

# 基于补偿事务的业务流程异常处理建模研究

雷义伟 贲可荣

(海军工程大学计算机工程系 武汉 430033)

**摘要** 对补偿触发条件的判断会影响补偿的正确性,由于业务流程的活动之间具有补偿依赖关系,尤其是并发结构中的补偿依赖,使得流程开发人员在设计补偿流程时容易出错。分析了顺序、选择、并发等几种基本流程结构及其复合结构中的补偿依赖,给出了补偿活动触发条件计算方法和补偿事务的 Petri 网模型构造过程,并通过实例说明了方法的可行性。

**关键词** 业务流程,异常处理,补偿,Petri 网

**中图分类号** TP302.8 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.7.019

## Research on Compensating Transaction Based Business Processes Exception Handling Model

LEI Yi-wei BEN Ke-rong

(Department of Computer Engineering, Naval University of Engineering, Wuhan 430033, China)

**Abstract** The decision of compensation trigger conditions affects the correctness of compensation. Due to the compensation relations among business process activities, especially the compensation relations in concurrent structures, process developers are prone to make mistakes when designing the compensation processes. This paper analyzed the compensation dependences in the structure of several basic processes such as sequence, selection, concurrency, and their composite structures. Computation methods of compensation trigger conditions, as well as the construction process of Petri-net models were given, and the feasibility of the method was also illustrated by an example.

**Keywords** Business process, Exception handling, Compensation, Petri nets

## 1 引言

事务机制是保证业务流程可用性的一种重要方式,事务的原子性、一致性、隔离性和持久性(Atomicity, Consistency, Isolation, Durability, ACID)使执行结果具有“全有或全无”的特点,从而保证程序执行状态的一致性。补偿指在流程出现异常时,撤销全部已完成活动产生的结果,使流程恢复到失效之前的某个状态,从而保证程序状态一致性。对补偿触发条件的判定准确与否决定了活动能否被正确补偿,补偿触发条件的误判会导致不必要的补偿或者需补偿的活动未被补偿,这些不正确的补偿行为将会影响业务流程的性能,甚至导致业务流程出错。业务流程的活动彼此存在着补偿依赖(即确定是否补偿某一活动,依赖于其他活动执行的状态),尤其是并发结构中的补偿依赖,随并发分支个数成指数级增长,使补偿条件难以被准确捕获,导致流程开发人员在设计补偿时极易出错。本文在已有工作的基础上,研究几种基本交互关系中各活动的补偿触发条件,给出了活动补偿触发条件计算方法。在此基础上,给出基本交互关系的 Petri 网模型补偿设计过程。

## 2 研究背景

补偿机制是业务流程中常用的一种异常处理手段,当业

务流程的活动出现异常时,它通过撤销已完成活动结果,使发生失效的流程恢复到失效之前的某个状态,从而使流程能继续运行下去。因其在保证业务流程可用性方面的重要作用,基于补偿的异常处理成为近年来研究的热点。

Colombo 等人对补偿事务进行了综述,比较了已有补偿事务的形式化模型<sup>[1]</sup>。Haddad 等人针对传统的服务选择算法关注 QoS 而忽略服务事务属性的问题,提出一种综合考虑功能属性、QoS 和事务属性的服务选择算法<sup>[2]</sup>。文献<sup>[3]</sup>先将服务按 4 种事务属性分类,再分别对 4 类不同属性服务集合采取基于 QoS 的服务选择方法构造所需的服务组合。文献<sup>[4]</sup>采用线性规划方法,在满足事务属性的前提下,使服务组合 QoS 最大化。这类研究主要关注如何选取服务以满足业务流程的事务属性,但只能保证得到的业务流程具备可补偿特性,因此,还需以某种方式组织其补偿活动有序进行,才能保证业务流程在出错时被正确补偿。Gaaloul 等人提出通过分析工作流日志,发现其事务行为,进而改进其恢复机制<sup>[5]</sup>。文献<sup>[6]</sup>从工作流日志中发现活动前驱和后继结点,得到活动间的调用依赖关系,生成工作流的补偿路径。但日志只能反映业务流程的一部分事务行为,具有不完全性。

为了解决上述问题,本文提出了一种用于业务流程开发阶段、基于补偿事务的异常处理建模方法,通过分析顺序、选

到稿日期:2014-06-19 返修日期:2014-10-09 本文受国家自然科学基金项目(61272108)资助。

雷义伟(1983-),男,博士生,主要研究方向为软件测试、软件维护,E-mail:syjdlyw@163.com;贲可荣(1963-),男,博士,教授,主要研究方向为软件工程、人工智能。

