

# 面向方面业务流程建模中的方面追踪

倪珊珊<sup>1</sup> 张璇<sup>1,2</sup> 李彤<sup>1,2</sup> 张瑞云<sup>1</sup>

(云南大学软件学院 昆明 650091)<sup>1</sup> (云南省软件工程重点实验室 昆明 650091)<sup>2</sup>

**摘要** 降低模型的复杂度在业务流程管理(BPM)领域是一个关键的问题。面向方面的业务流程建模主张从主流程中分离出不同的关注点,并单独建模,然后通过一定的编织机制组合方面与主流程。但如何验证方面织入对主流程的影响,是一个亟待解决的问题。基于 Petri 网提出并实现了面向方面业务流程建模中的方面追踪方法,并用一个银行业务流程的案例来验证了所提出的方法。

**关键词** 面向方面建模,方面追踪,业务流程建模,Petri 网,流程改进

**中图分类号** TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.8.044

## Aspect Tracing in Aspect Oriented Business Process Modeling

NI Shan-shan<sup>1</sup> ZHANG Xuan<sup>1,2</sup> LI Tong<sup>1,2</sup> ZHANG Rui-yun<sup>1</sup>

(School of Software, Yunnan University, Kunming 650091, China)<sup>1</sup>

(Key Laboratory of Software Engineering of Yunnan Province, Kunming 650091, China)<sup>2</sup>

**Abstract** Reducing the complexity of models is an important issue in the business process management(BPM) area. Aspect oriented business process modeling advocates separating different concerns from the main process and modeling them separately, and then combining aspects and main process with a certain weaving mechanism. But there is a problem on how to verify the impacts of weaved aspect on the main process. We presented and realized an aspect tracing method in aspect oriented business process modelling based on Petri-nets, discussed a case study based on the banking business process to demonstrate the proposed approach.

**Keywords** Aspect oriented modeling, Aspect tracing, Business process modeling, Petri-nets, Process improvement

## 1 引言

业务流程模型的复杂性导致它们难以被理解、维护和评估,因此降低模型的复杂度在业务流程管理(BPM)领域是一个关键的问题<sup>[1]</sup>。流程模型包括不同的活动,它们分别解决业务流程中的不同关注点。通常其中的一些关注点是分散和纠缠在多个业务流程中的,这会导致模型更加复杂,难以理解。面向方面是解决信息系统中横切关注点复杂性的一种重要方法,这种模式在编程领域已经被很好地研究和实现,如 AspectJ 开发语言。然而,业务流程管理中的面向方面才刚刚开始<sup>[2]</sup>。面向方面的业务流程建模主张从主流程中分离出不同的关注点,并单独建模,然后通过一定的编织机制组合方面与主流程。为了验证方面是否按照我们的期望工作,需要对其进行分析测试。在编程领域,可以通过单元测试用例对编织方面前后的程序分别进行功能测试,然后比较输出来决定方面是否产生异常影响<sup>[3]</sup>。但在面向方面的业务流程建模领域,如何验证方面织入对主流程的影响,是一个亟待解决的问题。

方面追踪的目标正是验证方面织入是否对主流程产生影响,以及验证产生的影响是正常影响还是异常影响。方面织

入主流程后,能够正常行使方面的功能,并且不破坏原始业务流程的功能,则方面的影响视为正常影响;方面织入主流程后,破坏了原始业务流程的功能或导致新的业务流程与原始业务流程不一致,则方面的影响视为异常影响。文中所指的原始业务流程 Petri 网是采用 Petri 网对业务流程建模所得到的模型,而织入方面后的业务流程 Petri 网是采用面向方面的思想,从原始业务流程中分离出横切关注点,用 Petri 网建模,并以方面的形式织入主流程的 Petri 网模型中所得到的模型。本文基于 Petri 网给出了方面影响正常的判定准则,提出并实现了一种面向方面业务流程建模中的方面追踪方法。在同一初始格局和给定目标格局下,分析方面织入前后的两个业务流程 Petri 网的变迁发生序列,以此来分析方面织入前后流程的一致性,即分析方面的织入对主流程产生的影响是否是正常影响,并用一个银行业务流程的案例来验证了本文所提出的方法。

