

WPDL 中的 JOIN 语义问题和分区解决方案^{*})

郝克刚^{1,3} 王斌君^{1,2} 安 贵³

(西北大学软件工程研究所 西安710069)¹ (中国人民公安大学科技系 北京102614)²

(西安协同数码股份有限公司 西安710075)³

The JOIN Semantic Issue in WPDL and the Synchronized Area Solution

HAO Ke-Gang^{1,3} WANG Bin-Jun^{1,2} AN Gui³

(Software Engineering Institute, Northwest University, Xi'an 710069)¹

(Department of Science and Technology, the Public Security University, Beijing 102614)²

(Xi'an Synchrobit Co., Ltd., Xi'an 710075)³

Abstract The semantic of JOIN in workflow process definition language WPDL is analyzed and studied. It points out that there is a problem about AND-JOIN. To solve the problem, the synchronized semantic of AND-JOIN in extended Xinpai-driven model is defined formally. It puts forward the concepts of synchronized area and asynchronized area. It gives the solution of true-and-false token rules in synchronized area and true token rules in asynchronized area, and some other issues arisen from the solution are studied systematically and completely. The constrain issues of control structures in synchronized area and asynchronized area are discussed, and its solution is put forward. Finally, the algorithm of how to find the focus point and synchronized area is given.

Keywords Workflow process definition language, WPDL, Xinpai-driven model, Synchronized area, True-and-false token, Focus point, Petri net

1 引言

workflow 管理系统(也称为过程管理系统)^[1]被称为是继数据库管理系统和人机界面管理系统之后第三次从应用中分离出来并成为一个应用系统不可或缺的子系统,它已有着广泛的应用前景,特别是在 Internet/Intranet 环境下它将成为必备的软件开发平台,吸引了大批的研究机构^[1]、专家学者^[3~5]和软件开发公司^[9~11]对其进行研究。

目前, workflow 管理系统中还有大量的问题需要认真、深入地加以研究,其中一个关键的问题是如何建立一个合理、全面、使用方便和能进行形式化验证的工作流过程模型。为此,我们在认真研究了现有各种工作流模型^[2~5]的基础上提出了工作流过程模型研究的三个层次^[6]:语言层、模型层和控制模型层。语言层研究工作流过程定义语言的表达能力以及它的语法语义等;模型层是为研究语言的表达能力以及语义等所建立的语言抽象模型,我们提出的信牌驱动模型便属于此类研究;控制抽象层是对第二层的进一步抽象,通过它可对模型的控制结构的静态和动态性质进行验证和模拟。关于它们的详细描述请参见文[8]。其中,语言层是其他两个层次模型研究的对象和基础。信牌驱动模型的研究以及采用 Petri 网所作的控制模型的研究,为扩展和完善工作流过程定义语言的语法和语义奠定了重要的理论基础^[6,7]。

工作流过程定义语言 WPDL (Workflow Process Definition Language)^[2]是 workflow 管理联盟 WfMC (Workflow Management Coalition)提出的用于工作流过程定义交流的标准界面文本。它有严格的语法和较严格的语义,是被各 workflow