择、并发等几种基本流程结构及其复合结构中的补偿依赖,给出补偿活动触发条件计算方法,提出补偿事务的 Petri 网模型设计过程,最后通过实例描述其建模过程。

### 3 基于补偿事务的异常处理

在活动或流程出现异常时,事务属性决定对其采取的异常处理方式,下面先介绍活动和流程的事务属性。

活动是构成业务流程的基本元素,活动的事务属性集合为  $\{p, c, r, pr, cr\}$ <sup>[7]</sup>, 其中:

(1)  $p$ (pivot): 具有  $p$  属性的活动一旦正确完成,它产生的结果不能被补偿;若活动失效,则它对系统不产生任何影响;

(2)  $c$ (compensatable): 具有  $c$  属性的活动正确完成后,可运行另一个活动撤销其产生的结果,称该属性为可补偿属性;

(3)  $r$ (retriable): 具有  $r$  属性的活动在经过有限次调用后,最终能正确完成,称该属性为可重试属性;

(4)  $pr$ (pivot retriable): 具有  $pr$  属性的活动会同时拥有  $p$  属性活动的特点和  $r$  属性活动的特点;

(5)  $cr$ (compensatable retriable): 具有  $cr$  属性的活动会同时拥有  $c$  属性活动的特点和  $r$  属性活动的特点。

由活动事务属性定义及活动与业务流程间关系,可得到业务流程的事务属性集合  $\{A, C, R, AR, CR\}$ <sup>[3]</sup>, 其中:

(1)  $A$ (atomic): 表示业务流程(business process, BP)中所有活动一旦执行成功,则不能被补偿;若 BP 中任一活动发生失效,则所有已正确完成的活动必须被补偿;

(2)  $C$ (compensatable): 表示 BP 中所有活动均具有可补偿事务属性;

(3)  $R$ (retriable): 表示 BP 中所有活动均具有可重试事务属性;

(4)  $AR$ (atomic retriable): 表示 BP 同时具有  $A$  属性和  $R$  属性;

(5)  $CR$ (compensatable retriable): 表示 BP 同时具有  $C$  属性和  $R$  属性。

当活动具有可重试属性(即  $tp_a \in \{r, pr, cr\}$ )时,表明活动在出现异常时,经有限次调用能最终完成,使流程能够继续下去;活动具有可补偿属性(即  $tp_a \in \{c, cr\}$ )时,表明活动在完成后,通过执行另一活动,能取消该活动产生的结果,使流程恢复到之前的某个状态。综上可得到基于事务的两种异常处理方式:基于重试的异常处理和基于补偿的异常处理。

活动按交互关系构成业务流程,业务流程继承其子活动的事务属性。若某一流程  $bp$  的事务属性  $tp_{bp} \in \{C, CR\}$ , 则流程的任一活动均可被补偿,设  $bp$  的活动  $a_j$  发生失效,则可按  $a_{j-1}, a_{j-2}, \dots, a_1$  的顺序进行补偿。由此可得出一个结论:若某一流程事务属性  $tp_{bp} \in \{C, CR\}$ , 则该流程可采用基于补偿的异常处理方式。该结论可作为判定业务流程能否采用基于补偿事务的异常处理方式的一个准则。

### 4 补偿活动的触发条件

业务流程活动整个生命周期是不同状态间变换的过程,活动状态变迁可用一个状态转换图表示,如图 1 所示。椭圆表示活动生命周期中 8 种状态  $\{initial, active, completed, compensated, failed, cancelled, aborted, final\}$ , 有向弧表示变迁发

生顺序,弧上标记表示具体变迁  $\{active(), complete(), faulting(), retry(), compensate(), cancel(), abort(), close()\}$ , 实线椭圆表示活动的一般状态,虚线椭圆表示事务性活动的特有状态。活动状态之间彼此相关,状态变迁同活动间交互有关。

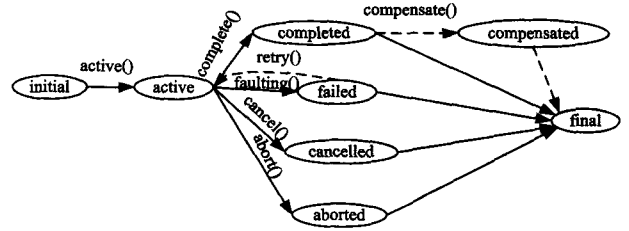


图 1 活动的状态转换图

设  $a_i$  和  $a_j$  为业务流程的两个活动,且  $a_i$  是可补偿的,补偿  $a_i$  的活动设为  $a_i'$ ,  $a_i'$  中用于补偿  $a_i$  的状态变迁为  $a_i'.cp()$ , 若  $a_j$  有状态变迁  $m_j()$ , 且该变迁发生后  $a_j$  所处状态能够触发  $a_i'.cp()$ , 则  $a_j$  当前所处状态为补偿变迁  $a_i'.cp()$  的补偿触发(前提)条件,也是变迁  $a_j.m_j()$  的后继条件。

