

基于配置的交互流程变化传播分析

詹 悦 方贤文 王丽丽

(安徽理工大学数学与大数据学院 安徽 淮南 232000)

摘 要 变化传播是业务管理系统的核心之一,旨在灵活适应变化的商务需求。然而,已有的变化传播主要处理由单个业务流程拓展的相似流程间的变化域问题,关于信息传递的交互流程的相关研究还存在一定问题。文中提出了基于配置的交互流程变化传播分析方法,在寻找变化域的基础上利用配置技术完善域内行为关系。在给定变化需求锁定相应的源变化域的情况下,利用编排与条件抽象寻找其他交互子流程受变化影响的目标变化域,并在保证域外结构一致性的原则下利用配置处理域内带约束的变化行为关系,然后根据兼容性检测配置后的交互流程模型是否合理,最后以具体的案例验证了该方法的可行性。

关键词 变化传播,配置,变化类型,变化域,编排

中图法分类号 TP391.9 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.09.046

Analysis of Interactive Process Change Propagation Based on Configuration

ZHAN Yue FANG Xian-wen WANG Li-li

(School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan, Anhui 232000, China)

Abstract Change propagation is one of the core of the business management system, and aims at adapting flexibly to changing business requirements. However, the existing change propagation mainly deals with the problem of the change region between similar processes extended by a single business process, and it is difficult to study the interactive process with information transfer. A method was proposed to analyze interactive process change propagation based on configuration. Based on the location of the change region, the configuration technology is used to improve the intra-region behavior relationship. In the case of the corresponding source change region locked by a given change demand, choreography and conditional abstraction are used to find target change region affected by changes in other interactive sub-processes. And the configuration was utilized to deal with the change behavior of constraints in the region under the principle of ensuring the consistency of the extra-region structure. Then, the rationality of the configured interactive process model was detected by compatibility. Finally, the feasibility of the method was verified by a specific case.

Keywords Change propagation, Configuration, Change types, Change region, Choreography

1 引言

业务流程最优化设计与实现是业务管理系统的核心问题之一,它不仅应用于单个内部业务流程,还应用于多个合作方协同执行或控制的交互流程。在不断变化的商务环境下,对流程的优化与更新是必不可少的。由于交互流程模型集的变化会相互依赖,一个流程的变化可能会对一系列相关流程产生直接或间接的影响^[1],因此为了保证交互流程集的一致性与可兼容性以适应变化需求,变化传播的分析与优化显得尤为重要。

变化传播实质上是保证同一流程在不同层次的一致性,以及与相关流程保持行为与结构的可兼容性。文献[2]给出源模型的变化域,利用行为轮廓解决边界变迁和内部变迁减

少的问题,以确定目标模型的变化域。文献[3]在文献[2]的基础上做出了改进,利用行为轮廓的行为抽象获得与源流程行为不一致的对应流程变化域。Cupta 等^[4]于 2010 年提出了一种新的动态方法用于评估变化传播,即基于变化分类的算法对变化进行分析。为了避免利益相关者因不同视角执行变化而产生的不一致现象,Grossmann 等^[5]提出了基于业务流程设计的方法,该方法能有效检测过程视图在运行中的一致性,并利用 BPMN 语言展现不同视角变化传播的过渡模型。

已有研究主要集中在变化传播的业务流程的优化分析上,但对于带有配置信息的业务流程优化则存在不足。对于一个可配置的业务流程模型,其能够根据特定的要求使用配置信息进行定制化处理。

到稿日期:2018-07-11 返修日期:2018-09-17 本文受国家自然科学基金资助项目(61572035,61272153,61402011),安徽省自然科学基金资助项目(1508085MF111),安徽省高校自然科学基金资助项目(KJ2014A067,KJ2016A208)资助。

詹 悦(1994—),女,硕士,主要研究方向为 Petri 网,E-mail:2272350170@qq.com;方贤文(1975—),男,博士,教授,主要研究方向为 Petri 网和可信软件,E-mail:280060673@qq.com(通信作者);王丽丽(1982—),女,副教授,主要研究方向为业务流程分析和软件认证。

已有的配置信息包括允许、隐藏以及阻塞 3 类参数信息,通过隐藏或者阻塞流程模型的某一部分达到实现特定要求的目的,将模型需要或保留的行为默认为允许状态,而阻塞或隐藏不需要的行为。文献[6]在假定行为合理的前提下,对可配置的流程模型提出了 3 种变迁: allowed(允许)变迁、hidden(隐藏)变迁、blocked(阻塞)变迁。该配置参数仅针对单个变迁作用,不适用多个变迁组成的活动片段。文献[7]通过用配置对业务流程的活动、资源和数据对象进行不同的设置,从而满足个别用户的任务需求。该方法主要针对无约束的静态行为活动,对存在带约束条件的变化行为,很难反映出用户想要的状态和信息。文献[8]提出一种算法,该算法通过提取、聚类、合并一个特定活动的流程片段,来构造一个可配置的流程片段,并允许流程设计者灵活调整和指定流程可配置部分来达到设计要求。但是,该配置算法仅从业务流程的控制流角度分析,对数据流的考虑甚少。文献[9]在不限时间内选择 C-YAWL 语言作为基本概念,对业务流程的不同变量进行配置,以维护可适应的过程变体。但在设定时间阈值的情况下,其任务执行的复杂度与错误率将明显上升。文献[10]提出了一种新方法,该方法利用带预标注业务能力合并流程模型,使之成为一个可配置模型。然而,该方法需要对业务流程模型中活动的属性进行额外的统计分析,计算复杂度相对较高。以上方法都是基于常规的流程模型,实际上大部分业务流程模型都是基于描述模型交互的初衷来建立的。因此,对于交互协同方的不同需求,本文使用可配置的流程模型为不同利益相关者提供不同的信息。