管理系统的产品所能接受的标准,通过它便于在不同的工作流产品之间交换过程定义。

WPDL 基于一种工作流过程定义的元数据模型(workflow process definition meta-data model),这种模型规定了工作流过程定义中各种元素以及语义。为了描述各活动之间的执行顺序和并行关系,它引入了活动的 JOIN 和 SPLIT 元素,JOIN 和 SPLIT 又分为 AND-JOIN, XOR-JOIN, AND-SPLIT, XOR-SPLIT 四种类型。在开发 workflow 管理系统 SynchroFLOW 产品(由西安协同数码股份有限公司和西北大学软件工程研究所合作)和构造扩展的信牌驱动模型时,我们发现 WPDL 的非形式化定义有许多不清楚的语义问题,AND-JOIN 的定义就是其中之一。AND-JOIN 在 WPDL 中被定义为与进入此活动的转移相关的并行线程的合并,要求同步,而且同步合并的线程的个数依赖于前面的 AND-SPLIT 转移条件的实际执行情况。既然这里规定 AND-JOIN 是一个同步结点,但又不要它的所有输入转移都激活,那么当有多少个输入转移激活后,该结点就不再等待,可以进入执行状态呢?按 WPDL 的规定,AND-JOIN 的输入转移激活的个数是与前面的 AND-SPLIT 动态执行结果相关。这种定义不够严格,且存在一系列问题,如这里所说的“AND-JOIN 前面的 AND-SPLIT”是否有固定的所指?如何求出这些 AND-SPLIT?此时,该 AND-JOIN 应该按什么规则计算输入转移的个数以表明它可进入执行状态?如果用限制 SPLIT 和 JOIN 类型严格配对(在 WPDL 的说明中选 FULL-BLOCKED)的方法解决这个问题,自然很容易,但却大大地限制了它的实际应用。本文提出用真假信牌和同步区的概念可以解决这个问题,在一

^{*})本文受国家“十五”重点科技攻关项目资助(项目编号:2001BA107C)。郝克刚 教授,博士生导师,目前研究方向为构件技术、分布式计算和软件理论等;王斌君 副教授,目前研究方向为软件工程、工作流模型和软件理论等;安 贵 工程师,目前研究方向为工作流系统、中间件。

定意义下是放宽了配对的限制,提高了信牌驱动模型的灵活性和建模能力。

2 信牌驱动模型中的真假信牌执行规则及相应的问题

“信牌”是我国古代官府执行任务的凭证,官员只有接到信牌才能执行相应的任务。我们借助这一概念,用活动和信牌箱将 Petri 网中变迁和位子等形式化的概念变成人们容易理解的形式建立了信牌驱动模型,详见文[6,7]。为了与 WPD 相对应,在信牌驱动模型中也引入了活动的 SPLIT 类型和 JOIN 类型。信牌驱动模型的基本运行规则描述如下:

过程开始时只有一个信牌箱(称为开始信牌箱)中有信牌,而其它信牌箱中无信牌。过程运行中,当一个活动的前信牌箱中信牌数按其 JOIN 类型满足一定条件时,该活动称为可以执行的活动;当可以执行的活动执行时,就按其 JOIN 类型从前信牌箱取走一定数量的信牌,并当活动结束后按其 SPLIT 类型向后信牌箱放一定数量的信牌。直到所有活动都不可执行时过程结束。

下面分别说明各种 JOIN 和 SPLIT 类型的执行规则,实际上也就是这些语言成分的语义。按我们提出的解决方案,执行规则分为两类:一种称为真信牌规则,一种称为真假信牌规则。

2.1 真信牌规则

ALL-SPLIT:当该活动结束后,要给它所有的后信牌箱中放相应数目(数目等于相应转移的权重,以下均相同)的信牌。XOR-SPLIT:当活动结束后,要在转移条件为真的后信牌箱中按给定的先后次序选择一个信牌箱放相应数目的信牌。AND-SPLIT:当活动结束后,要在转移条件为真的多个(按规定最少一个转移条件为真)后信牌箱中放相应数目的信牌。

ALL-JOIN:只有当该活动的所有前信牌箱中都有相应数目信牌时,该活动才可以执行。当活动启动后,开始执行时从每个前信牌箱中取走相应数目的信牌。XOR-JOIN:当它的前信牌箱中只要有一个信牌箱有相应数目的信牌,它就可以执行,每执行一次就从该前信牌箱中取走相应数目的信牌。AND-JOIN:按 WPD 的规定,当它的前信牌箱中有一定数目的信牌箱有相应数目的信牌才可以执行,这些信牌箱的数目取决于与它前面的 AND-SPLIT 执行时实际给几个信牌箱中放了信牌。前面已经指出,WPD 中 AND-JOIN 的语义不够明确,不能具体地操作。为此,我们在真信牌规则中不允许出现 AND-JOIN。

