_notify), 通信过程可以这样来描述

$$(\nu \text{notify}) (([\text{notify}/\text{finish}] \text{Activity}(\text{notify}) + [\text{notify}/\text{finish}] \text{Activity}(\text{receive_notify})) \mid \overline{\text{finish}}[\text{notify}])$$

(3) 如果“经销商流程”中的活动“部门经理审核”Activity(PMgr_Audit) 通过信道 Check 发送确认信息给“消费者流程”中的“经理审核”Activity(Mgr_notify), (下转第 76 页)