维普资讯 http://www.cqvip.com



# 基于 Pi-演算的工作流模式描述

薛 岗[1] 姚绍文[1] Joan Lu[2]

(云南大学网络智能计算实验室 昆明 650091)[1] (Huddersfield 大学计算机与软件学院 英国)[2]

**摘 要** 工作流模式包含了业务流程的基本特征和共性,其实现依赖于具体的流程建模语言或方法。Pi-演算是一种进程代数,可以应用于业务流程的形式化建模。本文使用 Pi-演算作为业务流程形式化的工具,对存在多种 BPMN 表示的工作流模式进行形式化建模,以探究工作流模式 Pi-演算的描述,检验 Pi-演算对业务流程行为特征的表现能力。

**关键词** 业务流程建模,工作流模式,Pi-演算

## Workflow Patterns Description in Term of Pi-calculus

XUE Gang[1] YAO Shao-wen[1] Joan Lu[2]

(Lab of Network Intelligent Computing, Yunnan University, Kuming 650091, China)[1]

(School of Computing and Engineering, University of Huddersfield, UK)[2]

**Abstract** Workflow patterns contain basic features of business process. How to implement these patterns depends on the modeling languages and methods. Pi-calculus, as a kind of process algebra, can be applied in business process modeling. This paper uses Pi-calculus, as a formalizing utility, to investigate some workflow patterns that may have multiple presentation versions with BPMN. The main goal is to explore expressive capabilities of Pi-calculus regarding business process and check representations of workflow patterns in Pi-calculus.

**Keywords** Business process modeling, Workflow patterns, Pi-calculus

## 1 引言

工作流管理系统(WFMS)最重要的功能之一就是业务流程建模,该功能可以使得工作流管理系统被看作一个元业务数据(如过程和组织图等)的仓库[1]。业务流程建模技术主要分为非形式化方法(如 SADT, IDEF0 等)及形式化方法(如 Petri 网、进程代数和有限状态机等)。业务流程在实际应用中可能包含大量复杂结构,然而,不同工作流产品或工作流描述语言对建模元素的理解、建模元素语义表达的差异导致了对相同业务流程的多种表示方式。另一方面,W. M. P. van der Aaslt 等人对商业环境下业务流程应用进行分析,提出多种独立于具体工作流描述语言的工作流模式[2]。这些模式包含了业务流程的基本结构、扩展结构,规定了工作流管理系统中的业务流程应具有的基本结构特征。基于 Pi-演算和 Petri 网的方法都可以应用于分析和验证同步系统模型,基于 Petri 网的建模方法主要提供直观的图形表示,Pi-演算则是基于文本的公式化表示[3]。基于 Petri 网的方法可较为准确地描述工作流模式,但部分工作流模式向 Petri 网映射的时候具有一定难度[4]。作为一种进程代数,Pi-演算同 Petri 网一样具有严格的数学基础,可以用于业务流程的建模。使用其描述的工作流模式具有简洁、具体,并弥补了部分基于 Petri 网方法对工作流模式进行建模的不足[7,8],但 Pi-演算不提供图形表示。

为了跨越业务流程设计与实现之间的距离,以便业务流程的表示可以被多种用户所理解,BPMI(the Business Process Modeling Initiative)开发了业务流程建模图形符号标准 BPMN[9],该标准支持直接将 BPMN 映射成为基于 XML 的业务流程执行语言(BPEL4WS)。在实现工作流模式时,部分工作流模式出现了多种 BPMN 描述版本。为探究工作流模式的 Pi-演算描述,检验 Pi-演算对业务流程行为特征的表现能力,本文对具有多种 BPMN 实现模式进行讨论,并使用 Pi-演算进行描述。

本文第 2 部分概述了工作流模式、BPMN 和 Pi-演算的相关技术及理论;第 3 部分就工作流模式多种实现版本进行讨论,并使用 Pi-演算进行描述;第 4 部分介绍了对本文所描述的模型进行检验的方法;第 5 部分介绍了与本文相关的部分工作;最后进行全文总结并说明了下一步的工作。

## 2 相关技术和理论概述

### 2.1 工作流模式

工作流模式包含了业务流程的基本结构,独立于具体业务流程描述语言,描述了业务流程在实际应用中所必须满足的基本结构特征。这些模式为商业工作流产品之间在技术上进行比较提供了基础,具体包含了基本的业务流程结构和扩展结构,主要包含 6 个分类、20 个模式[2]。