在具有信息传递的交互流程中,某一子流程因协同方的特别要求而需要施加特定变化时,将使得交互流程面对两大挑战:1)基本的变化模式并不能处理此类带有约束条件的变化问题;2)一般相似流程间的变化传播方法难以确定交互流程内受变化影响的目标变化域。因此,在文献[11]处理过程编排中的变化问题的基础上,本文提出了基于配置的交互流程变化传播分析方法,利用编排模型(树)以及条件抽象与投影前后集的关系确定交互流程的目标变化域,并使用配置技术实现子流程域内带约束变化的行为关系,最后根据兼容性判断配置变化后的交互流程的合理性。

2 基本概念

定义 1^[8](流程模型 Petri 网) 满足以下 3 个条件的四元组 $PN=(P,T;F,C)$ 被称作流程模型 Petri 网:

- 1) P 是一个有限非空库所集, T 是一个有限非空变迁集;
- 2) $P \cup T \neq \emptyset, P \cap T = \emptyset$;
- 3) $F=(P \times T) \cup (T \times P)$ 表示流关系;
- 4) $C=\{seq, chc, par, rpt\}$ 为流程网的结构类型,其元素分别表示顺序、选择、并行、循环。

定义 2^[6](Petri 网配置) PN 是一个 Petri 网, $C_N \in T \rightarrow \{allow, hide, block\}$ 是网 PN 的一个配置。其中, $PN=(P,T,F)$, 定义如下:

$A_N^C=\{t \in T | C_N(t)=allow\}$ 作为所有的允许变迁集;

$H_N^C=\{t \in T | C_N(t)=hide\}$ 作为所有的隐藏变迁集;

$B_N^C=\{t \in T | C_N(t)=block\}$ 作为所有的阻塞变迁集。

其中,配置参数 $allow$ 作为默认参数,适用于所有的变迁集。

定义 3^[2](变化域) 流程模型 Petri 网 $PN=(P,T;F,C)$, 若变化部分为 $PN_i=(P_i,T_i;F_i,C_i)$, 则 PN_i 是 PN 的一个子模块,即 $P_i \subseteq P, T_i \subseteq T, F_i \subseteq F, C_i \subseteq C, PN_i \subseteq PN$ 。

定义 4^[11](编排模型(树)) 一个编排模型结构树 G 表示交互流程 N 的全局视图,仅包含交互行为:

$G::=G_{node}$

$G_{node}::=I(Sender, Receiver, Message) | C_n$

$C_n::=SEQ(\{G_{node}\}) | CHC(\{G_{node}\}) | PAR(\{G_{node}\}) |$

$RPT(\{G_{node}\})$

其中, I_a 表示交互活动, C_n 表示控制节点, E 表示事件, SEQ 表示顺序关系, CHC 表示选择关系, PAR 表示并发关系, RPT 表示循环关系。

定义 5^[11](编排) 一个五元组 $C=(G,P,N,\varphi,\xi)$ 为一个编排,其中:

- 1) G 为编排模型(树);
- 2) P 为所有合作执行方集合;
- 3) $N=\{l_p\}_{p \in P}$ 为各个执行方子模型构成的交互模型;
- 4) $\varphi: l \leftrightarrow l'$ 为不同执行方子模型节点间的映射函数;
- 5) $\xi: G \leftrightarrow l$ 为编排模型与交互模型节点间的映射函数。

定义 6(共轭函数) 假定 $a \in P$ 对应于一个 P 与 P' 的交互活动,相反, a 的共轭函数 $\tau(a)=\overline{a} \in P'$ 表示 P' 与 P 的交互活动。如: $\tau(send(m, P'))=\overline{send(m, P')}=receive(m, P)$; $\tau(receive(m, P'))=\overline{receive(m, P')}=send(m, P)$ 。

当 l_p 流程内存在一个交互活动发送信息 m 给 P' 时,则同样地,在 $l_{p'}$ 中应有一个具有相同结构的交互活动来接收信息 m ,以保证交互流程的兼容性。

3 基于配置的交互流程变化传播

由于交互流程模型集的变化会相互依赖,一个流程的变化可能会影响另一个甚至一系列的相关流程。因此,交互流程变化传播将面临两大挑战:寻找受变化影响的目标变化域,确定域内变化的行为关系。特别地,对于变化域内存在带有约束的变化问题,普通的变化模式已难以满足该类行为关系。本文针对交互流程中带有约束的变化及其传播问题,主要通过以下步骤进行处理:1)配置该约束变化;2)判断该变化是否需要传播;3)计算直接或间接变化传播;4)寻找目标变化域;5)配置目标变化;6)计算兼容性。图 1 为粗略的交互流程变化传播流程图。其中,步骤 1)与步骤 5)利用配置技术处理变化方以及协同变化方的变化域内行为变化关系。

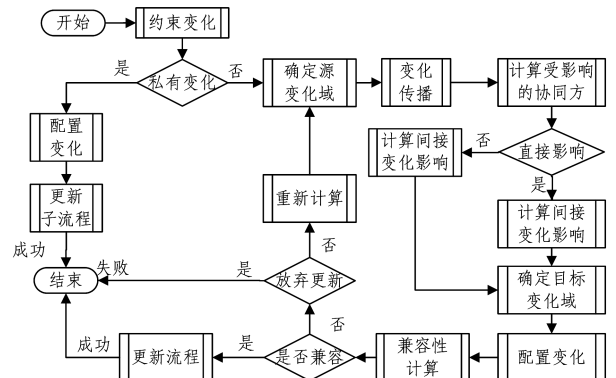


图 1 交互流程变化传播流程图

Fig. 1 Flow of interactive process change propagation

3.1 基于条件抽象的目标变化域的确定

由于交互流程由若干个子流程构成,当根据变化需求得到源变化子流程后,应进一步判断其变化是否会影响其他子流程,以及具体的变化影响途径或传播方式。因此,交互流程的变化传播的核心问题之一是寻找受变化影响的对应子流程目标变化域。

