

# 基于同步网的工作流过程变动问题研究

闫 哲, 赵 文, 袁崇义, 王立福

(北京大学计算机系, 北京 100871)

**摘 要:** 支持动态可变的特性是实际应用对工作流系统提出的要求, 已有的研究在表达能力和算法复杂度等方面存在问题. 本文以形式化的工作流模型——同步网为基础, 从逻辑和语义两个角度定义了过程变动, 继而分类描述了基本变动操作, 并讨论了变动的正确性问题. 复杂的过程变动可以被分解为若干基本变动操作的序列, 从而简化问题并且保证整个变动的正确性. 在此之上提出了一种新的管理过程变动的方法, 即以同步器为中心来管理过程实例的迁移问题.

**关键词:** 工作流; 过程变动; 同步网; 过程迁移

**中图分类号:** TP311 **文献标识码:** A **文章编号:** 0372-2112 (2006) 02-0226-06

## Research on Workflow Process Change Based on Synchronization Net

YAN Zhe ZHAO Wen YUAN Chong-yi WANG Li-fu

(Department of Computer Science and Technology, Peking University, Beijing 100871, China)

**Abstract** The feature of supporting dynamic change is required in the application of workflow management system. While some relational research fell short in terms of expressive capacity and algorithm complexity. Based on synchronization net a formal workflow model, this paper proposes a definition of process change from logic and semantic view, then classifies elementary change units, and discusses the correctness of change. A whole process change in need is considered as a sequence of those change units. Therefore the complex problem becomes easy and the correctness of change is preserved strictly. Following the description of change, a new method to manage process change is proposed, so as to resolve the migration problem of on-the-fly process instance, centered on synchronizer.

**Key words** workflow; process change; synchronization net; process migration

### 1 引言

企业中存在着大量的业务流程, 根据环境和客户的需要及时快速地调整业务流程, 已经成为企业经营能力的体现. 文献[1]中对引起流程变动的原因进行了分类概括, 从中可以看到流程变动的必要性. 工作流技术的出现旨在为业务流程提供自动化支持, 因此研究实际操作中存在哪些流程变动和它们符合的性质, 以及如何管理变动问题显得尤为重要.

一般来说, 工作流管理包含两个关键的阶段, 即过程模型的建立和过程实例的执行, 而过程变动则相应地发生在两个层次——过程模型和过程实例. 过程模型的变动将引发相关过程实例的变动, 涉及到已有过程实例的处理, 以及变动前后过程模型之间的关系等问题, 这是解决过程变动的关键. 而过程实例的变动影响到的只是当前一个实

例, 大多与异常处理相关联. 虽然用户可以根据需要进行过程修改, 但不能没有约束的随意变动, 这就需要在变动中保持过程原有的性质, 即需要考虑工作流的变动正确性问题.

虽然许多工作流项目就此问题提出了解决方案, 但因为工作流模型本身的限制, 使得处理过程变动的方法在表达能力和复杂性等方面存在问题. 本文以基于 Petri 网的工作流模型——同步网为基础, 分析了各类过程变动, 并给出了管理变动问题的方案.

### 2 同步网模型

文献[2, 3]中提出的同步网是基于 Petri 网理论用于解决工作流管理问题的建模工具. 根据抽象层次的不同, 同步网分为逻辑层、语义层, 以及执行层. 逻辑层关心过程中由哪些任务组成和这些任务之间的依赖关系; 语义层是

收稿日期: 2005-07-29 修回日期: 2005-10-20

基金项目: 国家自然科学基金 (No. 60473058); 国家重点基础研究发展规划 (973) (No. 2002CB312006)

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

在逻辑层之上为案例确定实际路径; 而执行层实施管理规则以推动过程实例的执行. 本文的研究工作就是在同步网的基础上进行的, 这里对文中用到的一些概念和定理作简要介绍.