基本控制流模式:顺序、平行分支、同步、排它选择、简单合并;

高级分支和同步模式:多选、同步合并、多合并、鉴别者;

结构模式:任意循环、隐含终止;

涉及多实例模式:非同步多实例、设计时确定多实例、运行时确定多实例、非确定多实例;

基于状态的模式:延时选择、交替平行路径、里程碑;

取消模式:取消活动、取消案例。

### 2.2 业务流程建模符号(BPMN)

薛 岗 博士研究生;姚绍文 博士,教授;Joan Lu 博士,教授。

维普资讯 http://www.cqvip.com

开发 BPMN 的初衷是想为业务流程建模提供一种简单的机制，同时亦能处理业务流程本身所包含的复杂性。解决这两个相互矛盾的方法就需要将图形符号的图形表征进行分类。BPMN 提供一个图形分类集，方便迅速地识别这些符号或图形，同时可在保证不改变这些图形符号基本外观的情况下添加额外信息。

BPMN 中有 4 个基本的元素类：流对象(Flow Object)、连接对象(Connecting Object)、泳道(Swimlanes)、工件(Artifacts)。其中：

1) 流对象(Flow Object)是用来描述商业流程行为的图形元素，有三种“流对象”：事件(Events)、活动(Activities)、闸道(Gateways)；

2) 连接对象是流对象之间的互连或和其它信息的连接：顺序流(Sequence Flow)、消息流(Message Flow)、联系(Association)；

3) 泳道(Swimlanes)是对主要的建模元素进行分组，使用池(Pools)、道(Lanes)；

4) 工件(Artifacts)是增加关于流程的附加信息。有四种标准的工件，建模者或建模工具可以自行增加工件，工件集有：数据对象(Data Object)、组(Group)、注解(Annotation)。

### 2.3 Pi-演算

Pi-演算是以过程间移动通讯为研究重点的并发理论，是对 CCS(Calculus of Communication System)的发展[5,6]。其基本计算实体为名字和过程，过程之间的通讯通过名字来完成。Pi-演算可用来描述结构不断变化的并发系统。

Pi-演算的基本语法定义如下[5,6]：

设 $N$ 为无限名字集，用 u,v,w,x,y,z 等小写字母表示名字集上的名字；过程标识符用 A,B,C 等大写字母表示；过程表达式用 P,Q,R 等大写字母表示，过程表达式是以下几种表示之一：

$P::=0$　空表达式

$P_1+P_2$　和表达式

$P_1 \mid P_2$　并行表达式

$(x)P$ 或 $(\nu x)P$　限制表达式

$[x=y]P$　匹配表达式

$y(x).P$ 或 $y.P$　输出前缀

$\bar{y}\langle x\rangle.P$ 或 $\bar{y}.P$　输入前缀

$\tau.P$　哑前缀

$!P$　重复表达式

$A(y_1,y_2,\cdots,y_n)$过程标识符

### 2.4 映射原则

为使用 Pi-演算对 BPMN 表示的业务流程进行描述，需要建立起 BPMN 中的图形元素和 Pi-演算中的计算实体之间的对应关系。在工作流管理系统中，每一个活动是一个独立的流程。流程具有前置和后置条件，流程前置条件为：必须等待本流程以前的流程执行完毕才能开始执行；流程的后置条件为：本流程执行结束以后触发其它流程进行执行[8]。ECA(Event/Condition/Action)方法可以应用于过程定义的映射[10]，过程前置条件遵循 ECA 原则的事件和条件部分，后置条件遵循 ECA 原则的行为部分。

具体映射原则为：

- “流对象”可以直接映射成为 Pi-演算中的过程实体；
- “连接对象”中的“顺序流”和“消息流”可以直接映射成为 Pi-演算中的通道或连接实体，其区别在于通道中是否带参数；
- “工件”中的“数据对象”可以映射成为 Pi-演算中的参数实体；
- “池”、“道”和“组”为业务流程范围表示，在 Pi-演算中，过程表达式中的符号与过程的组合具有相同语义；
- “联系”和“注释”为图形展示，非计算实体。

