

# 基于业务能力的可配置业务流程模型变化域分析

应 丽 方贤文 王丽丽 刘祥伟

(安徽理工大学数学与大数据学院 安徽 淮南 232001)

**摘 要** 为了满足用户的多样化需求,需要根据用户的实际需求对流程模型进行相应的配置。但是,在流程模型配置的过程中,易出现配置不合理或变化域等问题。为此,提出了一种基于业务能力的可配置业务流程模型变化域分析的方法。首先根据业务能力的类别和属性,将两个可配置源模型进行配置和合并;然后分析得到的业务能力注释可配置模型之间连接器类型的差异,发现可配置流程模型的变化域;其次通过流程模型间的映射关系发现可配置源模型的精确变化域;最后通过相关的实例分析,验证了该方法的有效性。

**关键词** 业务能力,可配置流程模型,连接器,变化域

**中图法分类号** TP391.9 **文献标识码** A **DOI** 10.11896/jsjx.180901692

## Change Region Analysis of Configurable Business Process Model Based on Business Capability

YING Li FANG Xian-wen WANG Li-li LIU Xiang-wei

(College of Mathematics and Big Data, Anhui University of Science and Technology, Huainan, Anhui 232001, China)

**Abstract** In order to meet the diverse needs of users, the process model needs to be configured according to the actual needs of users. However, in the process of process model configuration, problems such as unreasonable configuration or changing region are prone to occur. To solve these problems, a method for changing region analysis of configurable business process models based on business capabilities was proposed. Firstly, according to the categories and attributes of the business capabilities, two configurable source models are configured and merged. Then, the difference of connector types between the available business capability annotation models is analyzed to find the change region of the configurable process model. The exact change area of the configurable source model is found through the mapping relationship between the process models. Finally, the relevant example analysis is given to prove the effectiveness of the proposed method.

**Keywords** Business capability, Configurable process model, Connector, Change region

## 1 引言

随着电子商务的不断发展,用户的需求越来越多样化,业务流程模型可以通过配置来满足用户的多样化需求,但是在配置的过程中,往往存在处理方法不当而间接导致可配置的业务流程出现非一致性区域的问题。因此分析可配置的业务流程的变化域成为了业务流程管理的重难点之一。

目前,关于分析业务流程模型和可配置流程模型变化域的方法有许多,相关的研究人员在该方面也做了许多的工作。文献[1]介绍了一种支持相关流程模型之间的变更传播的方法,鉴于一个流程模型的变化,其利用相应活动的行为概况的行为抽象来确定另一个模型的变化区域。文献[2]提出了基于 Petri 网模块的动态分析方法,该方法首先通过模块化组合

来分析 BPM 的行为并寻找变化区域,然后使用 T-不变量来优化变化区域,以获得准确的变化区域。文献[3]提出了基于 Petri 网行为包含的业务流程变化域融合分析方法,该方法分别研究了业务流程控制流 Petri 网模型的变化域和数据流行为包含关系,确定了目标模型融合网的最终变化域。文献[4]提出了一种新的算法,该算法可以将流程模型合并到另一个可配置的流程模型。文献[5]提出了一个构建流程模型变更管理解决方案的框架,该解决方案提供了检测差异的通用技术,以及变更之间的依赖关系。文献[6]提出了 18 种变更模式和 7 种变更功能。文献[7]提出了一种在没有更改日志的情况下,基于计算差异来解决流程模型差异的变更操作的方法。文献[8]提出了一种基于设计的方法,该方法可以有效地检查一致性标准,并将过程中的变化从过程视图

到稿日期:2018-09-11 返修日期:2019-01-22 本文受国家自然科学基金项目(61572035,61402011),安徽省自然科学基金项目(1508085MF111,1608085QF149),安徽省高校自然科学基金重点项目(KJ2016A208)资助。

应 丽(1993-),女,硕士生,主要研究方向为 Petri 网,E-mail:2484627121@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,博士生,副教授,主要研究方向为业务流程分析和软件认证;刘祥伟(1977-),女,硕士,教授,主要研究方向为业务流程管理。

传播到其参考过程。

已有的研究主要通过模型中变化域的活动节点来搜索模型的变化范围,或者从整个模型的角度去分析业务流程模型的变化域,很少涉及分析可配置的流程模型变化域。针对这一问题,本文提出了一种基于业务能力的可配置业务流程模型变化域分析的方法。首先根据业务能力的类别和属性,将两个可配置源模型进行配置和合并;然后分析得到的业务能力注释可配置模型之间连接器类型的差异;最后通过流程模型间的映射关系发现可配置源模型的精确变化区域,这样有助于提高流程模型配置的正确性。