**定义 1** 设  $PT = (P, T; F, K, W)$  为  $PT$  网, 定义库所  $syn = (T_1, T_2, (a_1, a_2))$  为同步器, 其中  $T_1, T_2 \subset T, |T_1| = m_1, |T_2| = m_2, a_1 = W(syn, t_{2j}), a_2 = W(t_{1i}, syn), 1 \leq a_1 \leq m_1, 1 \leq a_2 \leq m_2 (1 \leq i \leq m_1, 1 \leq j \leq m_2, t_{1i} \in T_1, t_{2j} \in T_2)$ .

同步器是逻辑层的核心成分, 它借助同步器中同步距离的概念, 刻画了任务之间的同步关系, 如图 1. 同步器根据  $|T_1|, |T_2|$  及  $a_1, a_2$  的取值不同体现为串、并、选择等多种同步关系, 其组合可以实现多种工作流模式<sup>[4]</sup>.

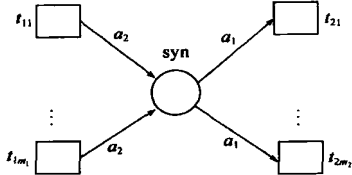


图 1 同步网示意图

**定义 2**  $WN = (P, T; F, K, W, M_0)$  称为工作流逻辑, 如果

- (1)  $(P, T; F, K, W)$  是  $PT$  网, 而且  $\forall p \in P: \cdot p \neq \emptyset \wedge p \neq \emptyset \Rightarrow p = (T_1, T_2, (a_1, a_2))$ .
- (2)  $\langle \cdot \rangle = \{(t, t') \mid t, t' \in T \wedge \exists p \in P: t \in \cdot p \wedge t' \in p\}$ , 则  $\langle \cdot \rangle$  满足:  $\langle \cdot \rangle \cap \langle \cdot \rangle^2 = \emptyset$  且  $(T, \langle \cdot \rangle)$  是连通图.
- (3)  $\forall p \in P: (\cdot p = \emptyset \Rightarrow M_0(p) = 1) \wedge (\cdot p \neq \emptyset \Rightarrow M_0(p) = 0)$ .

工作流逻辑表现为可能存在冲突的 Petri 网系统.

设  $WN = (P, T; F, K, W, M_0)$  为工作流逻辑, 如果入口库所中的托肯经传递、复制及合并(同步器)能流动到一个终点库所, 而且与冲突消解方案无关, 则称  $WN$  具有畅通性.

为了检验逻辑的畅通性, 文献 [2-3] 中采用图形化简的方法, 提出了 8 条化简规则, 并且证明这些规则不改变过程原来的畅通性.

**定理 1**

工作流逻辑如果使用文献 [2] 中 8 条规则能够化简为 1 个库所, 则认为是畅通的.

本文下一节的证明依赖于部分化简规则, 因此对它们作简单介绍.

**化简规则 1**

同步器  $p_1 = (T_1, T, (a, a_1))$  和  $p_2 = (T, T_2, (b_1, b))$  满足  $a_1 = b_1$ , 则可以化简为  $p = (T_1, T_2, (a, b))$ .

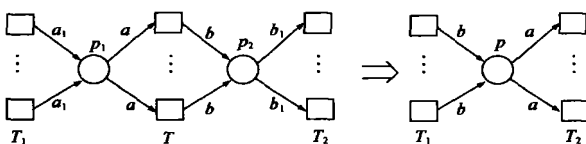


图 2 化简规则 1

**化简规则 2**

同步器  $p = (\{t_1\}, \{t_2\}, (1, 1))$ , 只要  $t_1 \cdot \cap t_2 = \{p\}$  即可以消去  $p$ , 将  $t_1$  和  $t_2$  合并为  $t, t = t_1 \cup t_2 - \{p\}, t = t_1 \cup t_2 - \{p\}$ .