过程触发是通过激活一个过程的输入通道来实现的。当过程在执行中带有可选条件时，通道激活以后进行可选条件检查。过程的执行是不可见的操作，过程执行完毕以后使用输出通道触发其它过程进行执行。

## 3 工作流模式的 Pi-演算描述

本部分内容仅针对工作流模式的多种 BPMN 实现进行讨论，并使用 Pi-演算进行形式化建模。关于模式的详细讨论及已经实现的形式化描述结果请参阅文献[2,7,8]。

### 3.1 平行分叉模式

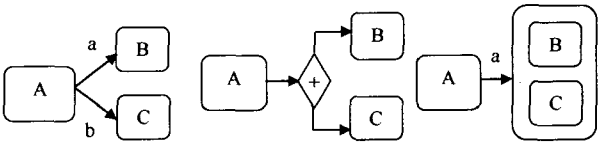


图 1　平行分叉模式三种 BPMN 参考实现

针对平行分叉模式，BPMN 的三种参考实现如图 1 所示。第三个版本是基于子流程的实现，使用“平行盒”的方式显示了一个流程中所包含的子流程 B 和子流程 C，它们互为并行。使用 Pi-演算的描述可以为：

$P_A(a)=\tau.\bar{a}$

$P(a)=(\nu b,c)\ a\ (Start(b,c)\mid P_B(b)\mid P_C(c))$

$Start(x,y)=(\bar{x}\mid\bar{y})$

$P_B(b)=b.\tau.P'_B(b)$

$P_C(c)=c.\tau.P'_C(c)$

过程 A 的后继过程为一复合过程，使用过程 Start 作为触发过程，触发过程 B 和过程 C 的并发执行。过程 B 和过程 C 执行完自身的任务以后变为状态 B’和 C’。BPMN 规范推荐在子过程中使用开始事件，但不强制使用，所以过程 Start 可以是隐含的或者直接使用开始事件来表示。同理，在同步化模式中，使用“平行盒”包含子流程，使用一个 End 过程来完成同步化工作，定义类似于 $End=a.b.\bar{c}$，通道 a 和 b 上都包含输入使用通道 c 触发后继活动。

### 3.2 多选模式

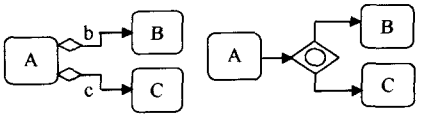


图 2　多选模式两种 BPMN 参考实现

BPMN 的两种参考实现如图 2 所示。第一个版本是基于 BPMN 条件流 b,c 连接后继流程 B 和 C，条件流工作时需要进行条件测试，测试通过才能激活通道。使用 Pi-演算的描述可以为：

$True(x)=\bar{x}\langle T\rangle$

$False(x)=\bar{x}\langle F\rangle$

$Evaluate(x,y,z)=x(u).\ y(v).\ ([u=v]True(z)+[u\neq v]False(z))$

维普资讯 http://www.cqvip.com

$Link(x,y,z) = (v\,w)\ Evaluate(x,y,w).\ w:(T\Rightarrow\bar{z},F\Rightarrow 0)$

$P_A = (v\,b_1,b_2,c_1,c_2)\ \tau.(Link(b_1,c_1,b) \mid Link(b_2,c_2,c))$

$P_B(b) = b.\tau.P'_B(b)$

$P_C(c) = c.\tau.P'_C(c)$

*Evaluate* 是测试过程，测试通道 $x$ 上的取值是否于给定条件 $y$ 匹配。如果匹配，返回布尔常量 $T(true)$ 或 $F(false)$，使用 Link 过程激活连接行为（上述描述中，Link 过程使用了“匹配表达式”的简写形式）。

### 3.3 任意循环模式

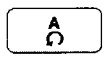


图 3 基本任意循环 BPMN 参考实现

图 3 为使用循环标志的循环流程，流程 A 可以为任务或子流程，Pi-演算描述为 $P_A = \tau.P_A$，该过程包含有输入和输出时可以描述为 $P_A(a,b) = a.\tau.(\bar{a}.P_A(a,b) + \bar{b})$。