本文第 2 节介绍与本文相关的工作;第 3 节给出了方面影响正常的判定准则,提出了一种基于 Petri 网的方面追踪方法;第 4 节通过一个银行业务流程的实际案例来阐明本方法可以方便有效地进行面向方面业务流程建模中的方面追踪;最后总结全文。

到稿日期:2014-08-05 返修日期:2014-10-09 本文受国家自然科学基金项目(61262025,61379032,61262024),云南省应用基础研究计划面上项目(2012FB118),云南省教育厅科学研究基金项目(2012Y257),云南省软件工程重点实验室开放基金项目(2011SE09),云南大学“中青年骨干教师培养计划”专项经费资助。

倪珊珊 硕士生,主要研究方向为面向方面建模、方面追踪;张璇 博士,副教授,主要研究方向为软件工程、需求工程、面向方面建模, E-mail: zhxuan@ynu.edu.cn;李彤 博士,教授,主要研究方向为软件过程、软件演化;张瑞云 硕士生,主要研究方向为面向方面建模、流程改进。

## 2 相关工作

本文的主要工作是提出业务流程模型中的方面追踪方法,因此,本节将对方面追踪相关领域的工作进行简要的介绍。

Bedir Tekinerdoğan 等<sup>[4]</sup>提出了一个关注点追踪元模型(Concern Traceability Meta-model, CTM),使整个生命周期中的关注点可追踪,并独立于所使用的方面语言;还可根据所要求的方面语言,定义专门的追踪模型,但并未提及具体的方面追踪方案。Yijun Yu 等<sup>[3,5]</sup>借用面向方面编程(AOP)的概念,探索解决横切关注点的分散性问题,并提出了一个关注点的追踪框架,但仍未提出详细的方面追踪方案。Marta S. Tabares 等<sup>[6]</sup>提出一个可追踪性元方面(Meta Aspect for Traceability, MAT),它提供追踪语义来识别源和目标元素之间的踪迹链以及它们的转换规则,通过验证转换规则的完整性和一致性来支持横切关注点的追踪,但其追踪语义的复杂性限制了其应用。Rémi Douence 等<sup>[7]</sup>呈现了一个基于踪迹的方面模型,它能够通过方程式推理来证明方面的性能,使用方面语言的限制来规范化表达式;并阐述了如何让方面的影响可控,以及如何实现方面的静态编织,但由于该方面模型是基于程序执行的踪迹来建立的,因此限制了其应用范围。陈生庆等<sup>[8]</sup>使用面向方面扩展了MCI操作语义,并定义了软件重构前后两个程序的观测等价性,但该方法只适用于软件工程领域的方面追踪。Maximilian Störzer 等<sup>[9]</sup>提出了一种方面动态分析方法,使用程序追踪来观测系统行为的改变,以此来分析方面通知应用到给定系统中所产生的影响,但其只适用于面向方面编程领域的方面追踪。

## 3 基于 Petri 网的业务流程建模中的方面追踪

本文通过对比原始业务流程 Petri 网与织入方面后的业务流程 Petri 网执行的变迁序列来研究方面织入的影响。由于 Petri 网执行会产生多条变迁序列,为了保证变迁序列具有可比性,因此做出如下限制:1)相同的初始格局;2)相同的目标格局;3)对于迭代织入方式,在做序列对比时,要首先去除重复执行的活动;4)在网中有选择结构时,要保证两个网选择同一条分支,即要保证方面织入前后的两个业务流程的起始活动和终止活动相同,且执行的路径是确定的。

在进行方面编织的过程中,可能会出现多个方面在同一个切入点织入主流程的情况,为了避免方面间的冲突,就需要通过分析变迁之间的依赖关系来确定各个方面的织入顺序。变迁之间的依赖关系有以下3种:直接数据依赖、间接数据依赖和控制依赖。

**定义 1(直接数据依赖)** 给定一个业务流程 Petri 网的变迁发生序列  $\sigma$  中的任意两个变迁  $t_i$  和  $t_j$ , 其中  $i < j$ , 称变迁  $t_j$  直接数据依赖于变迁  $t_i$ , 当且仅当  $t_j \cap t_i \neq \emptyset$ , 记作  $t_i \xrightarrow{dd} t_j$ 。