文献[8]给出了补偿依赖定义:  $depCps(s_1, s_2) = dep(s_1, fail()) \vee dep(s_1, compensate()) \vee dep(s_1, compensate()), s_2, compensate()$ 。但触发条件依赖于活动间交互类型,即使同一活动,交互类型不同,发生状态变迁的触发条件也不同。本文基于文献[8]给出的补偿依赖定义,研究业务流程中不同的流程结构对补偿依赖的影响,由此得出业务流程补偿触发条件的计算公式。

一个活动的后继条件可能成为其他活动的补偿触发条件,以此构成活动间的依赖关系。下面给出 3 种基本交互关系中的补偿触发条件、后继条件的计算公式。

(1) 顺序结构:  $a_1 \oplus a_2 \oplus \dots \oplus a_n$  的补偿触发条件、后继条件的计算公式:

$$pstCon(a_1 \oplus a_2 \oplus \dots \oplus a_n) = pstCon(a_1) = a_1.f_a \vee a_1.cp$$

$$pstCon_{\oplus}(a_i) = a_i.f_a \vee a_i.cp$$

$$preCon(a_i.cp()) = pstCon_{\oplus}(a_{i+1}) = a_{i+1}.f_a \vee a_{i+1}.cp,$$

其中,  $1 \leq i \leq n-1$

$$preCon(a_1 \oplus a_2 \oplus \dots \oplus a_n) = preCon(a_n) = pstCon(succAction)$$

符号  $pstCon$  表示顺序结构  $a_1 \oplus a_2 \oplus \dots \oplus a_n$  作为整体对顺序结构之外活动的后继条件;  $pstCon_{\oplus}$  表示顺序结构内部元素对该结构内部其它元素的后继条件。前缀  $preCon$  表示触发条件。后缀  $cp()$  表示补偿活动  $compensate()$ ,  $f_a$  表示失效状态  $failed$ ,  $cp$  表示已被补偿状态  $compensated$ 。

(2) 选择结构  $a_1 \otimes a_2 \otimes \dots \otimes a_n$  的补偿触发条件、后继条件的计算公式:

$$pstCon(a_1 \otimes a_2 \otimes \dots \otimes a_n) = (\bigwedge_{i=1}^n a_i.f_a) \vee (\bigvee_{i=1}^n (a_i.cp))$$

$$pstCon_{\otimes}(a_i) = null, \text{ 其中, } 1 \leq i \leq n.$$

$$preCon(a_i.cp()) = pstCon(succAction), \text{ 其中, } 1 \leq i \leq n$$

$$preCon(a_1 \otimes a_2 \otimes \dots \otimes a_n) = pstCon(succAction)$$

符号  $pstCon$  表示选择结构  $a_1 \otimes a_2 \otimes \dots \otimes a_n$  作为整体对选择结构之外活动的后继条件;  $pstCon_{\otimes}$  表示选择结构内部元素对该结构内部其它元素的后继条件。

(3) 并发关系  $a_1 \parallel a_2 \parallel \dots \parallel a_n$  的补偿触发条件、后继条件的计算公式:

$$\begin{aligned}
pstCon(a_1 \parallel a_2 \parallel \dots \parallel a_n) &= \bigwedge_{i=1}^n (a_i, fa \vee a_i, cp) \\
pstCon_{||}(a_i) &= a_i, fa \vee a_i, cp \\
preCon(a_1 \parallel a_2 \parallel \dots \parallel a_n) &= pstCon(succAction) \\
preCon(a_i, cp()) &= (\bigvee_{j \neq i} (pstCon_{||}(a_j)) \vee preCon(a_1 \parallel a_2 \parallel \dots \parallel a_n)) \wedge a_i, c, \text{其中}, 1 \leq i \leq n
\end{aligned}$$

符号  $pstCon$  表示并发结构  $a_1 \parallel a_2 \parallel \dots \parallel a_n$  作为整体对该结构之外活动的后继条件;  $pstCon_{||}$  表示并发结构内部元素对该结构内部其它元素的后续条件。后继  $c$  表示已完成状态 completed。由于并发结构中某个分支失效时,其它分支有可能未执行完,因此需要加入“已完成”作为判断条件,避免补偿未完成的活动。

基于上面 3 类补偿触发条件计算公式和 3 类后继条件计算公式,通过对这些公式的组合、迭代,可计算出业务流程中各活动的补偿触发条件。第 5 节将会通过一个实例描述其计算过程。