2.2 真假信牌规则

信牌分为真、假信牌,活动执行也分为真执行和假执行。

ALL-SPLIT:活动真执行后,要给所有的后信牌箱放相应数目的真信牌。XOR-SPLIT:活动真执行后,要在转移条件为真的后信牌箱中按给定的先后次序选择一个信牌箱放相应数目的真信牌,其它后信牌箱中放相应数目的假信牌。AND-SPLIT:活动真执行后,在转移条件为真的后信牌箱中放相应数目的真信牌,其它后信牌箱中放相应数目的假信牌。活动假执行后,无论活动的 SPLIT 是什么类型,均在所有后信牌箱中放相应数目的假信牌。

按真假信牌规则,一个活动当其所有前信牌箱中都有信牌(真信牌或假信牌),且各前信牌箱中信牌的个数不小于相应转移的权数时,它才可以执行。每次执行(真执行或假执行)均从所有前信牌箱中取走规定数目的信牌(等于相应转移的

权数)。真执行或假执行按下述情况决定。

ALL-JOIN 活动:当这些规定数目的前信牌全是真信牌时,该活动可真执行;当其中有一个是假信牌时,该活动假执行。XOR-JOIN 活动:当它的某个前信牌箱中有大于或等于相应权数的真信牌就真执行,否则就假执行。AND-JOIN 活动:若这些规定数目的信牌全是假信牌则假执行,否则真执行。所有活动,不论是手动活动还是自动活动,它的假执行都自动执行,而且除去取、放信牌的动作外不做任何动作(假执行)。

从以上两种规则的定义可以看出它们是对过程的并行成分的两种不同的语义解释。那么,在工作流过程定义中如何选择?是否可以只选择其中之一?还是两者都需要?

首先分析用真假信牌所带来的问题。例如,下面是一个零件组装的例子(图1),A、B 表示两个生产零件的活动,C 表示零件组装的活动。如果用真信牌方法描述非常方便。当 A (B) 生产一个合格零件就向 a1 (b1) 放一个信牌,否则就向 a2 (b2) 放一个信牌。C 是组装活动,当它的前信牌箱(a1和 b1)都有真信牌时表示有两个合格的零件,它才执行,否则等待。

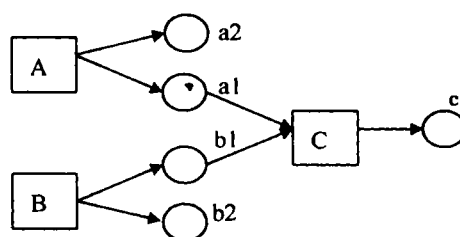


图1 零件组装的例子

但是,如果用真假信牌的方法描述:当 A (B) 生产一个合格零件就向 a1 (b1) 放一个真信牌,向 a2 (b2) 放一个假信牌;否则就向 a1 (b1) 放一个假信牌,向 a2 (b2) 放一个真信牌。当 C 的前信牌箱(a1和 b1)都有真信牌时,它才真执行;一真一假或两个都假,则假执行。这样的描述,不仅在 c1 中产生了大量的假信牌,增加了系统的负担,更重要的是真假信牌的执行产生了错误的语义:当 A 依次产生一个正品和一个次品,而 B 依次产生一个次品和一个正品时,C 应该正确地组装一次。但按真假信牌的执行规则,C 将连续两次假执行。

用真假信牌的方法很难描述上述类似的问题。问题的症结主要是真假信牌不能任意使用,真假信牌的引入主要是为了解决 AND-JOIN 的同步问题。如果不在 AND-JOIN 和与它对应的 AND-SPLIT 之间的范围内,使用真假信牌就会产生不正确的结果。同样,用真信牌的方法又很难描述另一些问题。