**定义 2(间接数据依赖)** 给定一个业务流程 Petri 网的变迁发生序列  $\sigma$  中的任意 3 个变迁  $t_i, t_k$  和  $t_j$ , 其中  $i < k < j$ , 若变迁  $t_j$  直接或间接数据依赖于变迁  $t_k$ , 变迁  $t_k$  直接或间接数据依赖于变迁  $t_i$ , 则变迁  $t_j$  间接数据依赖于变迁  $t_i$ , 记作  $t_i \xrightarrow{id} t_j$ 。

**定义 3(控制依赖)** 给定一个业务流程 Petri 网的变迁发生序列  $\sigma$  中的任意两个变迁  $t_i$  和  $t_j$ , 若  $t_j$  是否发生取决于  $t_i$ , 则称  $t_j$  控制依赖于  $t_i$ <sup>[10]</sup>, 记作  $t_i \xrightarrow{c} t_j$ 。

**定义 4** 直接数据依赖和间接数据依赖统称为数据依赖, 记作  $t_i \xrightarrow{d} t_j$ , 数据依赖和控制依赖统称为依赖。

**定义 5(不相关)** 给定一个业务流程 Petri 网的变迁发生序列  $\sigma$  中的任意两个变迁  $t_i$  和  $t_j$ , 若变迁  $t_j$  不依赖于变迁  $t_i$ , 同时变迁  $t_i$  也不依赖于变迁  $t_j$ , 则称变迁  $t_j$  与变迁  $t_i$  不相关<sup>[11]</sup>。

有了变迁之间的依赖关系后,可以定义变迁之间的发生顺序:顺序关系、选择关系、并发关系和迭代关系。

**定义 6(顺序关系)** 给定一个业务流程 Petri 网的变迁发生序列  $\sigma$  中的任意两个变迁  $t_i$  和  $t_j$ , 其中  $i < j$ , 若  $t_i \xrightarrow{d} t_j$ , 则称  $t_i$  与  $t_j$  之间为顺序关系。

**定义 7(选择关系)** 给定一个业务流程 Petri 网的变迁发生序列  $\sigma$  中的变迁  $t_1, t_2, \dots, t_n \in T$ , 若  $t_1 \xrightarrow{c} t_2, t_1 \xrightarrow{c} t_3, \dots, t_1 \xrightarrow{c} t_n$ , 则存在且只存在一个变迁  $t_i \in \{t_2, t_3, \dots, t_n\}$  必须发生, 称  $t_2, t_3, \dots, t_n$  之间为选择关系。

**定义 8(并发关系)** 给定一个业务流程 Petri 网的变迁发生序列  $\sigma$  中的任意两个变迁  $t_i$  和  $t_j$ , 若  $t_j$  与  $t_i$  不相关, 则称  $t_i$  与  $t_j$  之间为并发关系。

**定义 9(迭代关系)** 给定一个业务流程 Petri 网的变迁发生序列  $\sigma$  中的变迁  $t_i, t_j$  和  $t_k$ , 若  $t_i \xrightarrow{d} t_j, t_j \xrightarrow{d} t_k, t_k \xrightarrow{d} t_i$ , 则称  $t_i, t_j$  与  $t_k$  之间为迭代关系。

### 3.1 方面影响正常的判定准则

本节讨论方面织入对原始业务流程产生正常影响的判定准则,判定准则不能太过严格,只要方面织入后,能正确行使其功能,且不破坏原始业务流程的功能,即认为是正常影响,从而可以让更多的方面得以织入。

**判定准则 1:** 给定原始业务流程 Petri 网和织入方面后的业务流程 Petri 网分别为  $N_s$  和  $N_t$ , 对于给定的初始格局  $M_0$  和目标格局  $M$ , 在  $N_s$  中有变迁发生序列  $\sigma_1$ , 使得  $M_0[\sigma_1 > M$ , 在  $N_t$  中有变迁发生序列  $\sigma_2$ , 使得  $M_0[\sigma_2 > M$ 。若  $\sigma_1$  中的变迁发生顺序在  $\sigma_2$  中不改变, 则认为方面织入产生的影响是正常影响。

准则 1 太过严格,只能处理原始业务流程 Petri 网与织入方面后的业务流程 Petri 网中变迁顺序完全不变的情况,由于织入方面后可能会对原始业务流程 Petri 网作出改进,这就使得网中的变迁不能完全按照原来的顺序执行,比如并发会导致顺序变化,但这种变化应该属于正常影响。因此,本文将通过举例的方式,逐步放松判定准则 1 的限制,使更多产生正常影响的方面得以织入。