## 2 动机例子

本节利用某电子商城中买卖双方交易流程实例中可能发生的一种情况来说明本文的研究动机。图 1 和图 2 分别表示买家和卖家进行网上交易的 Petri 网流程。如图 1 所示,买家在浏览商品后提交订单时主要可以进行 3 个操作,即取消订单、修改订单和确认订单。在修改订单或确认订单后,买家可以要求送货的快递类型,接着买家支付货款,最后买家在收货时可以确认收货或者退货。如图 2 所示,卖家在接收订单并确认订单时,会明确发货快递的类型,同时需要核实订单的具体状态,主要是查看订单是否被取消,然后在买家付款后进行发货,最后还要核实订单是被买家确认收货还是退货。表 1 给出了图 1 中每个字母代表的活动名字。表 2 给出了图 2 中每个字母代表的活动名称。

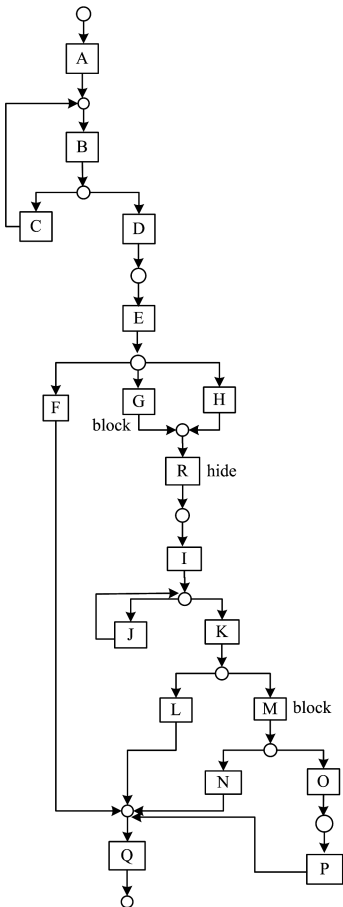


图 1 买家的 Petri 网流程图  $M_1'$

Fig. 1 Petri net flowchart  $M_1'$  of buyer

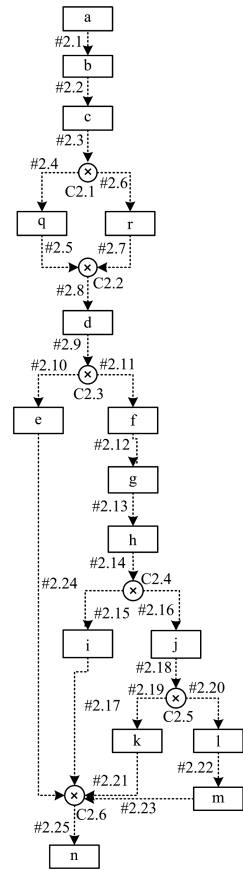


图 2 卖家的 Petri 网流程图  $M_2'$

Fig. 2 Petri net flowchart  $M_2'$  of seller

表 1 图 1 中各个字母代表的活动名称

Table 1 Name of activity represented by each letter in Fig. 1

活动	活动名称	活动	活动名称
A	开始	I	支付货款
B	浏览商品	J	快递状态查询
C	加入购物车	K	收到货物
D	直接购买	L	确认收货并评价
E	提交订单	M	申请退货
F	订单取消	N	申请未通过
G	修改订单	O	申请通过
H	确认订单	P	货物寄回并退款
R	指定快递	Q	交易结束

表 2 图 2 中各个字母代表的活动名称

Table 2 Name of activity represented by each letter in Fig. 2

活动	活动名称	活动	活动名称
a	开始	g	收到支付货款并发货
b	接收订单	h	订单情况跟踪
c	确认订单	i	确认收货并评价
q	指定快递	j	申请退货
r	默认快递	k	申请未通过
d	查询订单状态	l	申请通过
e	订单取消	m	货物收回并退款
f	订单未取消	n	交易结束

上述是买卖双方的交易流程,为了满足买卖双方的一些特殊需求,对上述业务流程模型进行了配置,但是在配置的过程中往往存在一些因素易导致对流程模型的配置不合理,从而使得到的可配置模型出现非一致性区域,即变化域。例如,某些卖家只有自己特定的送货渠道,无法满足买家指定的送货方式,这种情况说明买家无法享有顾客该有的合法权益。

如今,这一类型的问题普遍出现在电子商务流程中,但往往容易被忽视。为了发现流程模型中配置的不合理性,本文旨在基于业务能力发现可配置流程模型的精确变化域。

### 3 基本概念

**定义 1**<sup>[11]</sup>(Petri 网) 称满足下列条件的三元组  $N=(P, T, F)$  为一个 Petri 网:

$$P \cup T \neq \emptyset \quad (1)$$

$$P \cap T = \emptyset \quad (2)$$

$$F \subseteq (P \times T) \cup (T \times P) \quad (3)$$

$$\text{dom}(F) = \{x \in P \cup T \mid \exists y \in P \cup T: (x, y) \in F\} \quad (4)$$

$$\text{cod}(F) = \{x \in P \cup T \mid \exists y \in P \cup T: (y, x) \in F\} \quad (5)$$