## 5 模型的定义及建模过程

Petri 网可用来描述活动内部行为,以及活动间交互行为,包括并发模型。同时,有成熟工具用于 Petri 网设计和分析。基于 Petri 网的特点,同时考虑活动具有不同的事务属性,本文采用着色 Petri 网为业务流程基于补偿事务的异常处理模型进行建模。将模型表示为  $CP-CPN = (P, T; F, C)$ , 其中:

(1)  $P$  表示库所集合,  $P = P^b \cup P^p$ ,  $P^b$  表示基本库所集,即用于建模基本业务功能的库所集,  $P^p$  表示补偿库所集。

(2)  $T$  表示变迁集合,  $T = T^b \cup T^p$ ,  $T^b$  表示基本活动变迁集,  $T^p$  表示补偿活动变迁集。对任意  $t \in T^b$  且  $t$  的属性  $tp_i \in \{c, cr\}$ , 存在  $t' \in T^p$ ,  $t'$  能够取消  $t$  产生的结果。对任意  $t \in T^b$ ,  $tp_i \in \{sl\}$  ( $sl$  是活动属性的一种,表示活动是无状态的), 则用一类特殊活动  $\tau$  作为  $t$  的补偿活动,重要的是,补偿活动  $\tau$  不对  $t$  的结果作任何修改,但会使  $t$  处于已补偿的状态。

(3)  $F$  表示流关系映射,  $F: F^b \cup F^p, F^b: (P^b \times T^b) \cup (T^b \times P^b) \rightarrow \{0, 1\}, F^p: (P^p \times T^p) \cup (T^p \times P^p) \rightarrow \{0, 1\}, F^b$  表示基本流关系集合,  $F^p$  表示补偿时流关系集合。并且,  $F(\langle p, t \rangle) = 1 \Leftrightarrow F^p(\langle t', p' \rangle) = 1, F(\langle t, p \rangle) = 1 \Leftrightarrow F^p(\langle p', t' \rangle) = 1$ 。

(4)  $C$  表示颜色映射函数,  $C: C_p \cup C_T, C_p: P \rightarrow \Sigma, C_T: T \rightarrow \Sigma$ 。其中,  $\Sigma = \Sigma_{TP} \cup \{sl\}, \Sigma_{TP} = \{c, cr\}$ 。对于任意  $p \in P$ ,  $p$  到  $\Sigma$  的映射表示库所  $p$  中 token 的事务属性; 对于任意  $t \in T$ ,  $t$  到  $\Sigma$  的映射表示变迁  $t$  所对应活动的事务属性。

根据第 4 节中补偿触发条件、后继条件计算公式,以及 CP-CPN 定义,可以通过建模得到业务流程基于补偿事务的异常处理模型,具体的建模过程如下:

(1) 由业务流程的开发文档,构造出流程基本业务功能模型,这时的模型为一个传统的 CPN 模型,此时生成的业务流程,仅能在不出现任何异常的前提下完成基本的业务功能。

(2) 对 CPN 模型进行扩展。根据 CP-CPN 定义(2)中  $T = T^b \cup T^p$  可知: 对于每一个  $t (t \in T^b)$ , 向模型中添加变迁  $t$  的补偿变迁  $t' (t' \in T^p)$ 。根据 CP-CPN 定义(3)中  $F(\langle p, t \rangle) = 1 \Leftrightarrow F^p(\langle t', p' \rangle) = 1, F(\langle t, p \rangle) = 1 \Leftrightarrow F^p(\langle p', t' \rangle) = 1$  可知: 对每一个  $p (p: F^b(\langle t, p \rangle) = 1)$ , 向模型中添加库所  $p'$ , 并添加

$p'$  到  $t'$  的流关系(弧)使  $F^p(\langle p', t' \rangle) = 1$ ; 对每一个  $p (p: F^b(\langle p, t \rangle) = 1)$ , 向模型中添加库所  $p'$ , 并添加  $t'$  到  $p'$  流关系(弧)使  $F^p(\langle t', p' \rangle) = 1$ 。当不能再向模型中添加任何网元元素时,该步骤完成。

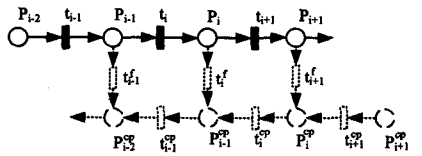
(3) 由第 4 节中的补偿触发条件和后继条件计算公式,计算补偿变迁集合中每一个  $t' (t' \in T^p)$  的补偿触发条件。遵循 Petri 网“与”、“或”构造方法,使用计算得到的补偿触发条件,在步骤(2)得到的模型中为每一个  $t' (t' \in T^p)$  添加补偿触发条件。

经过以上 3 个步骤可以得到业务流程的基于补偿事务的异常处理模型。图 2 是根据上述构造步骤生成的 3 种基本结构的异常处理模型。具体构造过程如下:

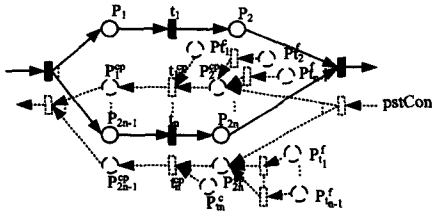
(1) 构造顺序结构补偿模型。图 2(a) 中实线部分表示的是一个顺序结构  $t_1 \oplus t_2 \oplus \dots \oplus t_n$  的基本流程图,  $T^b = \{t_1, t_2, \dots, t_n\}, P^b = \{p_0, p_1, \dots, p_n\}, F = \{\langle p_0, t_1 \rangle, \langle t_1, p_1 \rangle, \dots, \langle p_{n-1}, t_n \rangle, \langle t_n, p_n \rangle\}$ 。对每一个变迁  $t_i (t_i \in T^b)$ , 向模型中添加变迁  $t_i$  对应的补偿变迁, 得到  $T^p = \{t_1^p, t_2^p, \dots, t_n^p\}$ 。对每一个库所  $p_i (p_i \in P^b)$ , 向模型中添加  $p_i$  对应的补偿库所, 得到  $P^p = \{p_0^p, p_1^p, \dots, p_n^p\}$ 。对每一个弧  $\langle p_{i-1}, t_i \rangle$ , 向模型中添加弧  $\langle t_i^p, p_{i-1}^p \rangle$ , 对每一个弧  $\langle t_i, p_{i+1} \rangle$ , 向模型中添加弧  $\langle p_{i+1}^p, t_i^p \rangle$ , 得到  $F^p = \{\langle t_1^p, p_0^p \rangle, \langle p_1^p, t_1^p \rangle, \dots, \langle t_n^p, p_{n-1}^p \rangle, \langle p_n^p, t_n^p \rangle\}$ 。由公式  $preCon(a_i, cp()) = pstCon_{\oplus}(a_{i+1}) = a_{i+1}. fa \vee a_{i+1}. cp$ , 为每一个活动  $t_i$  添加补偿触发条件  $t_{i+1}. fa \vee t_{i+1}. cp$ 。当所有的补偿条件添加完毕时,模型构造完成。

(2) 构造并发结构补偿模型。图 2(b) 中实线部分表示的是一个并发结构  $t_1 \parallel t_2 \parallel \dots \parallel t_n$  的基本流程图,  $T^b = \{t_1, t_2, \dots, t_n\}, P^b = \{p_1, p_2, \dots, p_{2n-1}, p_{2n}\}, F = \{\langle p_1, t_1 \rangle, \langle t_1, p_2 \rangle, \dots, \langle p_{2n-1}, t_n \rangle, \langle t_n, p_{2n} \rangle\}$ 。对每一个变迁  $t_i (t_i \in T^b)$ , 向模型中添加变迁  $t_i$  对应的补偿变迁  $t_i^p$ , 得到  $T^p = \{t_1^p, t_2^p, \dots, t_n^p\}$ 。对每一个库所  $p_i (p_i \in P^b)$ , 向模型中添加  $p_i$  对应的补偿库所, 得到  $P^p = \{p_1^p, p_2^p, \dots, p_{2n-1}^p, p_{2n}^p\}$ 。对每一个弧  $\langle p_{2i-1}, t_i \rangle$ , 向模型中添加弧  $\langle t_i^p, p_{2i-1}^p \rangle$ , 对每一个弧  $\langle t_i, p_{2i} \rangle$ , 向模型中添加弧  $\langle p_{2i}^p, t_i^p \rangle$ , 得到  $F^p = \{\langle t_1^p, p_1^p \rangle, \langle p_2^p, t_1^p \rangle, \dots, \langle t_n^p, p_{2n-1}^p \rangle, \langle p_{2n}^p, t_n^p \rangle\}$ 。由公式  $preCon(a_i, cp()) = (\bigvee_{j \neq i} (pstCon_{||}(a_j)) \vee preCon(a_1 \parallel a_2 \parallel \dots \parallel a_n)) \wedge a_i, c$ , 为每一个活动  $t_i$  添加补偿触发条件  $(\bigvee_{j \neq i} (pstCon_{||}(t_j)) \vee preCon(t_1 \parallel t_2 \parallel \dots \parallel t_n)) \wedge t_i, c$ 。当所有的补偿条件添加完毕时,模型构造完成。