图 1(a)和图 1(b)分别代表原始业务流程 Petri 网和织入方面后的业务流程 Petri 网。在原始业务流程 Petri 网中,变迁的发生序列为  $t_1 t_2 t_3 t_4$ , 织入方面后的网中变迁的发生序列可以为  $t_1 t_2 t_3 t_4$ , 也可以为  $t_1 t_3 t_2 t_4$ 。由于前后两个序列中变迁的发生顺序改变了,根据判定准则 1,该方面的织入的影响是异常影响。然而,如果变迁  $t_2$  和  $t_3$  本身就是不相关的,那么它们执行顺序改变并不会产生影响。因此在这种条件下,方

面织人的影响是正常影响。

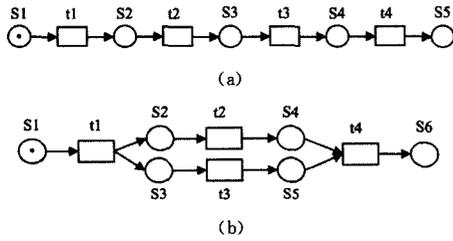


图1 判断准则2举例

在一个业务流程 Petri 网的变迁发生序列  $\sigma$  中,可以找到一些不相关的变迁,由于不相关的变迁之间的执行顺序可以是任意的,如果变换它们在  $\sigma$  中的顺序,可以得到一系列变迁序列,这些变迁序列的集合记为  $D(\sigma)$ <sup>[11]</sup>。在引入  $D(\sigma)$  的基础上,给出判定准则 2。

判定准则 2: 给定原始业务流程 Petri 网和织入方面后的业务流程 Petri 网分别为  $N_s$  和  $N_t$ , 对于给定的初始格局  $M_0$  和目标格局  $M$ , 在  $N_s$  中有变迁发生序列  $\sigma_1$ , 使得  $M_0[\sigma_1 > M$ , 若在  $N_t$  中有变迁发生序列  $\sigma_2$ , 且  $\exists \lambda \in D(\sigma_2)$ , 满足  $M_0[\lambda > M$ , 则认为方面织入影响是正常影响。

图 2(a)和图 2(b)分别代表原始业务流程 Petri 网和织入方面后的业务流程 Petri 网。在原始业务流程 Petri 网中,变迁的发生序列为  $\sigma_1 = t_1 t_2 t_3$ , 织入方面后的网中变迁的发生序列为  $\sigma_2 = (t_1 t_2 t_4)^n t_3$ 。  $\sigma_2$  的一个特例为  $t_1 t_2 t_4 t_1 t_2 t_4 t_3$ , 由于变迁序列中出现了重复,  $t_2$  发生之后又执行了  $t_1$ , 出现了顺序上的变化。根据判定准则 1 和判定准则 2, 该方面的织入的影响是异常影响。然而, 在迭代织入的情况下, 重复执行的变迁序列应视为整体来分析, 不应该将二次迭代的变迁序列与第一次执行的序列对比, 这种情况不应视为顺序改变。因此在这种条件下, 方面织入的影响是正常影响。由此, 给出判定准则 3。

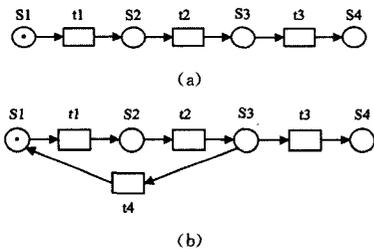


图2 判定准则3举例

判定准则 3: 给定原始业务流程 Petri 网和织入方面后的业务流程 Petri 网分别为  $N_s$  和  $N_t$ , 对于给定的初始格局  $M_0$  和目标格局  $M$ , 在  $N_s$  中有变迁发生序列  $\sigma_1$ , 使得  $M_0[\sigma_1 > M$ 。若以迭代织入方式织入方面后, 在  $N_t$  中有变迁发生序列  $\sigma_2 = t_1 \dots t_i \lambda^n \dots t_j t_k$ , 其中  $1 < i < k < j < n$ , 使得  $M_0[\sigma_2 > M$ , 无论  $n$  的值为多少, 方面织入的影响都视为正常影响。