式(2)指出,  $P$  和  $T$  是两个不相交的集合,它们是 Petri 网  $N=(P, T, F)$  的基本元素集,其中  $P$  的元素被称为库所; $T$  的元素被称为变迁; $F$  是 Petri 网  $N$  的流关系。用图形来表示一个 Petri 网时,一个小圆圈表示一个库所,一个小矩形表示一个变迁。对于  $x, y \in S \cup T$ ,若  $(x, y) \in F$ ,则从  $x$  到  $y$  画一条有向边。式(3)指出,有向边只存在于小圆圈和小矩形之间,任何两个小圆圈之间或任何两个小矩形之间没有有向边连接。式(4)指出,一个 Petri 网中不应有孤立结点。

**定义 2**<sup>[9-10,13]</sup>(流程模型) 若一个流程模型的 Petri 网是一个四元组  $N=(P, T, F, C)$ ,则需要满足以下几个条件:

(1)  $P$  为有限非空库所的集合,  $T$  为有限非空变迁的集合,  $P \cap T = \emptyset$ 。

(2)  $F \subseteq (P \times T) \cup (T \times P)$  为  $x$  中的流关系。

(3)  $\cdot x = \{y \mid (y \in P \cap T) \cap (y, x) \in F\}$  为  $x$  的前集,  $x \cdot = \{y \mid (y \in P \cap T) \cap (x, y) \in F\}$  为  $x$  的后集。

(4) 若  $x$  为  $P$  的初始库所,则  $\cdot x = \emptyset$ ;若  $x$  为  $P$  的终止库所,则  $x \cdot = \emptyset$ ;  $\text{dom}(F) \cup \text{cod}(F) = P \cup T$  且  $\text{dom}(F) = \text{cod}(F) = \{x \in P \cup T, \exists y \in P \cup T, (y, x) \in F\}$ 。

(5)  $C = \{od, or, pl, cy\}$  为流程模型的结构类型,即顺序、选择、并行和循环 4 种结构。

**定义 3**<sup>[12]</sup>(配置变迁类型)

(1) *allow*: 允许变迁,表示该活动可见并且可以被执行。

(2) *hide*: 隐变迁,表示该活动可以被跳过而直接执行下一个活动,即该活动不可见。

(3) *block*: 阻塞变迁,表示该活动的分支被禁止选择,也就是删除该活动的选择分支。

## 4 基于业务能力的可配置业务流程模型变化域分析

### 4.1 基于业务能力实现两个可配置流程模型的合并

业务能力是一个重要概念,研究者从不同的角度对其进行了定义。从控制流的角度来看:业务能力是组织通过明确捕获流程任务及其时间和逻辑顺序来实现目标的方式;从功能度来看:业务能力,通过服务、计算机程序等执行操作为客户创造价值。本文从功能的角度考虑业务能力,使用这个概念对于描述企业服务、业务流程和任务所实现的内容非常必要。

下面给出业务能力的形式化定义以及一个基于 Petri 网的业务能力将两个可配置源流程模型进行配置和合并的算法。

**定义 4**(业务能力)<sup>[4,14-15]</sup> 一个业务能力是一对能力  $Cap=(Category, Properties)$ ,其中:

(1) *Category*: 类别,是通过属性实现的所有业务能力描述中存在的特定属性。

(2) *Properties*: 属性,表示一组对(属性、值),它们对应于业务功能的一组特性。

**定义 5**(能力注释流程图)<sup>[4]</sup> 能力注释流程图是一个有向图,且由一个六元组  $G=(N, C, A, T, Cap, Cond)$  表示。

(1)  $N$  是一组工作节点,包括初始节点、最终节点、中间节点,且它们都是事件和活动节点。

(2)  $C$  是一组图形连接器,即 *AND-split*, *AND-join*, *OR-split*, *OR-join*, *XOR-split* 和 *XOR-join*。

(3)  $A$  是一组连接所有图节点的有向弧。

(4)  $T$  是一种类型函数,将每个节点与各自的类型相关联。

(5)  $Cap$  是一个与每个活动节点  $n$  关联的注释函数,且  $Cap(n) = (ActionCategory(n), Properties)$ 。