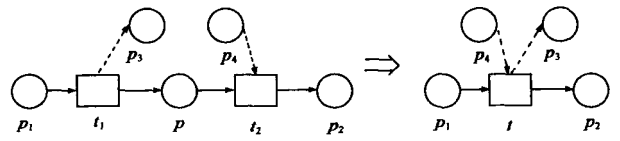


图 3 化简规则 2

**化简规则 3**

若变迁  $t$  和库所  $p_1, p_2$  满足  $t \in p_1 \cdot \wedge \{t\} = \cdot p_2$  且  $p_2 \neq \emptyset \vee p_1 \cdot = \{t\}$ , 则  $p_1$  和  $p_2$  可以完全合并为  $p, \cdot p = \cdot p_1, p \cdot = p_1 \cdot \cup p_2 \cdot - \{t\}$ . 若  $p_1$  中有托肯,  $p$  也有.

图 4 化简规则 3

**定义 3**  $\Sigma = (P, V, T; F, K, W, R, W_r, M_T, M_0)$  为业务流的工作流语义, 如果

- (1)  $(P, T; F, K, W, M_{op})$  是工作流逻辑, 其中  $M_{op} = M_0$  上的限制.
- (2)  $V$  是业务表单的显性变量集,  $R$  和  $W_r$  为  $T$  对  $V$  的读写关系.
- (3)  $\forall t \in T: M_T(t) = guard(t) + body(t)$ , 即  $M_T$  包含变迁的进入条件和执行主体.

工作流语义中引入变量, 通过变量的取值来消解工作流逻辑中的冲突, 从而为过程实例确定一条唯一的执行路径. 图 5 表示通过加入  $V$  元  $x$  在选择连接处选取一条合适的分支.

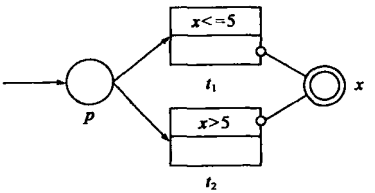


图 5 工作流语义中的  $V$  元

设  $\Sigma = (P, V, T; F, K, W, R, W_r, M_T, M_0)$  为工作流语义, 如果无论变量的取值都能成功消解工作流逻辑的冲突, 则称  $\Sigma$  具有正确性.

**3 过程变动操作**

为降低问题的复杂性, 本文将过程变动看作基本变动的序列, 通过对基本变动的研究来获得过程变动的性质. 根据同步网的层次结构, 处理变动问题首先分为两类, 即过程逻辑变动和过程语义变动.

**3.1 过程逻辑变动**

本节在组成过程的活动、活动间的连接关系等方面分析变动问题. 由于同步器用以刻画活动之间的同步关系, 过程中活动发生变动会直接影响同步器的描述, 而在逻辑网中一个活动涉及到前后两个同步器, 所以给出如下的定

义。

定义 4 四元组  $CIU = (SP_0, EP_0, SP_N, EP_N)$  称为过程逻辑变动单元, 其中  $SP_0$  和  $EP_0$  是变动之前的起止同步器,  $SP_N$  和  $EP_N$  是变动之后的起止同步器。

本文以同步器为中心描述变动, 可以清晰地刻画变动所涉及的活动。

定义 5 设  $CLU = (SP_0, EP_0, SP_N, EP_N)$  为过程逻辑变动单元, 如果该变动能够保持 workflow 逻辑的畅通性, 则称  $CLU$  具有正确性。

即如果变动前 workflow 逻辑是畅通的, 而变动后逻辑依然畅通, 则此变动是逻辑正确的。

变动 1-1 改变同步关系

图 6 表示通过同时改变相邻两个同步器的两端弧的加权值 (由  $m$  变成了  $m'$ ), 以改变中间活动集合  $T$  的同步关系。这种变动描述为  $((T_1, T, (m, a)), (T, T_2, (b, m)), (T_1, T, (m', a)), (T, T_2, (b, m')))$ 。