定义 7(条件抽象函数) 当满足以下条件时,一个抽象函数 $abstr_{\lambda}: \sigma \rightarrow \sigma'$ 表示一个模型 σ 根据抽象条件 λ 映射为模型 σ' 。

- 1) $\forall n \in \sigma$, 且满足 $\lambda \Rightarrow n \in \sigma'$ (n 为节点)。
- 2) $\forall n, n' \in \sigma \wedge n \rightarrow n'$ (n 为 n' 的前节点), 且满足 $\lambda \Rightarrow n, n' \in \sigma' \wedge n \rightarrow n'$ 。

当 $\lambda = P'$ 时, $abstr_{P'}(l_P)$ 表示交互子流程 l_P 内与协同方 P 和 P' 存在交互行为的抽象活动,可获得所有 P 与 P' 交互行为的视图。当 $\lambda = inter$ 时, $abstr_{inter}(l_P)$ 表示所有与 P 存在交互行为的抽象活动。因此,该条件抽象函数可以计算交互流程中存在信息传递的协同方以及具体的交互活动。

同样地,如果待执行的系列变化活动(或活动集合)具有信息的传递作用,则该变化必然会引起其他协同方子流程的同步变化。由于一般的变化模式都可由基本删除和添加组合完成,而这两种基本的变化模式产生的变化域分析方法存在很大差异,因此本文将交互流程的变化传播类型划分为两类。

1) 带约束删除模式的变化传播流程如图 2 所示。步骤 2 主要通过抽象函数 $abstr_{inter}(F)$ 分解确定所需要传播的变化片段 $F^1 = S(m_1, P_2) + R(m_2, P_3)$, 因活动 a 为该交互子流程 l_{P_1} 的私有变化活动,故不会影响其他子流程且不需要传播。步骤 3 以各个协同方作为约束条件来寻找交互子流程的目标变化域,由于删除模式仅涉及被删除活动本身,因此查找与源变化活动存在信息传递的其他子流程的交互活动,它们构成的域即为目标变化域。如 $abstr_{P_2}(F^1) = S(m_1, P_2)$, 可确定在协同方 l_{P_2} 内存在接收信息 m_1 的交互活动即为变化活动,利用共轭函数得到对应流程的交互活动 $F^2 = \tau(S(m_1, P_2)) = \overline{S(m_1, P_2)} = R(m_1, P_1)$ 及 $F^3 = \tau(R(m_2, P_3)) = \overline{R(m_2, P_3)} = S(m_2, P_1)$ 。最后通过配置参数实现带约束变化的传播问题。

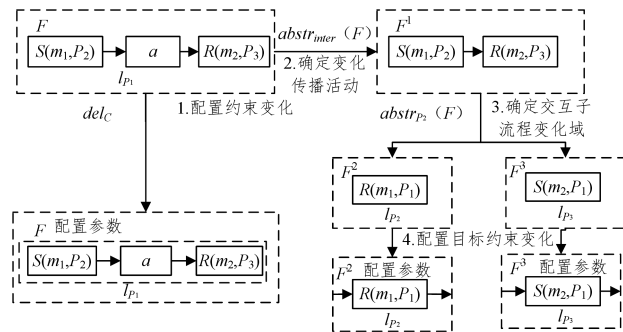


图 2 带约束删除模式的变化传播图

Fig. 2 Change propagation diagram with constrained deletion pattern

带约束删除模式的目标变化域直接由活动自身确定,若存在的交互活动在约束条件下被删除(隐藏),则其具有信息传递的协同交互活动也必然存在,目标变化域便被直接锁定。

但寻找带约束添加模式的目标变化域不仅关系其变化活动本身,还涉及到其插入的位置即前后集的活动。如果在模型(树) σ 中的一个节点为 n , 则其前集、后集分别为:

$$preset(n, \sigma) = \{n' \in \sigma \mid \exists SEQ(n', n) \in \sigma\}$$

$$sucset(n, \sigma) = \{n' \in \sigma \mid \exists SEQ(n, n') \in \sigma\}$$

交互活动能够通过条件抽象函数逐一对应,但由于其直接的前后集并不一定都属于交互活动,因此在给定源变化域的前提下,对目标变化域的确定就是对变化的交互活动前后集的提取与确定。

定义 8^[10](投影前集(后集)) 给定投影条件 λ , 满足 λ 的节点 n 的投影前集(后集)为:

$$T_preset_{\lambda}(n, \sigma) = preset(n, abstr_{\lambda}(\sigma))$$

$$T_sucset_{\lambda}(n, \sigma) = sucset(n, abstr_{\lambda}(\sigma))$$

例:在图 8 中, $T_preset_{商家}(快递要求, G) = \{下单\}$, $T_postset_{快递公司}(快递要求, G) = \{通知发货\}$ 。

2) 带约束添加模式的变化传播流程如图 3 所示,其与带约束删除模式的步骤 1—步骤 3 及步骤 6 相同,但目标变化域的确定更加复杂。步骤 4 根据定义 5 中 $\zeta: G \leftrightarrow l$ 的节点映射关系以及前后集的映射关系,可以确定该添加片段在编排模型(树)内的变化域,即 $pred_G = \zeta(pred)$, $succ_G = \zeta(succ)$ 。步骤 5 包含两个部分,首先根据定义 8(投影前后集)可抽象编排模型(树)内的变化域,使其仅包含某目标协同方的变化域,即 $pred_{2_G} = T_preset_{P_2}(pred_G)$, $succ_{2_G} = T_sucset_{P_2}(succ_G)$;然后再利用 $\zeta: G \leftrightarrow l$ 的关系,确定目标交互子流程的变化域,即 $pred_2 = \zeta(pred_{2_G})$, $succ_2 = \zeta(succ_{2_G})$ 。以同种分析方式确定其他协同方的目标变化域。