3 分区的解决方案

为了保持模型的充分表达能力,我们选择了两种规则综合使用的方案。把整个过程分为同步区和非同步区。真假信牌规则在同步区中使用,真信牌在非同步区中使用。

直观地讲,所谓过程中的同步区,即 AND-JOIN 结点与对应的前驱 AND-SPLIT 结点之间的区域。这里首先要确定 AND-JOIN 结点对应的前驱 AND-SPLIT 结点。如果限定 X-JOIN 结点(X 为 ALL、XOR 或 AND)和 X-SPLIT 结点之间严格配对而且是相互无嵌套,问题就非常简单,但是这种方案限制了模型的表达能力。所以我们没有做这种限制,而是通过定义和寻找 AND-JOIN 结点的聚焦点来确定其唯一的前驱 AND-SPLIT 结点(详见下面严格的定义)。由 AND-JOIN 及

其聚焦点所确定的区域称为同步区。

注意:虽然任何 AND-JOIN 结点的聚焦点是唯一的,但是反之不然,即不同结点的聚焦点可能是相同的,也就是说同步区可能重叠覆盖。例如图2:B和D的聚焦点都是A,而不是C。同步区A-D覆盖了同步区A-B。

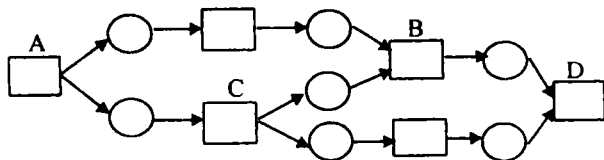


图2 聚焦点和同步区

下面对上述概念给出严格的形式定义。

定义1(扩展的信牌驱动模型) 多元式 $XP = (H, X, F, X_0, W, H_{SPLIT}, JOINH)$ 称为扩展的信牌驱动模型,其中: H 表示活动集合。 X 表示信牌箱集合。 $F \subseteq X \times H \cup H \times X$, 称为转移;对于活动 $b \in H$, 与 b 的输出转移相联系的后信牌箱称为 b 的后信牌箱, 与 b 的输入转移相联系的前信牌箱。 $X_0 \in X$ 是唯一的信牌箱, 它不是任何活动的后信牌箱, 称为开始信牌箱。 $W: F \rightarrow N$ 称为转移的权重。 H_{SPLIT} 是 H 的一个划分 $\{H_{ALL}, H_{AND}, H_{XOR}\}$, 即 $H_{ALL} \cap H_{AND} \cap H_{XOR} = \emptyset, H_{ALL} \cup H_{AND} \cup H_{XOR} = H$ 。 $JOINH$ 是 H 的另一个划分 $\{H_{ALL}, H_{AND}, H_{XOR}\}$, 即 $H_{ALL} \cap H_{AND} \cap H_{XOR} = \emptyset, H_{ALL} \cup H_{AND} \cup H_{XOR} = H$ 。

定义2(路径和非循环路径) 设 $XP = (H, X, F, X_0, W, H_{SPLIT}, JOINH)$ 为扩展的信牌驱动模型, 令 $L = a_1, a_2, \dots, a_n$ 是一个序列, $a_i \in X \cup H (i = 1, 2, \dots, n)$ 。如果 $(a_i, a_{i+1}) \in F (i = 1, 2, \dots, n-1)$, 则称 $a_1, a_2, \dots, a_n$ 是 XP 的一条路径 L , 且称 $a_i \in L (i = 1, 2, \dots, n)$, L 的长度为 n 。 $\forall i, j = 1, 2, \dots, n$, 如果 $i \neq j \Rightarrow a_i \neq a_j$, 则称 $a_1, a_2, \dots, a_n$ 是 XP 的一条非循环路径。如果存在 XP 的一条路径 $L: a_1, a_2, \dots, a_n$, 其中 a_1 是开始信牌箱结点, 则称 a_n 为可达的。

为了以下讨论方便, 我们假定 XP 是一个连通图, 且任何 $h \in H$ 均是可达的。

定义3(必经结点) 设 $XP = (H, X, F, X_0, W, H_{SPLIT}, JOINH)$ 为扩展的信牌驱动模型, $\forall B \in H$, 称 $A \in H$ 是 B 的必经结点, 当且仅当对任何从开始结点到 B 结点的非循环路径 L , $A \in L$ 。