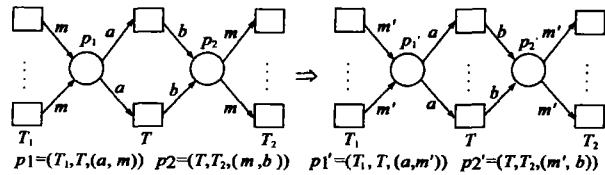


图 6 改变同步关系

根据同步网中给出的化简规则 1,  $(T_1, T, (m, a))$  和  $(T, T_2, (b, m))$  可以化简为  $(T_1, T_2, (b, a))$ 。经过如上变动,  $(T_1, T, (m', a))$  和  $(T, T_2, (b, m'))$  仍然可以化简为  $(T_1, T_2, (b, a))$ , 只要保证  $m' \geq m$  即可, 因此不改变原来过程的畅通性。

变动 1-2 增加分支活动

图 7 表示在两个同步器之间添加新活动 ( $t_n$ ), 以增加可能的分支。这种变动描述为  $((T_1, \{t\}, (1, 1)), (\{t\}, T_2, (1, 1)), (T_1, \{t, t_n\}, (1, 1)), (\{t, t_n\}, T_2, (1, 1)))$ 。

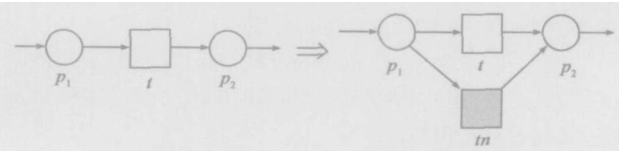


图 7 增加分支活动

根据同步网中给出的化简规则 3  $(T_1, \{t\}, (1, 1))$  和  $(\{t\}, T_2, (1, 1))$  可以化简为库所  $p$ 。经过变动得到的同步网, 根据化简规则 1,  $(T_1, \{t, t_n\}, (1, 1))$  和  $(\{t, t_n\}, T_2, (1, 1))$  仍然可以化简为库所  $p$ , 因此不改变原来过程的畅通性。

变动 1-3 增加串行活动

图 8 表示通过改变已有同步器和增加新的同步器, 在两个活动之间插入新的活动 ( $t_n$ )。这种变动  $SP_0 = EP_0$ , 可描述为  $((T_1, \{t\}, (1, 1)), (T_1, \{t\}, (1, 1)), (T_1, \{t_n\}, (1, 1)), (\{t_n\}, \{t\}, (1, 1)))$ 。

根据同步网中的化简规则 3  $(T_1, \{t_n\}, (1, 1))$  和  $(\{t_n\}, \{t\}, (1, 1))$  可化简为  $(T_1, \{t\}, (1, 1))$ , 即变换为原来的形式, 因此不改变原来过程的畅通性。

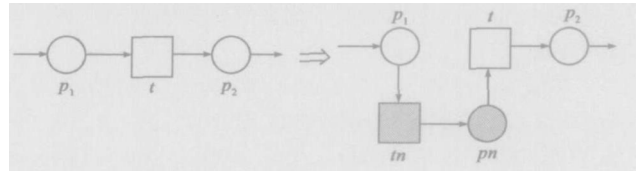


图 8 增加串行活动

变动 1-4 增加同步活动

图 9 表示在两个同步器之间添加新活动, 以增加可能的分支, 这是前面“变动 1-2”的普遍形式。这种变动描述为  $((T_1, T, (m, a)), (T, T_2, (b, m)), (T_1, T + \{t_n\}, (m, a)), (T + \{t_n\}, T_2, (b, m)))$ 。

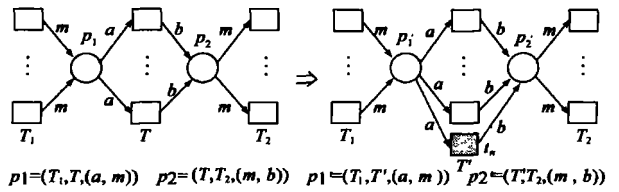


图 9 增加同步活动

根据同步网中给出的化简规则 1,  $(T_1, T, (m, a))$  和  $(T, T_2, (b, m))$  可以化简为  $(T_1, T_2, (b, a))$ 。经过如上变动,  $(T_1, T + \{t_n\}, (m, a))$  和  $(T + \{t_n\}, T_2, (b, m))$  化简为  $(T_1, T_2, (b, a))$  仍然成立, 因此不改变原来过程的畅通性。