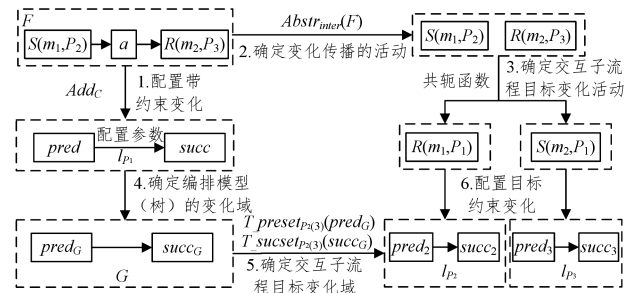


图 3 带约束添加模式的变化传播图

Fig. 3 Change propagation diagram with constrained addition pattern

3.2 业务流程配置分析

3.2.1 带约束的变化类型

根据不断变化的业务需求,在指定流程中需要作出相应调整,以响应变化。对于交互流程,3.1 节的两种分类方法已经能够确定受变化影响的源变化域以及目标变化域,但由于域内的行为关系是带有约束条件的,而基本的变化类型如添加(Add)、删减(Delete)和替代(Replace)等已无法满足这些带有特定约束条件的变化操作。比如,流程中新增加的局部变化过程仅在某一特定约束条件下才被执行,一般情况下,该局部流程可以直接跳过,仅仅使用 Add 模式并不满足此类情况需求。为了更准确地执行该类变化操作,对基本变化模式 CP(添加(Add)、删减(Delete)(其他变化模式可由添加与删除组合实现))附加一定的约束条件使其成为约束变化,如:

$CP_C ::= Add_C(Fragment, pred, succ, Constraint) \mid Del_C(Fragment, Constraint)$.

上述约束变化表示约束条件 $Constraint$ 成立时,该变化才被执行,否则按原流程执行方式进行。这有助于版本更新或者系统升级时兼容原有模式,而并不是简单的取代。

3.2.2 使用配置参数处理域内带约束的变化行为关系

定义 9(可配置的约束变化) 一个映射函数 $\pi: C \rightarrow CP_C$ 为约束变化的配置,其中 $C \subseteq \{allow, hide, block\}$ 是配置参数, $allow$ 为默认参数。

根据不同结构的约束变化应设置不同的配置参数,顺序或并发结构的带约束添加与带约束删除可使用 $hide$ 配置参数处理,而选择结构的添加与删除则由 $block$ 配置参数处理。可配置的约束变化如图 4 所示(其中,元素 A, B, F 等既可以代表过程片段,也可以代表单个活动)。 $\pi: hide \rightarrow Add_C$, $\pi: hide \rightarrow Del_C$, $\pi: block \rightarrow Add_C$ 以及 $\pi: block \rightarrow Del_C$ 。

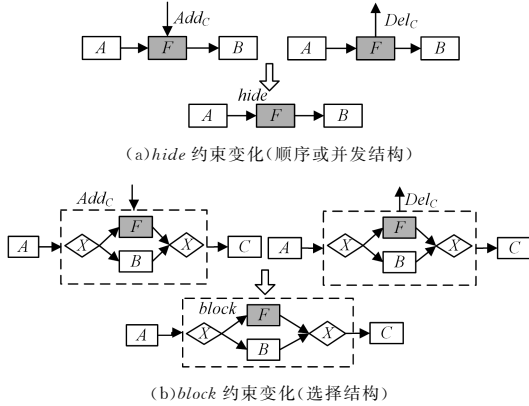


图 4 可配置的约束变化

Fig. 4 Configurable constraint changes

假设在流程内处于顺序或并发关系的元素 A 与 B 间按照特定约束条件 $Constraint$ 添加元素 F ,如图 4(a)所示,满足条件则按照 $A \rightarrow F \rightarrow B$ 的行为关系执行该局部流程;在不满足约束条件的情况下,跳过 F ,按原行为关系 $A \rightarrow B$ 执行。因此,为了在流程内处理该类约束条件的变化行为关系,可配置参数 $hide$ 能够将元素 F 的执行与否与其约束条件逐一对应,将两种行为关系在同一个流程内显示并按照约束条件选择执行。同理,对于带有约束条件的删除操作,配置参数 $hide$ 也可保证删除元素 F 前后的行为关系都存在。

对于带约束条件的变化元素 F ,若在添加或删除情况下都与元素 B 构成选择结构,则无法判断该选择结构对应的具体行为发生关系,如图 4(b)所示。如变化的约束条件成立时,分支结构只允许 $A \rightarrow F \rightarrow C$ 的行为发生;条件不成立时,只允许 $A \rightarrow B \rightarrow C$ 的行为关系存在。一般地,选择分支结构对元素 F 与 B 的选取是随机的,因此无法处理该类带约束条件的非随机选择结构。配置参数 $block$ 则可以有效地处理该类变化,可以按照约束条件强制选择某一分支结构,当条件不成立时,分支结构 $A \rightarrow F \rightarrow C$ 是被阻塞的不可选分支。

3.3 直接与间接变化传播

在实际商业环境中,参与交互流程的各个协同方,有的可以直接进行信息交流,而有的则需要借助中间协同方进行信息的传递。3.1 节分析的交互流程内变化传播问题都是基于

直接信息传递的,对于借助中间协同方进行信息传递的间接变化传播问题,本文利用编排模型(树)以及条件抽象函数方法,将其分解为直接变化传播问题进行处理。

如图 5 所示,协同方 P_1 需要发送某一信息 m 至协同方 P_3 ,表示为 $Send(m, P_3, P_1)$,但由于 P_1 与 P_2 无法直接交流,需要借助中间方 P_2 ,因此 P_2 存在既能接收从 P_1 发送的信息 m 又能将该信息发送至 P_3 的交互活动。