由于 BPMN 的图形元素可以将业务流程表示成为编程语言中常见的 While 和 Until 循环，因此任意循环模式还可以表示成为图 4。

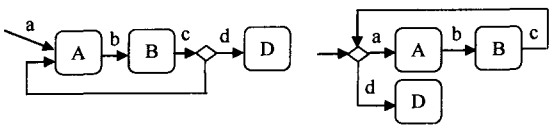


图 4 业务流程 Until 循环和 While 循环

图 4 表示业务流程中的 Until 循环和 While 循环，其 Pi-演算的描述为：

Until: $P_A(a,b) = !\ a.\tau.\bar{b}$

$P_B(b,c) = !\ b.\tau.\bar{c}$

$P_C(c,a,d) = !\ c.(\bar{a}+\bar{d})$

$P_D(d) = d.\tau.P'_D(d)$

While: $P_C(c,a,d) = !\ c.(\bar{a}+\bar{d})$

$P_A(a,b) = !\ a.\tau.\bar{b}$

$P_B(b,c) = !\ b\tau.\bar{c}$

$P_D(d) = d.\tau.\ P'_D$

可见 While 和 Until 循环的 Pi-演算描述非常类似，区别在于条件判别所处位置的不同。过程 C 在图中为菱形的闸道，实际应用中可在过程 C 中加入判别表达式。

任意循环模式也可以被看作是多实例模式的一种，多实例模式中并没有规定流程的多个实例是并行执行还是串行执行，所以多实例模式中如果流程的多个实例是按顺序执行，则可以按任意循环方式进行建模。

### 3.4 非确定多实例模式

在日常生活的应用场景中，业务活动的运行次数可能无法确定，例如商品销售过程无法确定商品的确切销售数量。计算机编程语言的违例处理也存在这样的问题，一段程序包含多个可能的违例语句，实际执行过程中可能同时出现多个或者不出现违例情况。使用 BPMN 符号元素实现如图 5 中所示过程也是非确定多实例模式的一种实现（图 5）。

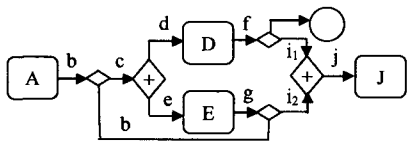


图 5 非确定多实例模式 BPMN 参考实现

过程 E 判别是否需要过程 D 的实例，过程 D 执行完毕和过程 E 同步化激活后继过程 I。其 Pi-演算描述可以为：

$P_A(b) = \tau.\bar{b}$

$P_B(b,c) = !\ b.\bar{c}$

$P_C(c,d,e) = !\ c.(\bar{d}|\bar{e})$

$P_D(d,f) = !\ d.\tau.\bar{f}$

$P_E(e,g) = !\ e.\tau.\bar{g}$

$P_F(f,i1) = !\ f.(\bar{i}_1 + 0)$

$P_G(g,i_2,b) = !\ g.(\bar{i}_2 + \bar{b})$

$P_I(i_1,i_2,j) = i_1.i_2.\bar{j}$

工作流模式中包含四个实例模式，其中设计时确定多实例、运行时确定多实例、非确定多实例模式都涉及到等待所有实例运行结束的同步问题。借助“容器”的概念，多实例过程可以当作一个特殊的过程，需要运行并管理其包含的多个实例。例如，设计时确定多实例模式可以描述成为：

$P_A(b) = \tau.\bar{b}$

$P_B(b,c) = (v\,s,e)b.\ (\bar{s}.\bar{s}.\bar{s} \mid P_{B1}(s,e) \mid e.e.e.\bar{c})$

$P_{B1}(s,e) = !\ s.\tau.\bar{e}$

$P_C(c) = c.\tau.P'_C$

上式中包含过程 B1 的三个实例，过程 B 负责实例化并等待所有实例运行结束以后激活后续活动。

### 3.5 里程碑模式

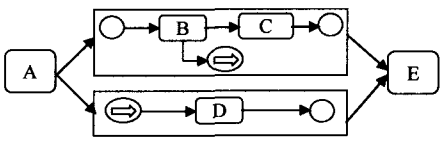


图 6 里程碑模式 BPMN 参考实现

图 6 是使用 BPMN 链接事件实现的一种里程碑模式。图中包含两个子流程，流程 D 的执行必须依赖于流程 B 的执行，流程 D 必须在流程 B 执行结束且在流程 C 还未开始执行之前开始执行。其 Pi-演算描述可以为：