(6)  $Cond$  是一个与条件  $c$  中的每个事件节点相关联的注释函数,  $Cond(n) = c$ 。

**定义 6**(能力注释的可配置流程图)<sup>[4]</sup> 能力注释的可配置流程图是一个有向图,且可以由一个八元组  $G=(N, C, A, T, Cap, CN, CC, Tag)$  表示,其中:

(1)  $N$  是一组工作节点,包括初始节点、最终节点、中间节点,这些节点都是事件和活动节点。

(2)  $C$  是一组图形连接器:即 *AND-split*, *AND-join*, *OR-split*, *OR-join*, *XOR-split* 和 *XOR-join*。

(3)  $A$  是一组连接所有图节点的有向弧。

(4)  $T$  是一种类型函数,将每个节点与各自的类型相关联。

(5)  $Cap$  是一个与每个活动节点  $n$  相关联的注释函数,且  $Cap(n) = (ActionCategory(n), Properties)$ 。

(6)  $CN \subseteq N$  是一组可配置的节点。

(7)  $CC \subseteq C$  是一组可配置的连接器的。

(8)  $Tag$  是一个标记函数,即将  $N, C$  和  $A$  中的每个项连接起来的模型的标识符。

算法 1 通过将两个可配置源模型进行配置和合并,得到最终的业务能力注释的可配置流程模型,具体操作如算法 1 所示。

**算法 1** 将两个可配置源流程模型进行配置和合并

输入: 流程图  $M_1'$ , 流程图  $M_2'$

输出: 最终的业务能力注释的可配置流程图  $CC$

步骤 1 首先根据定义 5,将流程图  $M_1'$  转化成流程图  $M_1=(N_{M_1}, C_{M_1}, A_{M_1}, Cap_{M_1})$ ,将流程图  $M_2'$  转化成流程图  $M_2=(N_{M_2}, C_{M_2}, A_{M_2}, Cap_{M_2})$ 。

步骤 2 根据 Similarity 的具体定义,首先将流程图  $M_1$  的每个事件和功能项目与流程图  $M_2$  的相应项目进行匹配和合并,然后将其余模型的项目(即连接器和弧)整合到结果模型中,得到合并后的流程图  $G$ 。

步骤 3 为  $N_G$  中的每个节点  $n$  赋值一个标签  $Tag$ (即标记函数)。

步骤 4 如果  $| \cdot n | > 1$ ,则需要创建一个新的可配置的连接器的  $CXOR$ ,此时  $CXOR$  的类型是 *XOR-join*;接着,对于每一个  $a \in A_G$ ,如果  $a. Destination = n$ ,则  $a. Destination \leftarrow n$ ,  $Tags.add$

(a. Tag);最后得到  $CXOR.Tag \leftarrow Tags, C_G.add(CXOR), CC_G.add(CXOR), CreateNewArc(Arc), Arc.Source \leftarrow CXOR, Arc.Destination \leftarrow n, Arc.Tag \leftarrow Tags, A_G.add(Arc)$ 。

步骤 5 同理,若  $|n'| > 1$ ,执行上述相似的操作,得到复杂的能力注释可配置流程图  $CC_G$ 。

步骤 6 接下来分析  $CC_G$  中的所有弧,若  $(Src1, PID1, Dest1) \in E_{CC_G} / \{Src1, Des1\} \subset RN_G$  中,且  $\lambda(Src1) = \lambda(Dest1)$ ,则  $ToBeReduced \leftarrow (Src1, Dest1)$ 。

步骤 7 在步骤 6 的基础上,如果  $(Src2, PID2, Dest2) \in E_{CC_G} / Src2$  或  $Dest2 \in ToBeReduced$ ,并且  $\lambda(Src2) = \lambda(Dest2) = \lambda(Src1)$ ,则  $ToBeReduced \leftarrow (Src2, Dest2)$ 。

步骤 8 创建新的连接器(Configurable-Connector),  $Operator = "XOR"$ ,此时  $Connector \in ToBeReduced$ ,如果  $\tau(Connector) = (PID, "AND")$  或  $\tau(Connector) = (PID, "OR")$ ,则  $Operator = "OR"$ ,此时  $\tau(Configurable-Connector) \leftarrow \tau(Connector)$ , $\tau(Configurable-Connector) \leftarrow \tau(id(G), Operator), \tau(Configurable-Connector) \leftarrow ture$ 。

步骤 9 对于每一个  $(Src, PID, Dest) \in E_{CC_G} / Src$  或  $Dest \in ToBeReduced$ ,且  $E_{CC_G} \leftarrow E_{CC_G} \setminus (Src, PID, Dest)$ ,如果  $Dest \in ToBeReduced$ ,并且  $Src \in ToBeReduced$ ,那么  $E_{CC_G} \leftarrow (Src, PID, Confiurable-Connector)$ 。

步骤 10 将上述所得的相同源和目标的弧进行合并,得到最终的业务能力注释的可配置流程图  $CC$ 。

#### 4.2 寻找出可配置源流程模型的精确变化域

算法 1 主要是根据业务能力的类别和属性将两个可配置源流程模型进行配置和合并。在此基础之上,本节提出了算法 2 来寻找可配置源流程模型的精确变化域。下面给出对业务能力的配置约束的定义以及算法 2 的具体操作。