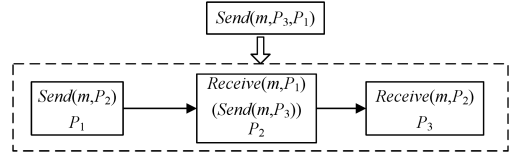


图 5 间接信息传播图

Fig. 5 Indirect information transfer diagram

存在间接信息传播的交互流程,其内部的信息传递必须满足以下几点:

- 1) $Send(m, P_3, P_1) = Send(m, P_2, P_1) + Send(m, P_3, P_2)$ 。
- 2) $Receive(m, P_1, P_3) = Receive(m, P_1, P_2) + Receive(m, P_2, P_3)$ 。
- 3) $Receive(m, P_1, P_2) = Send(m, P_3, P_2)$ 。
- 4) $\tau(Send(m, P_2, P_1)) = Send(m, P_2, P_1) = Receive(m, P_1, P_2)$ 。
- 5) $\tau(Send(m, P_3, P_2)) = Send(m, P_3, P_2) = Receive(m, P_2, P_3)$ 。
- 6) $\tau(Send(m, P_3, P_1)) = Send(m, P_3, P_1) = Receive(m, P_1, P_3)$ 。

同样地,如果带约束的变化交互活动传递的信息需要借助中间协同方,则对应变化传播也延伸为间接变化传播。如果编排模型中存在多个传递信息相同且属于顺序关系的节点,则其为变化的间接信息传递节点,即 $\exists n, n' \in \sigma, abstr_\lambda(\sigma) = \{n, n'\} \Rightarrow SEQ(n, n')$,其中 λ 为变化活动。

3.4 基于配置的交互流程变化传播算法

本文提出的变化传播算法主要从两个方面处理变化的交互流程(见算法 1):1)根据编排与条件抽象关系确定受变化影响的交互子流程的目标变化域;2)使用配置技术对域内结构进行调整,以满足带约束条件的变化行为关系。该算法以交互流程 N 、带约束变化元素 F 以及交互全局视图模型 G 为输入,其中元素 F 表示单个活动或多个活动组成的片段,输出结果为各个交互子流程同步更新后的完整流程 N' 。第 4—5 行分解变化元素 F ,确定是否包含交互活动,若不包含,则仅配置更新该子流程并输出最终更新流程 N' ,否则变化需要传播至其他受影响的子流程;第 7 行利用行为抽象投影确定交互变化活动所涉及的各协同变化方,即变化需要传播的目标子流程;第 8 行根据编排模型中传递信息相同的节点关系,来判断变化传播的直接与间接性,并将存在的间接变化传播问题分解为单个的直接变化传播进行处理,然后按照变化的类型使用不同的直接变化传播方式;第 9—10 行为带约束的删除模式变化传播;第 11—15 行为带约束的添加模式变化传播,两者均包含变化域的寻找以及域内行为关系的确定;最后更新所有的子流程并输出结果。

算法 1 基于配置的变化传播算法

1. Input: G, N, F
2. Output: N'
3. begin:
4. if $(\text{abstr}_{\text{inter}}(F) = \emptyset)$ then
5. $\pi: C \rightarrow F, l_p' \leftarrow l_p \Rightarrow N' \leftarrow N$ //配置变化,仅更新子流程
6. else //变化传播
7. $\sum F_i' \leftarrow \text{abstr}_{\text{inter}}(F), P_i' \leftarrow \{F_i' \text{ 所涉及的协同合作方} \}$
8. for $(|\text{abstr}_{F_i'}(G)| \geq 2)$ do { $\sum_{j=1}^{|\text{abstr}_{F_i'}(G)|} P_j \leftarrow P_i' \}$ //将间接变化传播分解为直接变化传播
9. if $\text{CP}_C(F_i') = \text{del}_C$ then //带约束的删除模式变化传播方式
10. for each $P' \in P_j \cup P_i'$ do $\{F_{P'}' = \text{abstr}_{P'}(F_i'), \pi: C \rightarrow \text{del}_C(F_i')\}$
11. else $\text{CP}_C(F_i') = \text{add}_C$ then //带约束的添加模式变化传播方式
12. for each $P' \in P_j \cup P_i'$ do $\{F_{P'}' = \text{abstr}_{P'}(F_i'), \text{pred}_G =$

$\zeta(\text{pred}), \text{succ}_G = \zeta(\text{succ})$
13. $\text{pred}_{P_G}' = T_{\text{preset}_P}(\text{pred}_G), \text{succ}_{P_G}' = T_{\text{succ}_P}(\text{succ}_G)$
14. $\text{pred}' = \zeta(\text{pred}_{P_G}'), \text{succ}' = \zeta(\text{succ}_{P_G}')$ //确定目标交互流程的变化域
15. $\pi: C \rightarrow \text{add}_C(F_i', \text{pred}', \text{succ}')$ //利用配置参数处理域内带约束的行为关系
16. end
17. $l_{p'}' \leftarrow l_{p'} \Rightarrow N' \leftarrow N$ //更新流程
18. end
19. end

4 实例分析

本文采用移动电子商务中的购物交易作为实例进行分析,如图 6 所示。某交易流程包括 3 个交互子流程,分别是顾客(P_1)操作流程、商家(P_2)操作流程以及快递公司(P_3)操作流程。字母所代表的具体变迁意义如表 1 所列。