$M(x,y) = x.\bar{y}$

$P_B(b,c,s) = (v\,l)\ (b.\tau.\bar{l}.\bar{c} \mid M(l,s))$

$P_C(c) = c.\tau.P'_C(c)$

$P_D(s) = (v\,d)\ (M(s,d) \mid d.\tau.P'_D(d))$

上式中的 M 表示链接事件，形式化建模的时候也可以直接省略链接事件直接使用通道实现图 6 中的流程。同时，还可以对过程 M 进行改进，引入链表结构[|L|](x)存储多个状态或数据。

## 4 流程模型的分析

业务流程应用以其正确性作为前置条件。形式化以后的业务流程模型可能会存在如死锁、活锁等错误，使用 Pi-演算描述的工作流模型必须满足两个基本性质，以保证其正确性[7]：(1)业务流程由初时状态开始，结束状态总是可达的；(2)业务流程中的任何一个活动都是可以执行的。

Pi-演算的分析工具 MWB'99 (The Mobility Workbench)[18]可以通过命令行交互的方式对 Pi-演算所描述的模型进行分析检查。本文所建立的流程模型可以在 MWB'99 中进行观察和分析。例如，本文所建立的平行分叉模式可以使用 $P_{PS} = (v\,a)\ (P_A(a) \mid P(a))$来描述整个系统模型，并假设过程 B′ 和过程 C′为“空表达式”(即 0)。整个系统运行状态的变迁如下所示(其中^表示 $v$，~$v$ 表示“值”，'$a$ 表示输

维普资讯 http://www.cqvip.com

出操作，$a$ 表示输入操作，$t$ 表示 $\tau$ 操作）：

步骤 0　$t.(\sim v).('\sim v.0|(b,c)\sim v.(Start(b,c)|P_B(b)|P_C(c)))$

步骤 1　$t.(\sim v,\sim v2).('\sim v.0|'\sim v2.0|\sim v.t.0|\sim v2.t.0)$

步骤 2　$t.(\sim v).('\sim v.0|t.0|\sim v.t.0)\ t.(\sim v).('\sim v.0|\sim v.t.0|t.0)$

步骤 3　$t.(t.0|t.0)\ t.(\sim v)('\sim v.0|\sim v.t.0)$

步骤 4　$t.t.0\ t.t.0$

步骤 5　$t.0$

整个变迁过程说明：系统模型由过程 $P_A$ 开始，并中止与空操作（0）；整个系统中的所有活动都被执行过。使用 MWB'99 工具的死锁检查时，$P_{PS}$ 中存在"死锁"，其原因主要是过程 B'和 C'被假设为"空表达式"。如果将上述两个过程替换成为过程 B 和过程 C（即过程工作完毕以后再次处于准备状态），则可避免"死锁"的发生。

## 5　相关工作

关于工作流相关模式的研究并不局限于工作流模式本身，还包含工作流资源模式、工作流数据模式等[16,17]。使用扩展着色 Petri 网对分布式系统进行建模和分析也形成了相应的模式。除使用 Petri 网和 Pi-演算以外，统一建模语言（UML）2.0 的活动图也被用于业务流程的模式分析上[11]。大量工作流建模方面的研究产生了很多基于图形的和基于形式化语言的业务流程的描述方法。Petri 网或 Pi-演算为大多数业务流程建模标准提供了理论上的支撑，这些建模标准包含 WS-CDL，YAWL，BPEL 等等[12]。

Petri 网和 Pi-演算之间的关系研究也在展开。文献[13]中阐述了 Pi-演算的 Petri 网语义及文献[14]中研究了 Pi-演算转换成 Petri 网的方法。

Pi-演算还被应用于 Web 服务组合的描述，文献[15]中使用 Pi-演算对 Web 服务及其组合进行形式化描述和建模，并对描述的模型进行了验证。

**结束语**　本文对部分工作流模式多种 BPMN 实现进行讨论，并使用 Pi-演算描述了这些模式的实现。通过讨论，说明 Pi-演算理论的应用可扩展到工作流领域，特别是业务流程建模方面。所建立的模型在语义上具有简练和清晰的特点，因此在工作流管理系统中的业务流程设计阶段采用基于 Pi-演算的建模方法有助于清楚地描述业务流程的行为。