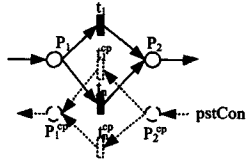
(3) 构造选择结构补偿模型。图 2(c) 中实线部分表示的是一个选择结构  $t_1 \otimes t_2 \otimes \dots \otimes t_n$  的基本流程图,  $T^b = \{t_1, t_2, \dots, t_n\}, P^b = \{p_1, p_2\}, F = \{\langle p_1, t_1 \rangle, \langle t_1, p_2 \rangle, \dots, \langle p_1, t_n \rangle, \langle t_n, p_2 \rangle\}$ 。对每一个变迁  $t_i (t_i \in T^b)$ , 向模型中添加变迁  $t_i$  对应的补偿变迁  $t_i^p$ , 得到  $T^p = \{t_1^p, t_2^p, \dots, t_n^p\}$ 。对每一个库所  $p_i (p_i \in P^b)$ , 向模型中添加  $p_i$  对应的补偿库所  $p_i^p$ , 得到  $P^p = \{p_1^p, p_2^p\}$ 。对每一个弧  $\langle p_1, t_i \rangle$ , 向模型中添加弧  $\langle t_i^p, p_1^p \rangle$ , 对每一个弧  $\langle t_i, p_2 \rangle$ , 向模型中添加弧  $\langle p_2^p, t_i^p \rangle$ , 得到  $F^p = \{\langle t_1^p, p_1^p \rangle, \langle p_2^p, t_1^p \rangle, \dots, \langle t_n^p, p_1^p \rangle, \langle p_2^p, t_n^p \rangle\}$ 。由公式  $preCon(a_i, cp()) = pstCon(succAction), preCon(a_1 \otimes a_2 \otimes \dots \otimes a_n) = pstCon(succAction)$ , 为选择结构整体添加补偿触发条件  $pstCon(succAction)$ 。



(a)顺序结构补偿模型



(b)并发结构补偿模型



(c)选择结构补偿模型

图2 3种基本流程结构的补偿模型

## 6 实例分析

本节将通过一个经简化的旅行在线服务系统描述模型的构造过程。该系统业务流程结构如图3所示,它提供4种在线服务:预订机票服务(fl\_bk)、预订火车票服务(tr\_bk)、预订酒店服务(ho\_bk)和在线支付服务(pay)。整个流程的执行过程为:(1)用户向系统提交预订请求;(2)系统根据用户请求,为用户预订飞机票或火车票,同时为用户预订酒店;(3)用户通过在线支付方式向服务提供者支付全部费用,流程结束。根据实际生活中的场景,用户存在退订服务的需求;系统在执行过程中,也会因流程的活动失效,需补偿已完成服务。例如,当前已经预定到飞机票,但酒店预订服务失效,则导致整个流程失效,这时需要取消已预订到的飞机票,即需要补偿预订飞机票服务。因此,在流程开发阶段,选用的4种服务均具有可补偿事务属性。该实例的建模过程如下所述。

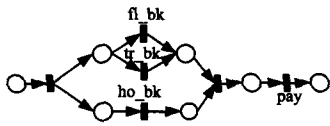


图3 旅行在线服务系统业务流程 Petri 表示

(1)计算流程中各活动补偿触发前提条件。采用第4节中补偿前提条件的计算方法,计算流程中各活动补偿触发的前提条件,计算结果如表1所列。

表1 旅行在线服务系统流程中各活动补偿活动的触发条件

| 活动    | 补偿活动   | 补偿前提条件  |
|-------|--------|---|
| pay   | pay_cp | user_cancel   |
| ho_bk | ho_cp  | $((fl\_bk\_fa \wedge tr\_bk\_fa) \vee fl\_bk\_cp \vee tr\_bk\_cp \vee pay\_fa \vee pay\_cp) \wedge ho\_bk\_c$ |
| fl_bk | fl_cp  | $(ho\_bk\_fa \vee ho\_bk\_cp \vee pay\_fa \vee pay\_cp) \wedge fl\_bk\_c$                                     |
| tr_bk | tr_cp  | $(ho\_bk\_fa \vee ho\_bk\_cp \vee pay\_fa \vee pay\_cp) \wedge tr\_bk\_c$                                     |

详细的计算过程如下:

用户取消订单

$$1) preCon(pay\_cp()) = pstCon(succAction)$$

$$\Rightarrow preCon(pay\_cp()) = user\_cancel$$

$$\Rightarrow pstCon(pay) = pay\_fa \vee pay\_cp$$

2)该部分为复合结构: $(fl\_bk \otimes tr\_bk) \parallel ho\_bk$

由公式  $preCon(a_1 \parallel a_2 \parallel \dots \parallel a_n) = pstCon(succAction)$

$$pstCon_{\parallel}(a_i) = a_i\_fa \vee a_i\_cp$$

$$preCon(a_i\_cp()) = (\bigvee_{j \neq i} (pstCon_{\parallel}(a_j)) \vee preCon(a_1 \parallel a_2 \parallel \dots \parallel a_n)) \wedge a_i\_c$$