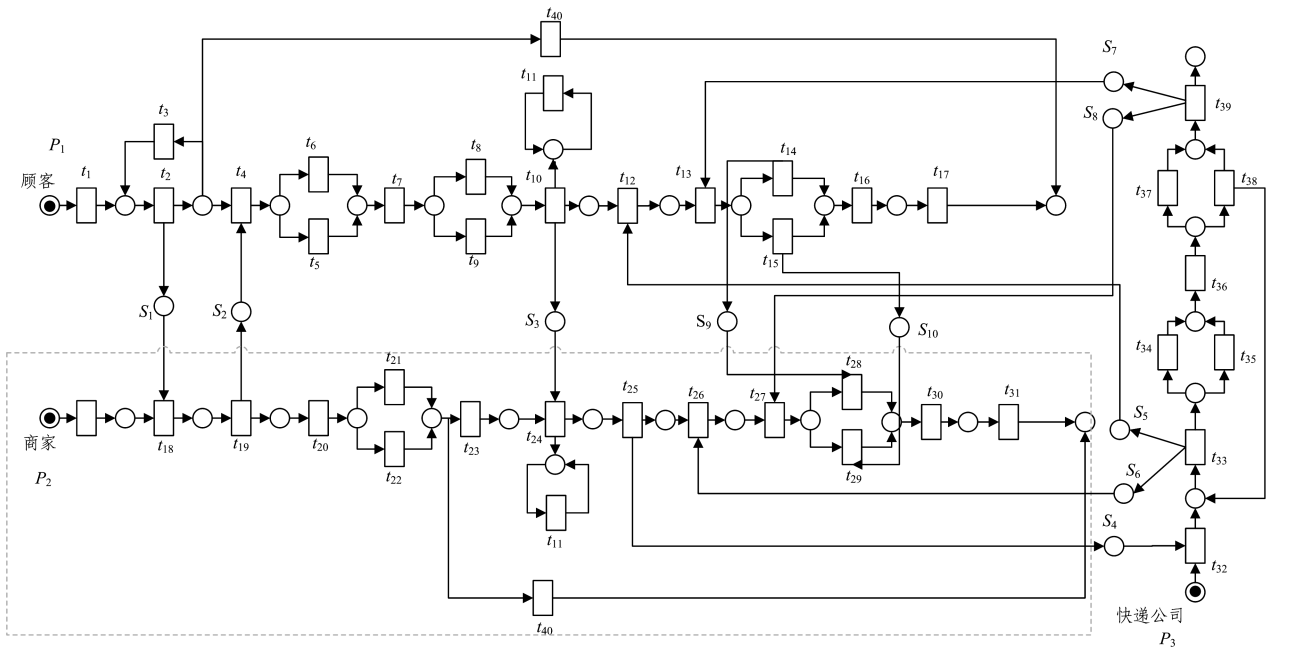


图 6 交易流程图
Fig. 6 Transaction flow diagram

表 1 字母代表的活动事件

Table 1 Activity events represented by letters			
字母	活动名称	字母	活动名称
t ₁	挑选商品	t ₁₅	退货
t ₂	发送订单信息	t ₁₆	接收评价请求
t ₃	重新选择	t ₁₇	评价并反馈
t ₄	要求身份验证	t ₁₈	接收订单消息
t ₅	会员享有优惠	t ₁₉	确认订单
t ₆	普通价	t ₂₀	查看库存
t ₇	准备下单	t ₂₁	进货
t ₈	货到付款	t ₂₂	库存足够
t ₉	在线支付	t ₂₃	默认快递
t ₁₀	确定订单并提交	t ₂₄	接收确定订单
t ₁₁	查询状态	t ₂₅	通知快递公司
t ₁₂	已发货通知	t ₂₆	已发货通知
t ₁₃	收货通知	t ₂₇	收货通知
t ₁₄	确认付款	t ₂₈	确认收款
		t ₂₉	接收退货
		t ₃₀	请求评价
		t ₃₁	接收评价反馈
		t ₃₂	接收订单
		t ₃₃	发货通知
		t ₃₄	货车运送
		t ₃₅	其他方式运送
		t ₃₆	货物状况
		t ₃₇	未损坏
		t ₃₈	损坏或丢失
		t ₃₉	通知收货
		t ₄₀	取消订单

图 6 所示为网上购物的基本流程,但无法满足因商业变动或人为因素导致的变化需求。例如,某顾客因收货地址偏远,仅支持特定的快递类型,因此该顾客将直接对商家指定快递发货要求。而一般默认情况下,商家都会选用默认的快递方式。另外,也存在发货后顾客发现地址有误,从而需要立即修改并通知快递公司更改路线的情况,但顾客并不能直接发送消息给快递公司,需要通过中间交互方即商家反馈给快递公司进行相应处理。因此,在交互流程中当某一执行方存在变化时,必将影响其他相关流程的进展,如第一种影响即为直接影响,后者属于间接影响。对于以上两种变化情况,本文算法能够有效地优化处理该流程内的变化传播问题,从而更好地适应变化环境。

由案例可知,在当前基本的购物流程上存在两处带约束的变化,分别为 $Add_C(X, t_7, t_8, C_X)$ (其中, $X = send$ (某指定

快递,商家,顾客), C_X 为指定某一类快递类型)和($Add_C(Y, t_{12}, t_{13}, C_Y)$ (其中, $Y=send$ (新地址,快递公司,顾客), C_Y 为修改地址发生在发货之后的一段时间内)。

根据定义 9 对上述两个带约束的变化采取对应的配置措施,得到如图 7 所示的顾客子流程,其中阴影部分表示源变化域。

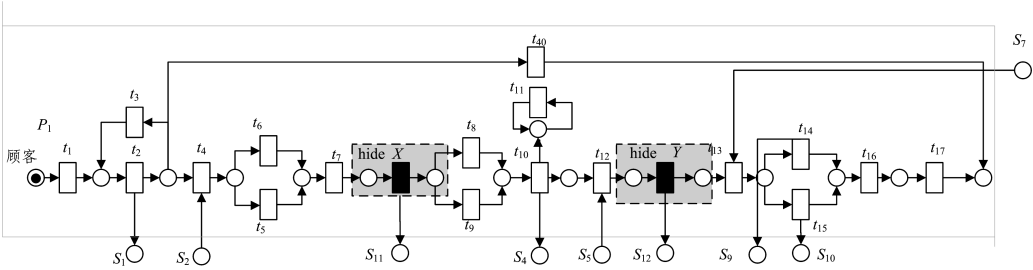


图 7 更新的顾客子流程

Fig. 7 Updated customer subprocess

图 8 为交易流程的编排过程结构树,由交互的行为关系构成整个流程的全局视图。通过该结构树可以发现,顾客制定快递要求这一特定变化仅影响商家流程。另外,根据算法 1 中 $|abstr_Y(G)|=2$ 可知,发货后的修改地址在编排模型(树)中对应两个节点,分别为 n_8 与 n_9 。该变化影响涉及中间子流程的传递作用,因此该变化传播为间接变化