定义4(两点之间的距离) 设 $A, B \in H$, 则 A 和 B 之间所有路径中长度最短的路径长度称为 A 和 B 之间距离。

定义5(聚焦点) 设 $XP = (H, X, F, X_0, W, H_{SPLIT}, JOINH)$ 为扩展的信牌驱动模型, $B \in ANDH$, 如果 A 是 B 的必经结点中和 B 的距离最短的结点, 则称 A 是 B 的聚焦点。

从聚焦点的定义可以推论出: 如果 A 是 B 的聚焦点, 则从结点 A 到结点 B 的非循环路径中都不再含有其它必经结点。

定理1(聚焦点的唯一性) 设 $XP = (H, X, F, X_0, W, H_{SPLIT}, JOINH)$ 为扩展的信牌驱动模型, $B \in ANDH$, 如果 $\exists A \in ANDH$ 是 B 的聚焦点, 则 A 是唯一的。

证明 假定 B 的聚焦点不唯一, 有 $A1 \neq A2$ 都是 B 的聚焦点。按定义有非循环路径 $L1: X_0, \dots, A1, \dots, B$ 和 $L2: X_0, \dots, A2, \dots, B$, 且 $A1 \in L2(A2, B)$, $A2 \in L1(A1, B)$ 。由于 $A2$ 是 B 的必经结点, 而 $A2 \in L1(A1, B)$, 则有 $A2 \in L1(X_0, A1)$ 。由 $L1(X_0, A2)$ 和 $L2(A2, B)$ 连接组成的路径 $L3$ 显然是 X_0 到 B 的非循环路径。因为 $A1 \in L1(X_0, A2)$, $A1 \in L2(A2, B)$, 所以

$A1 \in L3$ 。这与 $A1$ 是 B 的必经结点相矛盾, 本定理得证。

定义6(同步区) 设 $XP = (H, X, F, X_0, W, H_{SPLIT}, JOINH)$ 为扩展的信牌驱动模型, $B \in ANDH$, A 是 B 的聚焦点, 则从 A 出发到达 B 的所有路径结点的集合称为结点 B 的同步区。将过程定义中所有 AND-JOIN 活动的同步区的总合称为过程的同步区。

4 引出的问题和解决办法

真假信牌方法的基本思路是解决 AND-JOIN 的同步问题: 从聚焦点 A 传来的信牌, 经过同步区, 不管变得是真还是假, 在到达结点 B 的前信牌箱后进行同步, 使结点 B 开始执行。如果在同步区某条路径的执行中产生新的真假信牌(多流)或丢失某个真假信牌, 则会产生同步混乱, 使结点 B 无法正常执行。为了使 AND-JOIN 类型的结点能正常工作, 下面对同步区中各种控制结构可能引出的问题进行详细的分析, 指出使用的限制并提出相应的解决办法。

4.1 静态活动问题

静态活动是一种特殊的活动。当它的执行条件一旦满足并执行后, 就可以无限地再次执行。静态活动的多次执行可以不断地向后信牌箱产生信牌, 形成多流。但是, 由于它只处在同步区的某一个并行分支上, 同步区的 AND-JOIN 只能同步它的一次执行。在同步区中使用静态活动会产生同步混乱, 所以禁止在同步区中使用静态活动。

4.2 循环问题

为了在同步区中正常地使用循环, 我们还需对真假信牌规则做适当的修改和限制。

首先, 我们来分析循环尾活动后的转移。将其中的用于循环反馈的转移线标识为 goback 线, 规定 goback 线的权重为 1。如果循环尾活动是 AND-SPLIT 类型, 为保证不产生多流, 在一次执行中此循环尾所有 goback 转移只能有一个为真。而且, 一旦某 goback 转移为真, 其它的转移就不能为真。如果循环尾活动是 XOR-SPLIT 类型, 显然不成问题, 自然不允许循环尾活动是 ALL-SPLIT 类型。