定义 7<sup>[4]</sup>(连接器(业务能力的配置约束))

- (1) *AND-split*: 活动继续进入多个并行分支。
- (2) *AND-join*: 所有活动分支。
- (3) *OR-split*: 在该活动之后可以激活多个活动分支。
- (4) *OR-join*: 多个活动分支在此活动之前被激活。
- (5) *XOR-split*: 必须选择一个可能的活动分支。
- (6) *XOR-join*: 恰好只有一个活动分支。

#### 算法 2 寻找可配置源流程模型的精确变化域

输入: 流程模型  $M_1$  和  $M_2$ , 以及可配置的流程模型  $CC$

输出:  $M_1'$  的变化域  $CM_1'$ ,  $M_2'$  的变化域  $CM_2'$

步骤 1 基于业务能力类别和属性的约束,对流程模型  $M_1$  和  $M_2$ , 以及可配置的流程模型  $CC$  分别进行实例化配置,得到  $M_1$  中的连接器  $C1.n$ ,  $M_2$  中的连接器  $C2.n$ ,  $CC$  中的连接器  $C'1.n, C'2.n$ 。

步骤 2 根据定义 7 得到  $M_1$  中的连接器  $C1.n$  的类型(记为  $CR_1 = \{CR_1^1, CR_1^2, \dots, CR_1^n\}$ ), 以及  $M_2$  中的连接器  $C2.n$  的类型(记为  $CR_2 = \{CR_2^1, CR_2^2, \dots, CR_2^n\}$ )。

步骤 3 同理,计算出  $CC$  中的连接器  $C'1.n, C'2.n$  的类型,记为  $CR_{CC} = \{CR_1^{1'}, CR_1^{2'}, \dots, CR_1^{n'}, CR_2^{1'}, CR_2^{2'}, \dots, CR_2^{n'}\}$ 。

步骤 4 如果  $CR_1^1 \in CR_1, CR_1^{1'} \in CR_{CC}, CR_1 = \{CR_1^1, CR_1^2, \dots, CR_1^n\}, CR_{CC} = \{CR_1^{1'}, CR_1^{2'}, \dots, CR_1^{n'}, CR_2^{1'}, CR_2^{2'}, \dots, CR_2^{n'}\}$ , 存在  $CR_1^1 \neq CR_1^{1'}$  使得  $C1.n \neq C'1.n$ , 则  $CC$  中的连接器  $CR_1^{1'}$  之后的变迁  $t \in CM_1 \wedge t \in T$ , 因此把变迁  $t$ 、弧  $(t, t)$ 、弧  $(t, t')$ 、

库所  $t$ 、库所  $t'$  添加到  $M_1$  的变化域  $CM_1$  中。

步骤 5 如果  $CR_2^1 \in CR_2, CR_2^{1'} \in CR_{CC}, CR_2 = \{CR_2^1, CR_2^2, \dots, CR_2^n\}, CR_{CC} = \{CR_1^{1'}, CR_1^{2'}, \dots, CR_1^{n'}, CR_2^{1'}, CR_2^{2'}, \dots, CR_2^{n'}\}$ , 存在  $CR_2^1 \neq CR_2^{1'}$  使得  $C2.n \neq C'2.n$ , 则  $CC$  中连接器  $CR_2^{1'}$  之后的变迁  $t \in CM_2 \wedge t \in T$ , 因此把变迁  $t$ 、弧  $(t, t)$ 、弧  $(t, t')$ 、库所  $t$ 、库所  $t'$  添加到  $M_2$  的变化域  $CM_2$  中。

步骤 6 输出  $M_1$  的变化域  $CM_1$  和  $M_2$  的变化域  $CM_2$ 。

步骤 7 在模型间的映射关系的基础上,输出  $M_1'$  的变化域  $CM_1'$  和  $M_2'$  的变化域  $CM_2'$ 。

## 5 实例分析

本文通过某电子商城购物的实例来阐述本文方法的有效性。表 3 给出了  $M_1$  的节点类型和业务能力,表 4 给出了  $M_2$  的节点类型和业务能力。

首先利用算法 1,并根据定义 5 将图 1 所示的买家的可配置 Petri 网流程图转化成图 3 所示的  $M_1, M_1 = \langle N_{M_1}, C_{M_1}, A_{M_1}, Cap_{M_1} \rangle$ , 其在表 3 中表示了  $N_{M_1}, C_{M_1}$  和  $Cap_{M_1}, A_{M_1} = \{ \#1.n/n = [1, 31] \}$ 。