传播。根据定义:

$$\begin{aligned} Y &= send(\text{新地址,快递公司,顾客}) \\ &= send(\text{新地址,商家,顾客}) + send(\text{新地址,快递公司,商家}) \end{aligned}$$

可提取中间协同方为商家,进而转变为直接变化传播问题。

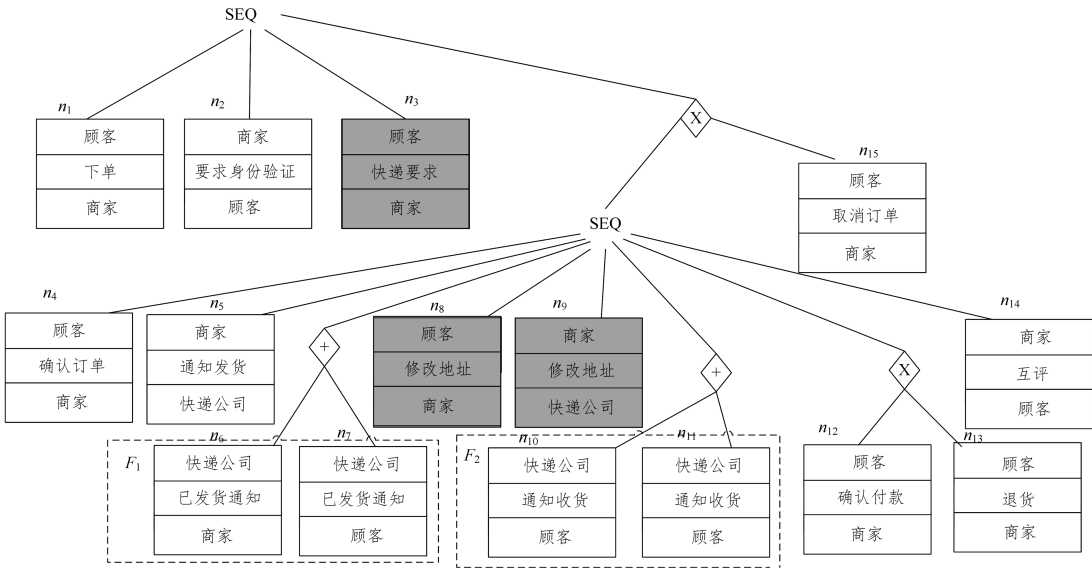


图 8 交易编排模型(树)

Fig. 8 Transaction orchestration model (tree)

由于该案例中涉及的两处变化都为单个的交互变化活动,因此不仅需要更新顾客子流程,还需更新其他受变化影响的交互流程。针对交互流程内的第一个带约束变化(指定快递) $Add_C(X, t_7, t_8, C_X)$,根据算法 1 中的添加模式变化传播步骤确定目标变化域 $R(X^1)$ (如图 9 中的阴影部分 $R(X^1)$),以及对应的目标约束变化为 $Add_C(X^1, t_{21}, t_{24})$,其中 $X^1=Receive$ (指定快递要求,顾客),由于与 t_{23} (默认快递)构成选择结构,因此目标配置参数应设为 $block$ 。

对于分解为两次直接变化传播的第二个带约束变化(修改地址),由流程到编排模型的映射关系有 $pred_G=\zeta(t_{12})=\{n_6, n_7\}$ 及 $succ_G=\zeta(t_{13})=\{n_{10}, n_{11}\}$,将前后集的交互节点按协同方分解为商家 $pred_{2G}=T_predset_{p_2}(pred_G)=\{n_6\}$, $succ_{2G}=T_predset_{p_2}(pred_G)=\{n_{10}\}$ 与快递公司 $pred_{3G}=$

$\{n_7\}$, $succ_{3G}=\{n_{11}\}$,再利用编排模型到交互流程的映射关系得到商家与快递公司的变化域的前后集为 $pred_2=\zeta(pred_{2G})=\{t_{25}\}$, $succ_2=\zeta(succ_{2G})=\{t_{26}\}$ 及 $pred_3=\zeta(pred_{3G})=\{t_{33}\}$, $succ_3=\zeta(succ_{3G})$ 。因此,根据算法 1 中的添加模式变化传播步骤确定的最终目标变化域 $R(Y_1)$ 与 $R(Y^1)$ 分别为商家子流程的变化域与快递公司子流程的变化域,目标约束变化为 $Add_C(Y_1, t_{26}, t_{27})$ 以及 $Add_C(Y^1)=(Y^1, t_{33}, t_{39})$ 。其中, $Y_1=Receive$ (新地址,顾客,商家) $=Send$ (新地址,快递公司,商家), $Y^1=Receive$ (新地址,商家,快递公司)。由于活动 Y_1 在卖家子流程变化域内与其他活动构成顺序结构,而在快递公司子流程变化域内构成并发结构,因此配置参数均为 $hide$ 。更新所有的子流程,得到最终的交易流程如图 9 所示。