今后的研究或开发工作主要可以在如下几个方面展开：继续对 Pi-演算相关理论进行深入细致的研究，特别是"标签"特性和"分布式 Pi-演算"等，这些问题的研究有助于继续完成工作流数据和资源模式的 Pi-演算形式化建模，实现业务流程的化简、仿真、分析和验证等功能；由于 Pi-演算不提供图形化的描述，可以针对业务流程的特点，开发基于 Pi-演算的业务流程描述符号库；开发 Pi-演算业务流程执行系统，对本文相关研究成果进行验证，也是将面向功能的特性引入应用领域而进行的有益尝试。

## 参 考 文 献


[1] van der Aalst W, van Hee K. 工作流管理—模型、方法和系统. 王建民，等译. 清华大学出版社，2004

[2] van der Aalst W M P, ter Hofstede A H M, et al. Workflow Patterns. Distributed and Parallel Databases, 2003, 14(1): 5-21

[3] van der Aalst W M P. Pi calculus versus Petri nets: Let us eat "humble pie" rather than further inflate the "Pi hype". BPTrends, 2005, 3(5): 1-11

[4] van der Aalst W M P, ter Hofstede A H M. Workflow Patterns: On the Expressive Power of (Petri-net-based) Workflow Lanuages // Jensen K, ed. Proceedings of the Fourth Worksho Pon the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2002), volume 560 of DAIMI, Aarhus, Denmark, August 2002. University of Aarhus, 2002: 1-20

[5] Milner R. A Calculus of Mobile Process, Part I/Part II. Journal of Information and Computation, 1992, 100(1): 1-77

[6] Milner R. The Polyadic Pi-calculus: A Tutorial. Technical Report, ECS-LFCS-91-180. Computer Science Department, University of Edinburgh, UK, October 1991

[7] Yang Dong, Zhang Shensheng. An Approach for Workflow Modeling Using pi-calculus. Journal of Zhejiang University (SCIENCE), 2003(6)

[8] Puhlmann F, Weske M. Using the Pi-Calculus for Formalizing Workflow Patterns. BPM 2005, volume 3649 of LNCS, Berlin, Springer-Verlag, 2005: 153-168

[9] Business Process Management Initiative (BPMI). Business Process Modeling Notation (BPMN), Version 1.0. May, 2004. http://www.bpmn.org/ Documents/BPMN V1-0 May 2004.pdf

[10] Kappel G, Rausch-Schott S, Retschitzegger W. Coordination in Workflow Management System-A Rule-based Approach // Connen W, Neumann G, eds. Coordination Technology for Collaborative Application - Organizations, Processes, and Agents (LNCS 1364). Springer, Berlin, Heidelberg: 99-120

[11] Mulyar N A, van der Aalst W M P. Patterns in Colored Petri Nets. BETA Working Paper Series, W P139. Eindhoven University of Technology, Eindhoven, 2005

[12] Havey M. Essential Business Process Modeling. O'Reilly, August 2005

[13] Busi N, Gorrieri R. A Petri Net Semantics for Pi-Calculus // Lee I, Smolka S, eds. Proceedings of CONCUR 95, Lecture Notes in Computer Science 962. Springer-Verlag, 1995: 145-159

[14] Devillers R, Klaudel H, Koutny M. A Petri net translation of pi-calculus terms. Technical Report Series, CS-TR-887. Published by the University of Newcastle upon Tyne, School of Computing Science, UK, January 2005

[15] 廖军，谭浩，刘锦德. 基于 Pi-演算 Web 服务组合的描述和验证. 计算机学报，2005，28(4)

[16] Russell N, ter Hofstede A H M, et al. Workflow Data Patterns. QUT Technical report, FIT-TR-2004-01. Queensland University of Technology, Brisbane, 2004

[17] Russell N, ter Hofstede A H M, et al. Workflow Resource Patterns. BETA Working Paper Series, W P127. Eindhoven University of Technology, Eindhoven, 2004

[18] Victor B, Moller F. The Mobility Workbench - A Tool for the Pi-Calculus // Proceedings of CAV'94. Stanford, Springer-Verlag, volume 818 of LNCS, 1994: 428-440