表 3  $M_1$  的节点类型和业务能力

Table 3 Nodes types and business of  $M_1$  capabilities

节点: $n$	类型: $T(n)$	业务能力: $Cap(n)$
开始	开始节点	不适用
浏览商品	函数	浏览商品 $Cap_{M_1}$ a cmm: 能力; cmm: 实现; bt: 登录购物账号; xsd: 字符串 bt: 商品名称; dbo: 平台 bt: 价格; xsd: 价格
加入购物车	事件	不适用
直接购买	事件	不适用
提交订单	函数	提交订单 $Cap_{M_1}$ a cmm: 能力; cmm: 实现; bt: 检查订单情况; bt: 修改; bt 确认; bt: 取消
确认订单	事件	不适用
修改订单	事件	不适用
订单取消	事件	不适用
指定快递	事件	不适用
支付货款	函数	支付货款 $Cap_{M_1}$ a cmm: 能力; cmm: 实现; bt: 检查货物; bt 查询货物状态
快递状态查询	事件	不适用
收到货物	函数	收到货物 $Cap_{M_1}$ a cmm: 能力; cmm: 实现; bt: 检查货物; bt: 确认收货; bt: 评价
确认收货并评价	事件	不适用
申请退货	函数	申请退化 $Cap_{M_1}$ a cmm: 能力; cmm: 实现; bt: 通过; bt: 寄回货物 bt: 收到退款; bt: 未通过
申请通过	事件	不适用
货物寄回并退款	事件	不适用
申请未通过	事件	不适用
交易结束	终止节点	不适用



由表 5 可知匹配的项:A(开始)→A(开始),H(确认订单)→c(确认订单),R(指定快递)→q(指定快递),F(订单取消)→e(订单取消),L(确认收货并评价)→i(确认收货并评价),O(申请通过)→l(申请通过),P(货物寄回并退款)→m(货物收回并退款),N(申请未通过)→k(申请未通过)。

2)合并  $M_1$  和  $M_2$  中的函数项,两个功能项都具有相同的功能,生成的合并功能保持不变:

申请退货+退货申请=申请退货

两种功能共享相同的属性但具有不同的值,得到的合并功能是具有可配置值的属性的可配置功能:

提交订单+接收订单=生成订单

3)合并流程图  $M_1$  和  $M_2$ ,得到复杂的能力注释的可配置流程图  $CC_G$ ,如图 5 所示。

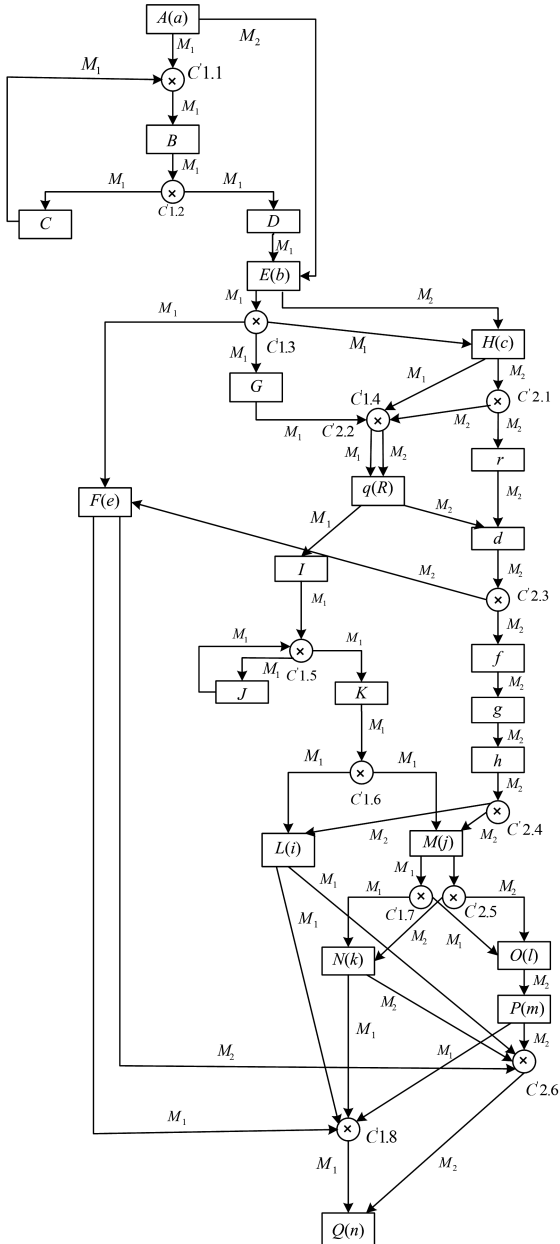


图 5 复杂的能力注释的可配置流程图  $CC_G$

Fig. 5 Configurable process graph  $CC_G$  of complex capability annotated

业务能力注释的可配置流程图  $CC$ ,如图 6 所示。

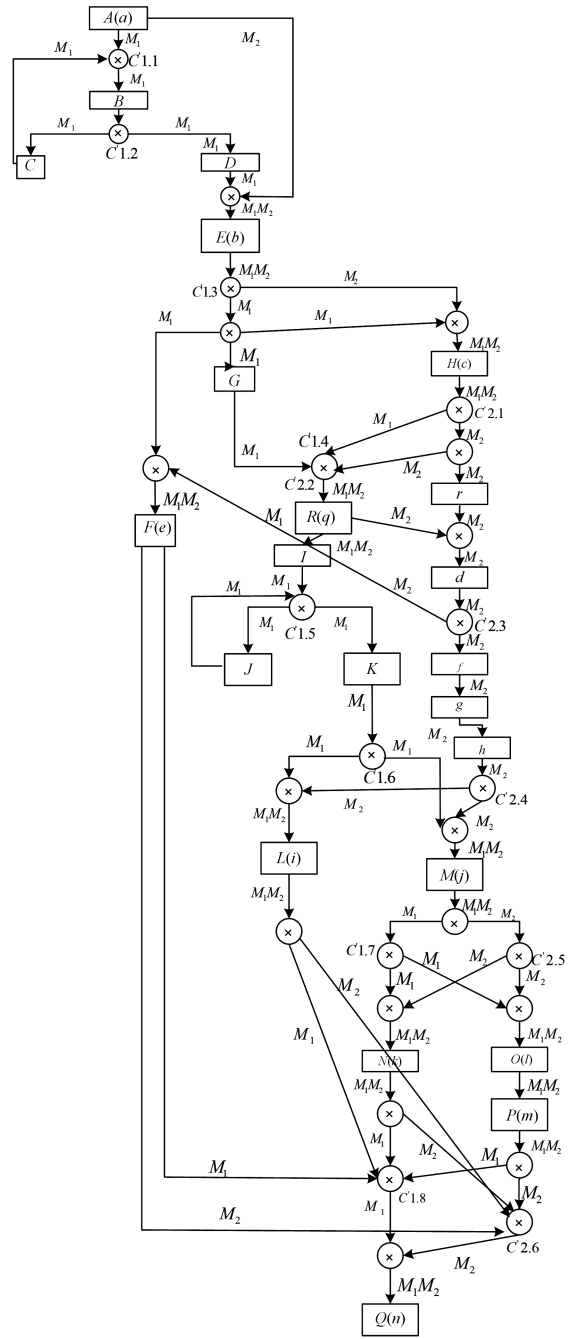


图 6 最终的业务能力注释的可配置的流程图  $CC$

Fig. 6 Configurable process graph  $CC$  of final business capability annotation

根据业务能力的配置约束可以得到  $M_1$  和  $CC$  的连接器类型以及  $M_2$  和  $CC$  的连接器类型,分别如表 6 和表 7 所列。

表 6  $M_1$  和  $CC$  的连接器类型对比

Table 6 Comparison of  $M_1$  and  $CC$  connector type

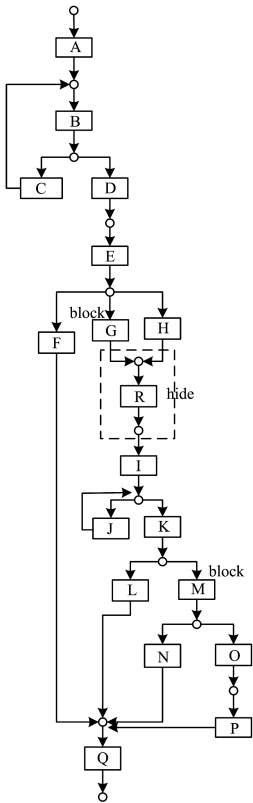
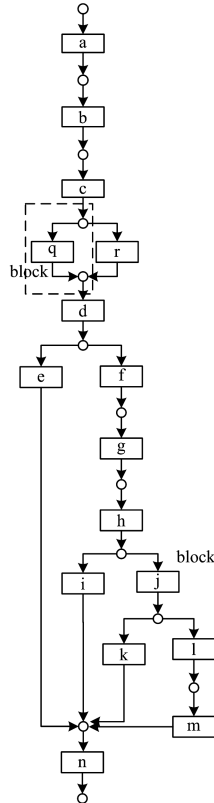
连接器	$M_1$	连接器	$CC$
C1.1	XOR-join	C'1.1	XOR-join
C1.2	XOR-split	C'1.2	XOR-split
C1.3	XOR-split	C'1.3	OR-split
C1.4	XOR-join	C'1.4	OR-split
C1.5	OR-split	C'1.5	OR-split
C1.6	XOR-split	C'1.6	XOR-split
C1.7	XOR-split	C'1.7	XOR-split
C1.8	AND-join	C'1.8	AND-join