图 3(a)和图 3(b)分别代表原始业务流程 Petri 网和织入方面后的业务流程 Petri 网。在基本网中, 变迁的发生序列为  $\sigma_1 = t_1 t_2$ , 织入方面后的网中变迁的发生序列可以为  $t_1 t_2$ , 也可以为  $t_1 t_3$ 。在这种情况下, 可能会出现方面织入阻止了变迁  $t_2$  的发生并用方面中的  $t_3$  代替  $t_2$  的情况。在上述标准的条

件下, 该方面的织入是不允许的。但不能说方面织入的影响一定是异常影响。如果  $t_3$  中包含了  $t_2$  的全部功能, 那么这种改变就是正常影响。由此, 给出判定准则 4。

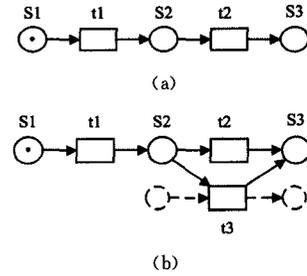


图3 判定准则4举例

判定准则 4: 给定原始业务流程 Petri 网和织入方面后的业务流程 Petri 网分别为  $N_s$  和  $N_t$ , 对于给定的初始格局  $M_0$  和目标格局  $M$ , 在  $N_s$  中有变迁发生序列  $\sigma_1$ , 使得  $M_0[\sigma_1 > M$ , 在  $N_t$  中有变迁发生序列  $\sigma_2$ , 使得  $M_0[\sigma_2 > M$ ,  $\exists t_i \in T$  且  $\#(t_i/\sigma_1) \neq 0$ ,  $\#(t_i/\sigma_2) = 0$ ,  $\sigma_2$  中用  $t_j$  (且  $\#(t_j/\sigma_1) = 0$ ) 代替了  $t_i$ , 其中  $\#(t_i/\sigma_1)$  表示变迁  $t_i$  在序列  $\sigma_1$  中出现的次数。如果  $t_j$  中包含了  $t_i$  的全部功能, 则认为方面织入的影响是正常影响。

### 3.2 方面追踪方法算法

基于以上判定准则, 本文提出并在 PIPE 下实现了基于 Petri 网的面向方面业务流程建模中的方面追踪方法的算法。该算法的大致思路是: 首先去除方面织入前后的两个序列的重复部分(这主要是消除迭代织入方式产生的重复), 以及在序列  $T$  中出现而在序列  $S$  中未出现的变迁(因为目的是判定方面织入对原始流程产生的影响)。接着用循环来判断  $S$  和  $T$  中的变迁是否相同, 如果不同, 则将其放入差异集  $P$  中。全部判定完成后, 如果差异集  $P$  为空, 则两条序列是一致的, 认为方面织入的影响是正常影响; 否则, 要判定  $P$  中元素的结构关系  $R_s$  和  $R_T$  (因为执行序列是结构的直接反映)。如果  $P$  中元素的结构在方面织入前后发生了变化, 且变为并发或者选择, 则认为方面织入的影响是正常影响, 否则认为方面织入产生的影响是异常影响。

#### 算法 SeqCompare

输入: 织入方面前的 Petri 网  $N_s$  及其一条变迁发生序列  $S$ , 织入方面后的 Petri 网  $N_T$  及其一条变迁发生序列  $T$

输出: true 或 false

begin

去除重复变迁 // 满足判定准则 3

去除在序列  $T$  中出现而序列  $S$  中未出现的变迁;

$k=0; P=\emptyset; i=0; j=0$

while  $i < |S|$  do

while  $j < |T|$  do

begin

if  $S[i] == T[j]$ ;

$i++$

$j++$

else

$P[k] = T[j-1]$

$k++$

```

P[k]=T[j] //将不一致的变迁放入差异集 P 中
j++
end
if P==∅
return true //满足判定准则 1
else
获取 P 中的元素在 NS 和 NT 中的结构关系 RS 和 RT;
for i=0 to |P| do
for j=0 to |P| do
begin
if i≠j
if RS(P[i],P[j])≠RT(P[i],P[j])
if RT(P[i],P[j])=="并发关系" or RT(P[i],P[j])=="选择关系"
return true //满足判定准则 2 和 4
else
return false
end
end
end