变动 1-5 移除串行活动

图 10 表示移除两个同步器中间的活动  $t$  这种变动  $SP_N = EP_N$ , 可描述为  $((T_1, \{t\}, (1, 1)), (\{t\}, T_2, (1, 1)), (T_1, T_2, (1, 1)), (T_1, T_2, (1, 1)))$ 。

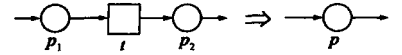


图 10 移除串行活动

根据同步网中的化简规则 3 可以直接进行上述化简, 不改变原来过程的畅通性。

变动 1-6 移除同步活动

图 11 表示移除两个同步器中间的活动, 这是前一种变动的普遍形式。这种变动描述为  $((T_1, T, (m, a)), (T, T_2, (b, m)), (T_1, T - \{t\}, (m, a)), (T - \{t\}, T_2, (b, m)))$ 。

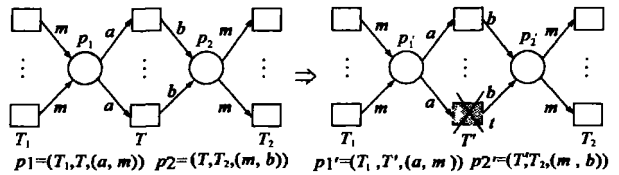


图 11 移除同步活动

根据同步网中给出的化简规则 1,  $(T_1, T, (m, a))$  和  $(T, T_2, (b, m))$  可化简为  $(T_1, T_2, (b, a))$ 。经过如上变动,  $(T_1, T - \{t\}, (m, a))$  和  $(T - \{t\}, T_2, (b, m))$  化简为  $(T_1, T_2, (b, a))$  仍然成立, 因此不改变原来过程的畅通性。

(b, a) 仍然成立, 只要保证  $\Pi^{-}\{t\} | > = m$  即可, 因此不改变原来过程的畅通性.

### 3.2 过程语义变动

语义变动关注用于消解冲突的 V 元和变迁上的执行条件, 因此其定义主要包含 V 元及其读写关系.

定义 6 五元组  $CSU = (Var, Ra, Wra, Rd, Wrd)$  称为过程语义变动单元, 其中 Var 是 V 元变量, Ra 和 Wra 是针对 Var 添加的读写关系集合, Rd 和 Wrd 是针对 Var 除去的读写关系集合.

定义 7 设  $CSU = (Var, Ra, Wra, Rd, Wrd)$  为过程语义变动单元, 如果该变动能够保持消解逻辑冲突的性质, 则称 CSU 具有正确性.

即如果变动前过程中存在的冲突均被成功消解, 变动后冲突仍然均被消解, 则此变动是语义正确的.

#### 变动 2-1 增加 V 元

图 12 表示增加 V 元, 用于消解冲突. 该变动可描述为  $(y, \{(t1, y), (t2, y)\}, \varnothing, \varnothing, \varnothing)$ .

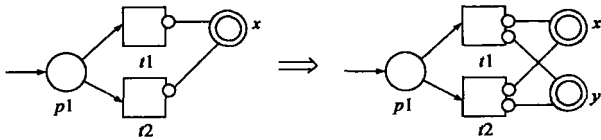


图 12 增加 V 元

#### 变动 2-2 改变读写关系

图 13 表示改变 V 元的读写关系, 根据需要调整显性数据的传递方式. 该变动可描述为  $(x, \varnothing, \{(t2, x)\}, \varnothing, \{(t1, x)\})$ .