此外, 为保证不产生多余和有害的假信牌, 还需对真假信牌规则做适当的修改, 规定对循环尾活动。如果是真执行, 在选中的转移条件为真的后信牌箱中放真信牌, 其它后信牌箱中不放任何信牌。如果是假执行, 只在非 goback 转移的后信牌箱中放假信牌, goback 转移的后信牌箱中不放信牌。

最后, 为保证循环正常执行, 还需在同步区中引入异步 OR-JOIN 类型的活动, 循环头活动就应是异步 OR-JOIN 类型。异步 OR-JOIN 活动的执行规则如下: 当它的前信牌箱中只要有一个信牌箱有一定数目(等于相应转移的权重)的信牌, 它就可以执行。每执行一次, 就从该前信牌箱中取走相应数目的信牌。

注意: 循环头可以不是活动而是信牌箱, 而信牌箱的运行机制类似于异步 OR-JOIN 活动的功能。

4.3 异步 OR-JOIN 问题

在同步区中引入异步 OR-JOIN 类型的活动以及在不同步区中出现类似于异步 OR-JOIN 活动功能的多输入弧的信牌箱结构(图3), 均会带来一些异步 OR-JOIN 问题。

异步 OR-JOIN 将在所处的并行分支上产生先后的多流, 使得在同步区的 AND-JOIN 结点处不能进行正确的同步。为了保持模型的一定建模能力, 也能保证其正确执行, 我们允许同步区中存在上述结构, 但必须在后面活动的输入弧上加适当的权重, 以保证正确地同步。

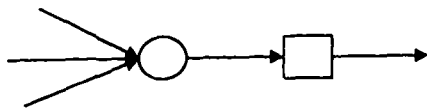


图3 异步 OR-JOIN 问题

4.4 前出问题

当有一条线由同步区中的一个活动转移到该同步区的开始活动前面的某个信牌箱时,称为前出情况。图4所示是一个前出情况的例子,由 B 到 E 是同步区。而 A 后的 goback 线指向同步区的前面。按前述的 goback 规则,如果 goback 转移条件为真,则 a 后的信牌箱将不放信牌,使 E 结点不能正确同步。解决办法是:如果 goback 线指向同步区的前面,则此线不标注为 goback 线,也不按 goback 转移规则处理,而按同步区中的一般节点处理。

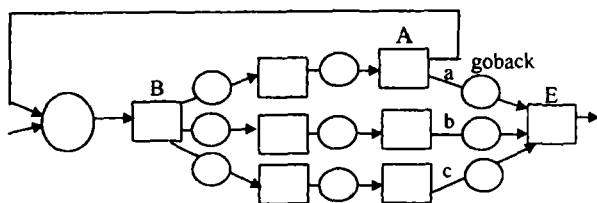


图4 前出问题

4.5 后入问题

当从同步区后继的某个活动转向该同步区中时,形成后入情况。图5是一个后入情况的例子。我们规定,不允许后入情况出现,因为它将无法在同步区的 AND-JOIN 节点 E 形成同步。其它的非本同步区后面的转移进入本同步区是允许的,因为这将构成一个更大的同步区。

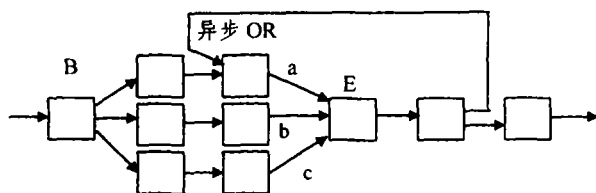


图5 后入问题

5 求焦点和同步区的算法

求同步区的关键是求与 AND-JOIN 节点配对的聚焦点,由 AND-JOIN 节点与其对应的聚焦点所围成的区域称为同步区。下面首先重点讨论求一个 AND-JOIN 节点的聚焦点的基本思想和算法,由计算聚焦点的算法可直接求出相应的同步区。

算法1 (聚焦点)