```

上述算法中  $N_S$  是织入方面前的 Petri 网,  $S$  为其中一条变迁序列,  $N_T$  是织入方面后的 Petri 网,  $T$  为其中一条变迁序列。该算法的输出有两种情况, true 表示方面织入的影响是正常影响, false 说明方面织入的影响是异常影响。  $P$  存储的是差异集, 即两个变迁序列不一致的部分。

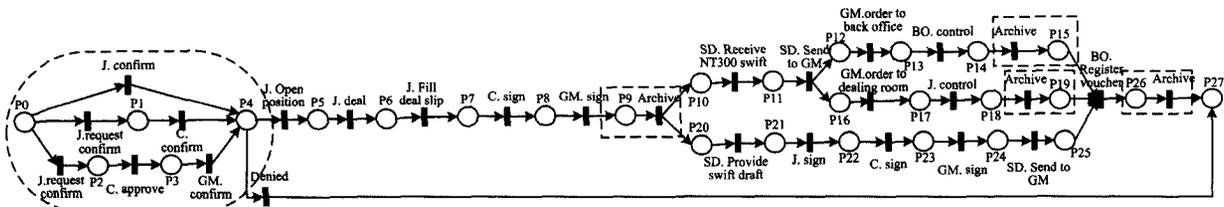


图 4 传统投机交易流程的 Petri 网模型

图 5 为投机交易流程的面向方面 Petri 网模型, 该模型从主流程中分离出了两个方面流程, 分别是安全性方面和日志记录方面, 并在主流程中标出了方面的织入位置。该模型中, 主流程中的变迁发生序列为  $\sigma_2 = J. Open position \rightarrow J. deal \rightarrow J. Fill deal slip \rightarrow C. sign \rightarrow GM. sign \rightarrow Archive \rightarrow ((SD. Receive NT300 swift \rightarrow SD. Send to GM \rightarrow ((GM. order to back office \rightarrow BO. control \rightarrow Archive) \parallel (GM. order to dealing room \rightarrow J. control \rightarrow Archive))) \parallel (SD. Provide swift draft \rightarrow J. sign \rightarrow C. sign \rightarrow GM. sign \rightarrow SD. Send to GM)) \rightarrow BO. Regi-$

#### 4 案例分析

本文引用 Amin Jalali 在其研究中采用的银行业务流程的案例, 并选择其中的投机交易流程(Deal for speculation)<sup>[12]</sup> 进行分析, 文中所涉及的所有的分析工作都是用本文所提出并实现的方法在 PIPE 下完成的。

图 4 为传统投机交易流程的 Petri 网模型, 它将横切关注点和核心关注点建模在同一个网中, 虚线椭圆框表示的是与安全性有关的活动, 虚线方框表示与日志记录有关的活动。从这个模型中可以看出, 横切关注点是散落在整个网中的, 尤其是日记记录, 这会导致网结构的复杂性增加且造成结构混乱。经过分析, 得到这个网中的变迁发生序列为  $\sigma_1 = (J. confirm \mid (J. request confirm \rightarrow C. confirm) \mid (J. request confirm \rightarrow C. approve \rightarrow GM. confirm)) \rightarrow (Denied \mid (J. Open position \rightarrow J. deal \rightarrow J. Fill deal slip \rightarrow C. sign \rightarrow GM. sign \rightarrow Archive \rightarrow ((SD. Receive NT300 swift \rightarrow SD. Send to GM \rightarrow ((GM. order to back office \rightarrow BO. control \rightarrow Archive) \parallel (GM. order to dealing room \rightarrow J. control \rightarrow Archive))) \parallel (SD. Provide swift draft \rightarrow J. sign \rightarrow C. sign \rightarrow GM. sign \rightarrow SD. Send to GM)) \rightarrow BO. Register voucher \rightarrow Archive)$ 。其中, “|”表示两个变迁或变迁序列之间是选择关系, “||”表示两个变迁或变迁序列之间是并发关系。

ster voucher  $\rightarrow Archive$ 。织入方面后, 该网的变迁发生序列为  $\sigma_3 = (J. confirm \mid (J. request confirm \rightarrow (C. confirm \mid (C. approve \rightarrow GM. confirm)))) \rightarrow (Denied \mid (J. Open position \rightarrow J. deal \rightarrow J. Fill deal slip \rightarrow C. sign \rightarrow GM. sign \rightarrow Archive \rightarrow ((SD. Receive NT300 swift \rightarrow SD. Send to GM \rightarrow ((GM. order to back office \rightarrow BO. control \rightarrow Archive) \parallel (GM. order to dealing room \rightarrow J. control \rightarrow Archive))) \parallel (SD. Provide swift draft \rightarrow J. sign \rightarrow C. sign \rightarrow GM. sign \rightarrow SD. Send to GM)) \rightarrow BO. Register voucher \rightarrow Archive)$ 。