可得:

$$preCon(ho\_cp) = (pstCon_{\parallel}(fl\_bk \otimes tr\_bk) \vee pstCon(pay\_cp)) \wedge ho\_bk\_c$$

$$\text{由公式 } pstCon(a_1 \otimes a_2 \otimes \dots \otimes a_n) = (\bigwedge_{i=1}^n a_i\_fa) \vee (\bigvee_{i=1}^n a_i\_cp) \text{ 可得:}$$

$$preCon(ho\_cp) = ((fl\_bk\_fa \wedge tr\_bk\_fa) \vee fl\_bk\_cp \vee tr\_bk\_cp \vee pay\_fa \vee pay\_cp) \wedge ho\_bk\_c$$

由公式  $preCon(a_1 \parallel a_2 \parallel \dots \parallel a_n) = pstCon(succAction)$

$$pstCon_{\parallel}(a_i) = a_i\_fa \vee a_i\_cp$$

$$preCon(a_i\_cp()) = (\bigvee_{j \neq i} (pstCon_{\parallel}(a_j)) \vee preCon(a_1 \parallel a_2 \parallel \dots \parallel a_n)) \wedge a_i\_c$$

可得:

$$preCon(fl\_bk \otimes tr\_bk) = (pstCon_{\parallel}(ho\_bk) \vee pstCon(pay\_cp)) \wedge (fl\_bk \otimes tr\_bk)\_c$$

$$\Rightarrow preCon(fl\_bk\_cp()) = (ho\_bk\_fa \vee ho\_bk\_cp \vee pay\_fa \vee pay\_cp) \wedge fl\_bk\_c$$

$$\Rightarrow preCon(tr\_bk\_cp()) = (ho\_bk\_fa \vee ho\_bk\_cp \vee pay\_fa \vee pay\_cp) \wedge tr\_bk\_c$$

(2)根据步骤(1)中得到的补偿触发条件,经过第5节中模型的3个构造步骤,容易得到该实例的基于补偿的异常处理模型,详细构造过程不再赘述,最终生成的模型如图4所示。

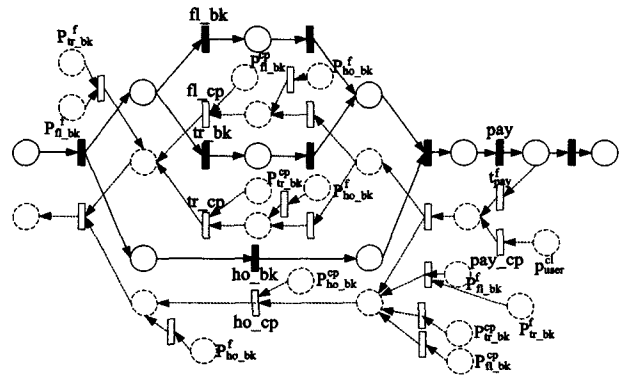


图4 基于补偿机制的异常处理模型

结束语 本文在已有工作的基础上,通过分析顺序、选择、并发等几种基本流程结构及其复合结构中的补偿依赖,给出了补偿活动触发条件计算方法,进一步给出补偿事务的Petri网模型设计过程,通过实例描述该模型的运用。补偿活动触发条件计算方法和补偿事务模型建模方法,能在业务流程异常处理设计中为业务流程开发人员提供技术支持。由于该模型是基于着色Petri网的,因此,可直接使用已有的Petri网验证工具(如CPNTools),验证模型的活性、有界性和可达性等正确性属性,验证正确后的模型可用于基于模型的流程异常处理设计。下一步工作主要研究如何优化模型的建模方

(下转第113页)