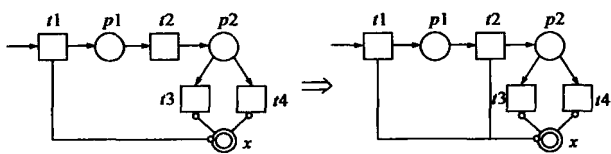


图 13 改变读写关系

#### 变动 2-3 删除 V 元

图 14 表示移除 V 元及相应的读写关系. 该变动可描述为  $(y, \varnothing, \varnothing, \{(y, t1), (y, t2)\}, \varnothing)$ .

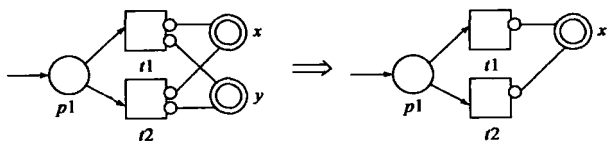


图 14 删除 V 元

语义变动的正确性取决于相关活动的执行条件能否保证按照逻辑网中给定的同步关系执行.

### 3.3 复合变动方式

根据前文所述, 过程变动是基本变动的序列.

定义 8 工作流过程变动定义为  $WFC = cu_1 cu_2 \dots cu_n$ , 其中  $cu_i (1 \leq i \leq n)$  是 CLU 或者 CSU, 出现顺序表示作用于工作流过程的操作顺序.

下面以改变活动的执行顺序为例来描述过程变动.

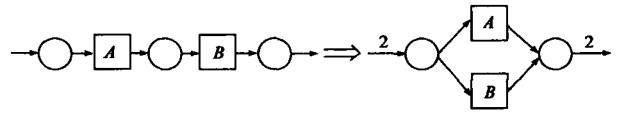


图 15 过程变动示例

图 15 表示将顺序执行的活动 (A 和 B) 改变为并行执行的活动. 这种变动可以看作如下基本变动操作的执行序列:

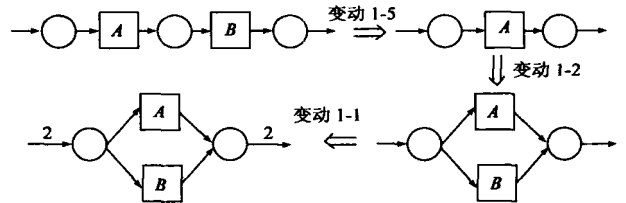


图 16 过程变动执行序列

图 16 给出了过程变动所包含的一系列基本变动操作, 开发人员可以使用它们来修改原有的业务流程, 并且保证过程变动的正确性.

## 4 过程变动管理

在工作流管理系统中, workflow 模型对应若干 workflow 实例. 在某一时刻有的实例已经执行完成, 有的实例正在执行. 如果 workflow 模型发生改变, 需要采取某种机制使得未完成的工作流实例仍能正确执行. Aalst 给出了 5 种可能的方案<sup>[1]</sup>:

- (1) 终止执行: 流程实例终止执行并归档.
- (2) 回滚重新启动: 流程实例终止执行, 回滚到开始位置, 然后按照新的模型定义重新执行.
- (3) 继续执行: 已有流程实例按照原来的模型执行, 而新启动的流程按照新的模型执行, 即“老人老办法, 新人新办法”.
- (4) 迁移到新定义: 未完成的流程实例迁移到新的模型定义中, 继而按照新定义继续执行. 因为过程变动中可能移除某些活动, 或者改变活动执行顺序, 因此判断某一时刻流程实例是否能够进行迁移是首要的问题. 如果不能迁移, 流程实例还需按照原有的模型定义继续执行, 直到能够迁移为止. 原模型和新模型之间的状态对应关系决定迁移过程的进行.
- (5) 临时性措施: 对于过程实例的变动, 通过外部干预使得实例继续执行.