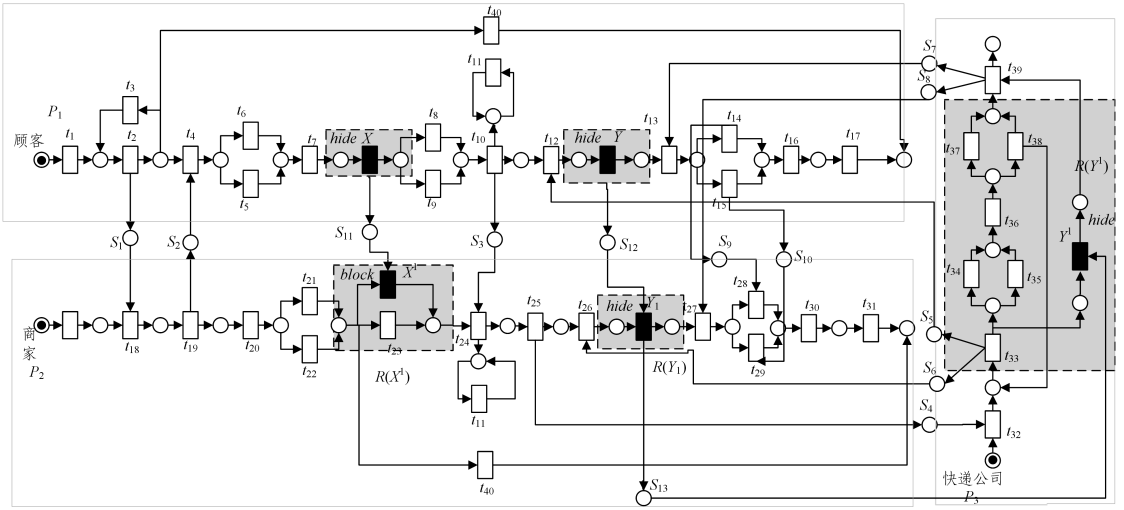


图 9 更新的交互流程

Fig. 9 Updated interaction process

该交易流程内更新变化后的信息传递函数满足：

$$\begin{aligned} \tau(\text{Send}(\text{指定快递}, \text{商家}, \text{顾客})) &= \text{Send}(\text{指定快递}, \text{商家}, \text{顾客}) \\ &= \text{Receive}(\text{指定快递}, \text{顾客}, \text{商家}) \\ \tau(\text{Send}(\text{新地址}, \text{快递公司}, \text{顾客})) &= \text{Send}(\text{新地址}, \text{快递公司}, \text{顾客}) \\ &= \text{Receive}(\text{新地址}, \text{顾客}, \text{快递公司}) \end{aligned}$$

通过共轭函数保证了变化后该交易流程的可兼容性，即更新后的交互流程是合理的。

结束语 本文主要针对交互流程内出现的带约束条件的变化问题，提出了基于配置技术的变化传播分析方法。利用编排模型(树)以及条件抽象与投影前后集的关系确定交互流程的目标变化域，并使用配置技术实现域内带约束的变化行为关系。与一般变化模式的删除与添加属性相比，该配置约束变化既可以保留原有的行为关系，又可以按条件选择新的行为关系，能更有效地适应变化的商业环境。未来将对本文算法做出新的改进，并对利用配置技术处理其他带约束的基本变化模式进行更深层次的研究。

参 考 文 献

[1] GROSSMANN G,MAFAZI S. Change Propagation and Conflict Resolution for the Co-Evolution of Business Processes [J]. International Journal of Cooperative Information Systems, 2015, 12(8):102-140.

[2] WEIDLICH M,WESKE M,MENDLING J. Change Propagation in Process Models using Behavioral Profiles [J]. IEEE International Conference on Services Computing, 2009, 10(8):33-40.

[3] WEIDLICH M,MENDLING J,WESKE M. Propagating changes between aligned process models[J]. Journal of Systems & Software, 2012, 85(8):1885-1898.

[4] GUPTA C,SINGH Y,SINGH D. A Dynamic Approach to Estimate Change Impact using Type of Change Propagation [J]. Journal of Information Processing Systems, 2010, 4(6):597-608.

[5] GROSSMANN G,MAFAZI S. Change Propagation and Conflict Resolution for the Co-Evolution of Business Processes [J]. In-

ternational Journal of Cooperative Information Systems, 2015, 12(8):102-140.

[6] VAN D A W M P,DUMAS M,GOTTSCHALK F, et al. Preserving correctness during business process model configuration [J]. Formal Aspects of Computing, 2010, 22(3/4):459-482.

[7] BONFIGLIO S,ALBANI L. Managing Configurable Business Process as a Service to Satisfy Client Transactional Requirements[C]// IEEE International Conference on Services Computing. IEEE, 2015:154-161.

[8] ASSY N,CHAN N N,GAALOUL W. An Automated Approach for Assisting the Design of Configurable Process Models[J]. IEEE Transactions on Services Computing, 2015, 8(6):874-888.

[9] DÖHRING M,REIJERS H,SMIRNOV S. Configuration vs. Adaptation for Business Process Variant Maintenance: An Empirical Study [J]. Information Systems, 2014, 39(6):108-133.

[10] DERGUECH W,BHIRI S,CURRY E. Designing Business Capability-Aware Configurable Process Models [J]. Information Systems, 2017, 72(6):77-94.

[11] FDHILA W,INDIONO C,RINDERLE-MA S, et al. Dealing with change in process choreographies: Design and implementation of propagation algorithms[J]. Information Systems, 2015, 49(4):1-24.

[12] VAN D A W M P,DUMAS M,GOTTSCHALK F, et al. Preserving correctness during business process model configuration [J]. Formal Aspects of Computing, 2010, 22(3/4):459-482.

[13] SALAY R,GORZNY J,CHECHIK M. Change Propagation due to Uncertainty Change [C]// International Conference on Fundamental Approaches To Software Engineering. Springer-Verlag, 2013:21-36.

[14] DEMUTH A,RIEL M. Co-evolution of Metamodels and Models through Consistent Change Propagation[J]. The Journal of Systems and Software, 2016, 11(5):281-297.

[15] GERBEC M. Safety Change Management-A new Method for Integrated Management of Organizational and technical changes [J]. Safety Science, 2016, 7(6):22-31.