1. 求出从开始结点到所有 AND-JOIN 结点的无循环路径。
2. 求出每个 AND-JOIN 的必经结点及它们与相应 AND-JOIN 结点的最短距离。
3. 每个 AND-JOIN 结点的距离最短的必经结点即为该结点的聚焦点。

上面是一个粗略的聚焦点算法。为了使其可计算和程序化,下面给出一个较细致和高效的计算聚焦点的算法。我们称模型中结点到开始结点的距离为结点的层数。用下述算法可

以求出模型中所有结点的层数。

算法2.1 (求出模型中所有结点的层数)

1. 模型中的开始结点层数属性为1,并将其放入待处理队列中;
2. 从待处理队列中取出一个结点 a,考虑模型中对应元素的所有直接后继元素。如果该后继元素尚未赋予层数,则赋予层数属性为 a 的层数+1,并且放入待处理队列中;如果该后继元素已赋予层数,则不再赋予,也不再进队列;
3. 如果待处理队列非空转2,否则转4;
4. 结束。

算法2.2 (聚焦点和同步区)

开始时令集合 A(聚焦点)和集合 B(同步区)为空。

1. 将给定的 AND-JOIN 结点加入集合 A 中;
2. 将集合 A 中层数最大的结点取出并加入集合 B 中,并按模型求出它的父结点。如果该父结点不在集合 A ∪ B 中,就将其父结点加入集合 A,否则不加入;
3. 若集合 A 中只有一个元素,则转4,否则转2;
4. 集合 A 中的元素就是所求的聚焦点。将它加入集合 B,则集合 B 中的元素构成了该 AND-JOIN 结点和它的聚焦点之间的同步区。结束。

利用算法2.2可以计算出信牌驱动模型所给出的过程定义中所有 AND-JOIN 的聚焦点和所有的同步区。

最坏情况下,算法2.1是将信牌驱动模型的过程定义中所有活动访问一遍,它的时间复杂度为 $O(n)$;算法2.2的复杂度也为 $O(n)$ 。所以,计算信牌驱动模型的过程定义中一个聚焦点和同步区的复杂度为 $O(n^2)$,计算所有聚焦点和同步区的复杂度为 $O(n^3)$ (算法2.1只运行一次)。

结束语 本文在前期工作流模型研究的基础上对 WPD 中有争议的 AND-JOIN 语义进行了深入探讨,在信牌驱动模型层中提出了聚焦点、同步区、非同步区和真假信牌等概念,并给出了一个求 AND-JOIN 结点聚焦点的算法,从而彻底解决了 AND-JOIN 语义和正确执行问题。用同步区的概念与用配对的概念相比,使得模型适应面更广,使用起来更加方便和直观。在同步区中使用真假信牌的概念,明确了同步区中各种控制活动的动态变化规则和特性。本文的研究成果已在当前所开发的产品(工作流管理系统 SynchroFLOW)中得到了具体的应用。

参考文献

- 1 WfMC. The Workflow Reference Model. <http://www.aiim.org/wfmc>
- 2 WfMC. Workflow Process Definition Interchange. <http://www.aiim.org/wfmc>
- 3 史美林,向勇,杨光信. 计算机支持的协同工作理论与应用. 北京: 电子工业出版社,2000
- 4 范玉顺. 工作流管理技术基础. 北京: 清华大学出版社,2001
- 5 van der Aalst W M P. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 1998. 1~3
- 6 岳晓丽,杨斌,郝克刚. 信牌驱动式工作流计算模型. 计算机研究与发展, 2000, (37)12: 1513~1519
- 7 郝克刚,王斌君. 非确定 Petri 网. 小型微型计算机系统(已录用)
- 8 Wang Binjun, Hao Kegang. Three Levels of Workflow Process Model. In: Proc. International Symposium on Future Software Technology. 2001. 142~146
- 9 HP changengine Workflow. <http://www.hp.com/go/changengine>
- 10 IBM MQSeries Workflow. <http://www.software.ibm.com/ts/mqseries/workflow>
- 11 iPlanet Process Manager. <http://www.iplanet.com/>