4)将复杂的能力注释可配置流程图  $CC_G$  转化为最终的

表7  $M_2$  和 CC 的连接器类型对比Table 7 Connector type comparison of  $M_2$  and CC

连接器	$M_1$	连接器	CC
C2.1	XOR-split	C'2.1	XOR-split
C2.2	XOR-join	C'2.2	XOR-split
C2.3	XOR-split	C'2.3	XOR-split
C2.4	XOR-split	C'2.4	XOR-split
C2.5	AND-join	C'2.5	AND-join
C2.6	AND-join	C'2.6	AND-join

由算法2可知:在表6和表7的基础上,可以相应地在可配置源模型中找出可配置源模型的变化域,如图7和图8中的虚线区域所示。

图7 买家流程  $M_1'$  中的变化域Fig. 7 Change region in buyer process  $M_1'$ 图8 卖家流程  $M_2'$  中的变化域Fig. 8 Change region in buyer process  $M_2'$ 

**结束语** 本文基于业务能力的基础知识对可配置流程模型变化域进行了分析,从而有利于发现可配置流程模型精确的变化域。在配置的过程中往往存在处理方法不当从而间接导致可配置流程模型出现非一致性的区域的问题,因此配置得到的流程模型不是真正合理的模型。本文首先根据业务能力的类别和属性将两个可配置源模型进行配置和合并,然后分析得到的业务能力注释可配置模型之间连接器类型的差异,其次通过模型间的映射关系发现源可配置流程模型的精确变化域。在未来的工作中,还需要发现一种正确且合理的配置方法,使得配置后的流程模型不再出现变化域问题。

## 参考文献

[1] WEIDLICH M, MENDLING J, WESKE M. Propagating

changes between aligned process models [J]. Journal of Systems and Software, 2012, 85(8): 1885-1898.

- [2] FANG X W, LIU L, LIU X W. Analyzing method of change region in BPM based on module of Petri net. Information Technology Journal, 2013, 12(8): 1655-1659.
- [3] FANG X W, ZHAO F, FANG H, et al. The Fusion Analysis Method about the Change Region of the Business Process Model Based on Behavior Inclusion in Petri Net [J]. Chinese Journal of Computers, 2018, 41(3): 695-708.
- [4] DERGUECHA W, BHIRIB S, CURRY E. Designing business capability-aware configurable process models [J]. Information Systems, 2017, 72(12): 77-94.
- [5] GERTH C, KÜSTER J M, ENGELS G. Language-independent change management of process models [C] // 12th International Conference on Model Driven Engineering Languages and Systems. 2009: 152-166.
- [6] WEBER B, REICHERT M, RINDERLE-MA S. Change patterns and change support features-Enhancing flexibility in process-aware information systems [J]. Data & Knowledge Engineering, 2008, 66(3): 438-466.
- [7] KÜSTER M, GERTH C, FRSTER A, et al. Detecting and resolving process model differences in the absence of a change log [C] // Proceedings of the 6th International Conference on Business Process Management. Springer Verlag, 2008: 244-260.
- [8] GROSSMANN G, MAFAZI S, MAYER W, et al. Change Propagation and Conflict Resolution the Co-Evolution of Business Process [J]. International Journal of Cooperative Information Systems, 2015; 24(3): 1540002.
- [9] WEIDLICH M, MENDLING J, WESKE M. Efficient consistency measurement based on behavioral profiles of process models [J]. IEEE Transactions on Software Engineer, 2011, 37(3): 410-429.
- [10] SMIRNOV S, WEIDLICH M, MENDLING J. Business process model abstraction based on behavioral profiles [C] // Proceedings of the 8th International Conference on Service Oriented Computing. San Francisco, USA, 2010: 1-16.
- [11] WU Z H. Petri nets theory [M]. Beijing: Mechanical Industry Press, 2006: 6-22.
- [12] VAN DER AALST W M P, LOHMANN N, LA R M. Ensuring correctness during process configuration via partner synthesis [J]. Information Systems-IS, 2012, 37(9): 574-592.
- [13] SERGEY S, WEIDLICH M, MENDLING J. Business process model abstraction based on behavioral profiles [C] // Proceedings of the 8th International Conference. San Francisco, USA, 2010: 1-16.
- [14] DERGUECH W, BHIRI S. Modelling, interlinking and discovering capabilities [C] // Proceeding. of ACS International Conference on Computer Systems and Applications (AICCSA). Ifrane, Morocco, 2013: 22-34.
- [15] BHIRI S, DERGUECH W. Capabilities as Attribute Featured Entities [J]. Web Information Systems and Technologies, 2012, 140(6): 70-85.