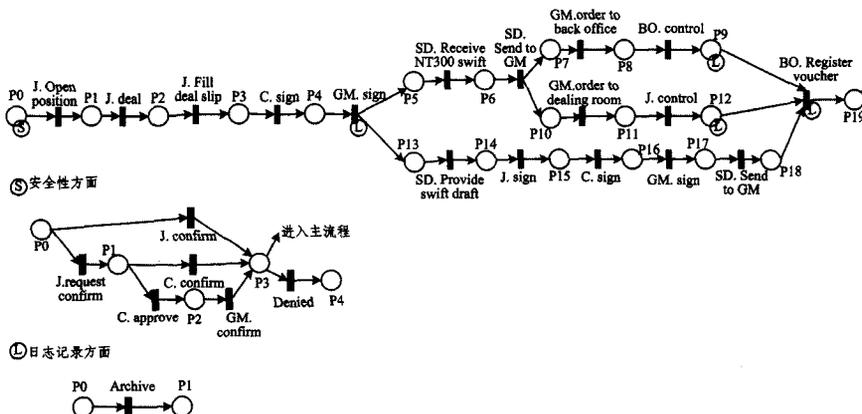


图 5 面向方面的投机交易流程的 Petri 网模型

对比  $\sigma_2$  和  $\sigma_3$ ,  $\sigma_2$  中顺序执行的所有变迁在  $\sigma_3$  中都按原来的顺序执行, 方面织入后只是扩展了原来的变迁序列, 满足判定准则 1; 而互为并发关系的几个变迁可以有多种发生序列, 不管它们按哪个序列发生, 其影响都是正常影响, 满足判定准则 2; 对于选择关系的几个变迁, 哪些变迁能发生是由方面中的逻辑关系来决定的, 对基本网不会造成异常影响。

对比  $\sigma_1$  和  $\sigma_3$ , 作者在分离出方面后, 对方面也作了相应的改进, 将重复的变迁 (即变迁  $J. request confirm$ ) 作了再次合并, 使原模型中与安全性相关的变迁从 7 个减少为 6 个, 与日志记录相关的变迁也都合并为 1 个。通过这样的处理, 大大减少了模型的复杂度。所以说  $\sigma_1$  和  $\sigma_3$  在本质上是一样的。即分离出横切关注点之后的模型与原模型是一样的, 且降低了模型的复杂度, 使之更易理解。

Amin Jalali 的关注点分离方式是一种可行的面向方面建模方式, 对 Amin Jalali 分离出的安全性方面再作进一步的划分, 来进一步验证本文的方面追踪方法。

将图 5 中 Amin Jalali 分离出的安全性方面细化为 5 个子方面, 在进行安全性方面编织前, 要先将 5 个子方面编织成一个 Petri 网, 然后再将得到的 Petri 网作为一个方面织入主流程。经过方面间的结构分析, 可以得到依赖关系,  $S1$  和  $S2$  控制依赖于  $P0$ ,  $S3$  和  $S4$  控制依赖于  $S2$ , 而  $S5$  数据依赖于  $S4$ 。由此便得到了各子方面的织入顺序, 如图 6 所示。按此织入顺序编织各子方面, 可得到 Amin Jalali 分离出的方面。