- of large-scale data sets of protein-protein interactions [J]. *Nature*, 2002, 417: 399-403
- [2] Watts D J, Strogatz S H. Collective dynamics of ‘small-world’ networks [J]. *Nature*, 1998, 339: 440-442
- [3] Barabasi A L, Oltvai Z N. Network biology: understanding the cell’s functional organization [J]. *Nature Review Genetics*, 2004, 5(2): 101-103
- [4] MacQueen J. Some methods for classification and analysis of multivariate observations [C]// *Proceedings of the 5<sup>th</sup> Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, University of California Press, 1967: 281-297
- [5] Guo Hai-xiang, Zhu Ke-Jun, Gao Si-wei, et al. An improved genetic K-means algorithm for optimal clustering [C]// *Proceedings of 6<sup>th</sup> IEEE International Conference on Data Mining Workshops*. Washington DC: IEEE Computer Society, 2006: 793-797
- [6] Zhang T, Ramakrishnan R, Livny M. An efficient data clustering method for very large databases [C]// *Proceeding ACM SIGMOD Conference on Management of Data*. Montreal, Canada, 1996: 103-114
- [7] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in Large Spatial Databases with Noise [C]// *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD’96)*. Portland, OR, 1996: 226-231
- [8] Ben-Hur A, Horn D, Siegelmann T H, et al. Support vector clustering [J]. *Journal of Machine Learning Research*, 2002, 2: 125-137
- [9] Tipping M E. The Relevance Vector Machine [M]// *Advances in Neural Information Processing Systems 12*. 2000: 652-658
- [10] Dongen S V. *Graph Clustering by Flow Simulation* [D]. Utrecht, University of Utrecht, 2000
- [11] Ruan Pei-ying, Hayashida M, Maruyama O, et al. Prediction of heterotrimeric protein complexes by two-phase learning using neighboring kernels [C]// *BMC Bioinformatics*. 2014, 15, S2: S6
- [12] Satuluri V, Parthasarathy S. Scalable graph clustering using stochastic flows: applications to community discovery [C]// *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA, 2009: 737-746
- [13] Satuluri V, Parthasarathy S, Ucar D. Markov clustering of protein interaction networks with improved balance and scalability [C]// *Proceeding BCB’10 Proceedings of the 1th ACM International Conference on Bioinformatics and Computational Biology*. New York, NY, USA, 2010: 247-256
- [14] Shih Y K, Parthasarathy S. Identifying functional modules in interaction networks through overlapping Markov clustering [J]. *Bioinformatics*, 2012, 28(18): i473-i479
- [15] Shih Y K, Parthasarathy S. Identifying protein complexes in AP-MS data with negative evidence via soft Markov clustering [C]// *Proceeding BCB’13 Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*. New York, NY, USA, 2013: 440
- [16] 岑涌, 罗林开. 一种改善非平衡分布数据 SVM 分类能力的新策略 [J]. *计算机与数学工程*, 2006, 34(11): 103-105  
Cen Yong, Lou Lin-kai. A novel strategy for improving the performance of SVM classification for unbalance distribution data [J]. *Computer & Digital Engineering*, 2006, 34(11): 103-105
- [17] 方霞, 席金菊. 基于变异和启发式选择的蚁群优化算法 [J]. *计算机工程与应用*, 2013, 49(24): 24-27  
Fang Xia, Xi Jin-ju. Ant colony algorithm based on mutation features and selected heuristic [J]. *Computer Engineering and Applications*, 2013, 49(24): 24-27
- [18] Güldener U, Munsterkotter M, Kastenmuller G, et al. CYGD: the Comprehensive Yeast Genome Database [J]. *Nucleic Acids Research*, 2005, 33(1): 364-368
- [19] Zhang Ai-dong. *Protein Interaction Networks* [M]. UK: Cambridge University Press, 2009
- [20] 尤梦丽, 雷秀娟. PPI 网络聚类的评价方法的研究与应用 [J]. *计算机科学*, 2013, 40(12): 254-258  
You Meng-li, Lei Xiu-juan. Study and application of evaluating methods of PPI network clustering [J]. *Computer Science*, 2013, 40(12): 254-258

(上接第 94 页)

法, 通过状态合并等方式压缩模型的状态空间。

### 参 考 文 献

- [1] Colombo C, Pace G J. Recovery within Long-Running Transactions [J]. *ACM Computing Surveys*, 2013, 45(3): 1-28
- [2] Haddad J E, Manouvrier M, Rukoz M. TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition [J]. *IEEE Transactions on Services Computing*, 2010, 3(1): 73-85
- [3] Liu Hai, Zhang Wei-min, Ren Kai-jun, et al. A Novel Selection Approach for Transactional Web Services Composition [C]// *2010 Ninth International Conference on Grid and Cloud Computing*. Nanjing, China, 2010: 450-456
- [4] Kokash N, Arbab F. Formal Design and Verification of Long-Running Transactions with Extensible Coordination Tools [J]. *IEEE Transactions on Services Computing*, 2013, 6(2): 186-200
- [5] Gaaloul W, Gaaloul K, Sami Bhiri, et al. Log-based transactional workflow mining [J]. *Distributed and Parallel Databases*, 2009, 25(3): 193-240
- [6] Zhang Hua, Wang Qian. Implementing mechanism of service oriented workflow compensation [J]. *Journal of Southeast University (Nature Science Edition)*, 2009, 39(1): 40-46
- [7] Zhao Zong-tao, Jun Wei, Li Lin, et al. A concurrency control mechanism for composite service supporting user-defined relaxed atomicity [C]// *The 32nd Annual IEEE International Computer Software and Applications Conference*. Turku, Finland, 2008: 275-278
- [8] Bhiri S, Godart C, Perrin O. Transactional Patterns for Reliable Web Services Compositions [C]// *Proceedings of the 6th international conference on Web engineering*. California, USA, 2006: 137-144