相比较而言, 继续执行和迁移到新定义的方法开销小, 效率高, 在实际工作中经常采用. 继续执行的方法需要维持模型的不同版本信息, 并且记录过程实例与模板版本的对应关系. 而迁移到新定义的方法更加灵活, 能迅速体现出过程变动的效果, 是本节要解决的主要问题. 这给 workflow 管理系统提出了更高的要求, 除了管理过程模型的不同版本之外, 还需要检查实例是否可以迁移, 以及解决怎

样迁移的问题. 文献 [5] 给出了通过计算变动区域来判断能否迁移实例的方法, 但是算法比较复杂, 执行效率较低. 文献 [6] 借用继承的思想对迁移的前提条件和状态映射给出了解决办法, 但适用性受到限制.

本文对过程变动的管理主要关注逻辑变动, 给出了一种新的方法, 即以同步器为研究对象, 通过分析发生变动的同步器来确定能否迁移以及如何迁移过程实例. 在过程定义发生变动后, 当过程实例执行至某同步器时, 由 workflow 管理系统来判断能否进行迁移, 而判断的依据就是过程变动集合. 下面给出验证规则, 对于每个 CLU 检查是否满足某一条验证规则.

- (1)  $SP_0$  没有执行
- (2)  $EP_0$  已经执行完成
- (3)  $SP_0$  中有 token, 同时 token 达到容量数
- (4)  $EP_0$  中有 token, 同时 token 达到容量数

一旦所有 CLU 都满足上述的某一条验证规则, 则允许将该过程实例迁移到新的定义中. 迁移时需要考虑状态的映射, 即原来过程实例中的 token 如何转移到新的实例中. 对于过程变动集合 WFC 之外同步器上的 token 可以原封不动的转移到新的过程实例中对应同步器上. 对于 WFC 之内同步器上的 token, 则由旧同步器转换到新同步器上, 即由  $SP_0$  上的 token 迁移至  $SP_N$ , 由  $EP_0$  上的 token 迁移至  $EP_N$ . 对于新增同步器的情况, 将  $SP_0$  (即  $EP_0$ ) 上的 token 迁移至  $SP_N$ . 对于删除同步器的情况, 将  $SP_0$  和  $EP_0$  上的 token 迁移至  $SP_N$  (即  $EP_N$ ). 转换时还需要考虑 token 数目的调整, 由源同步器的容量数目转换为目标同步器上的容量数目, 以保证新的过程实例顺利执行. 下图表示符合规则的过程实例迁移到新模型中的各种情况.

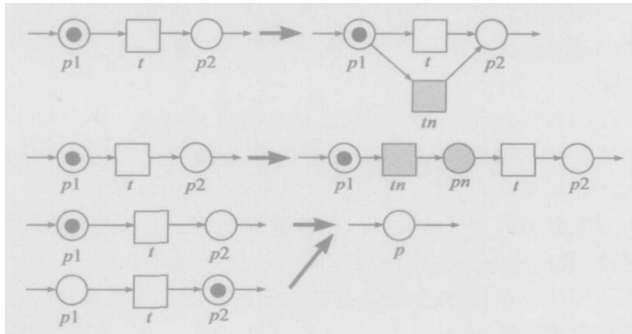


图 17 过程实例迁移

此方法以同步器为中心, 充分利用了 Petri 网的局部确定性原理, 不需要对新旧过程整体考虑以计算变化区域, 而且适用性广泛.

## 5 相关工作

目前 workflow 领域存在若干种处理变动的方法, 其中绝大部分依赖于各自的工作流模型, 从而受到模型表达能力的影响. 文献 [7] 提出了 workflow 变动的框架, 针对 workflow 的若干方面 (过程、信息、操作等) 描述了变动的基本操作, 并

通过 workflow 结构和执行状态给出了变动的正确性约束. 由于文中变动所依赖的模型类似于活动图, 因此活动之间的关系刻画起来比较复杂, 在定义修改操作时没有明确给出活动之间的关系.

文献 [8] 提出了一种 workflow 演化方法, 采用特定 Petri 网作为描述 workflow 的模型, 提出变动区域作为变动的关注点, 这样变动就体现在子网之间的替换, 并且通过 workflow 执行序列的合法性描述了变动的正确性. 方法中的变动区域很直观, 但是不能通过程序自动产生.