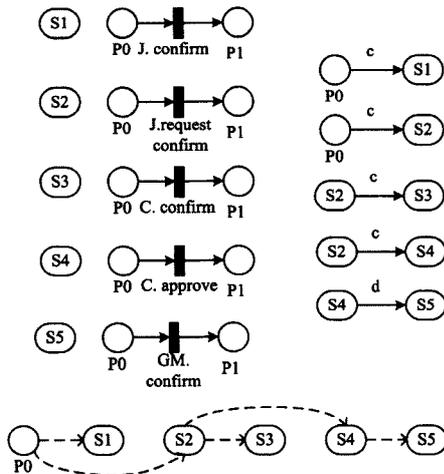


图 6 安全性子方面及其依赖关系

若忽略了方面之间的关系, 不做方面间依赖分析, 就会出现合成的方面与原流程不一致的情况。在本例中, 安全性方面  $S$  所构成的 Petri 网中包含选择和顺序两种结构, 在不作依赖分析的情况下, 可能会导致合成的方面出现结构上的变化, 比如选择结构变为顺序结构, 如图 7 所示。这会导致方面织入前后的两个业务流程 Petri 网执行所得到的变迁序列不一致, 这并不满足本文的方面织入影响正常的判定准则, 所以不作依赖分析就进行方面编织可能会对原始流程产生异常影响。

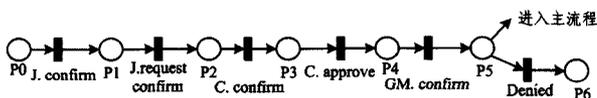


图 7 选择结构变为顺序结构后的 Petri 网模型

**结束语** 本文提出并实现了一种基于 Petri 网的面向方面建模中的方面追踪方法, 通过对方面织入前后的两个业务流程 Petri 网的变迁发生序列进行对比, 来识别方面流程对主流程的影响。该方法具有一定的灵活性, 不仅能用于业务流程领域的面向方面建模的方面追踪, 也可用于业务流程改进中的方面追踪, 通过分析改进前后的两个业务流程的 Petri 网模型的变迁序列差异来分析流程改进的影响, 还可用于软件领域的面向方面建模的方面追踪。今后将通过进程代数来对方面织入前后的两个业务流程 Petri 网进行一致性验证来开展对方面追踪的进一步研究。

## 参考文献

- [1] Jalali A, Wohed P, Ouyang C, et al. Dynamic Weaving in Aspect Oriented Business Process Management [C] // On the Move to Meaningful Internet Systems; OTM 2013 Conferences. 2013; 2-20
- [2] Jalali A, Wohed P, Ouyang C. Operational semantics of aspects in business process management [C] // On the Move to Meaningful Internet Systems; OTM 2012 Workshops. 2012; 649-653
- [3] Yu Y, Niu N, González-Baixauli B, et al. Requirements engineering and aspects [M] // Design Requirements Engineering: A Ten-Year Perspective. Springer, 2009; 432-452
- [4] Tekinerdoğan B, Hofmann C, Aksit M, et al. Metamodel for tracing concerns across the life cycle [M] // Early Aspects; Current Challenges and Future Directions. Springer, 2007; 175-194
- [5] Yu Y, Niu N, González-Baixauli B, et al. Tracing and validating goal aspects [C] // 15th IEEE International Requirements Engineering Conference, 2007. 2007; 53-56
- [6] Tabares M S, Moreira A, Anaya R, et al. A traceability method for crosscutting concerns with transformation rules [C] // Proceedings of the Early Aspects at ICSE; Workshops in Aspect-Oriented Requirements Engineering and Architecture Design. 2007, 7
- [7] Douence R, Fradet P, Südholt M. Trace-based aspects [J]. Aspect-Oriented Software Development, 2004, 35, 141-150
- [8] 陈生庆, 张立臣, 陈广明. 面向方面软件重构等价性形式化证明方法 [J]. 计算机科学, 2006, 33(7): 257-261  
Chen Sheng-qing, Zhang Li-chen, Chen Guang-ming. An Equivalence Proving in Formal Method for Aspect Oriented Refactory [J]. Computer Science, 2006, 33(7): 257-261
- [9] Störzer M, Krinke J, Breu S. Trace analysis for aspect application [C] // Workshop on Analysis of Aspect-Oriented Software (AAOS). 2003
- [10] Li T. An Approach to Modelling Software Evolution Processes [M]. Springer, 2008
- [11] 宋巍, 马晓星, 吕建. Web 服务组合动态演化的实例可迁移性 [J]. 计算机学报, 2009, 32(9): 1816-1831  
Song Wei, Ma Xiao-xing, Lu Jian. Instance Migration in Dynamic Evolution of Web Service Compositions [J]. Chinese Journal of Computers, 2009, 32(9): 1816-1831
- [12] Jalali A, Wohed P. Foundation of Aspect Oriented Business Process Management [D]. Stockholm: Stockholm University, 2011