文献 [5] 运用变动区域的思想, 明确区分了静态、动态和最小变动区域, 并给出了自动产生变动区域的算法. 但是算法的复杂度相比文中给出的方法较高, 因为文中对变动的描述直接反映了变动区域.

本文提出的方法建立在同步网基础之上, 而同步网基于传统的 Petri 网理论, 具有局部确定性、形式化的描述和分析方法; 同时同步网的提出来源于实践, 层次化的方法使得对过程的表达能力很强. 因此, 相比其他方法本文中的方法更简单, 并且可以由程序自动完成.

## 6 总结

workflow 的过程变动问题由来已久, 在实际应用中用户不断提出变动的需求, 如何使用形式化的方法刻画过程变动, 并对过程变动进行有效的管理是重点研究的问题. 本文以 workflow 的同步网模型为基础, 分类描述了基本变动操作, 同时证明了这些变动的性质, 并且在此之上提出了一种新的管理过程变动的方法. 目前这部分工作已经在基于同步网的工作流管理系统中得到了应用. 本文中的方法还存在不足, 如对数据部分的变动管理没有涉及等. 下一步的工作包括完善过程中语义变动的管理.

## 参考文献:

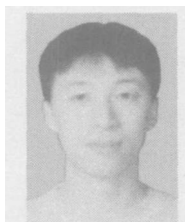
- [1] W M P van der Aalst Dealing with workflow change Identification of issues and solutions [J]. Comput Syst Sci & Eng, 2000, 15(5): 267-276
- [2] Chong Yijuan. Principals and Application of Petri Nets [M]. Publishing House of Electronics Industry, 2005. 213-258
- [3] Wen Zhao. Research on Workflow Modeling and Model Verification Technique [D]. Beijing Peking University, 2002 (in Chinese)
- [4] W M P van der Aalst A H M ter Hofstede B K iepusze-wskij A P Barros Workflow Patterns [R]. Eindhoven Eindhoven University of Technology, 2000
- [5] W M P van der Aalst Extenuating the dynamic change bug A concrete approach to support workflow change [J]. Information Systems Frontiers, 2001, 3(3): 297-317.
- [6] W M P van der Aalst How to handle dynamic change

- and capture management information: an approach based on generic workflow models [ J ]. *Comput Syst Sci Eng*. 2001, 16( 5 ): 295- 318
- [ 7 ] M Reichert P Dadam. ADEPTflex-supporting dynamic changes of workflows without losing control [ J ]. *Journal of Intelligent Information Systems* 1998, 10( 2 ): 93 - 129
- [ 8 ] C Ellis K Keddara G Rozenberg. Dynamic change within workflow systems [ A ]. *Proc Conf on Organizational Computing Systems* [ C ]. Milpitas USA, August 1995
- [ 9 ] Stefanie Rinderle Manfred Reichert Peter Dadam. Correctness criteria for dynamic changes in workflow systems: A survey [ J ]. *Data & Knowledge Engineering* 2004, 50( 1 ): 9- 34
- [ 10 ] Markus Kradošer Andreas Geppert. Dynamic workflow schema evolution based on workflow type versioning and workflow migration [ A ]. *Proceedings of the Fourth ECIS International Conference on Cooperative Information Systems* [ C ]. Edinburgh Scotland 1999 104 - 114
- [ 11 ] Paulo Dias Pedro Vieira Rita-Silva Rita-Silva. Dynamic evolution in workflow management systems [ A ]. *Proceedings of 14th International Workshop on Database and Expert Systems Applications (DEXA '03)* [ C ]. Prague Czech Republic; 2003. 254- 260

#### 作者简介:



闫 哲 男, 1980 年生于河北衡水, 博士生, 主要研究领域为软件工程、工作流及相关技术. Email: yzh@ cs.pku.edu.cn.



赵 文 男, 1967 年生于辽宁大连, 博士, 主要研究领域为软件工程、工作流及相关技术.