

# 基于接口变迁的交互流程模型挖掘方法

翟鹏珺 方贤文 刘祥伟

(安徽理工大学信息与计算科学系 安徽 淮南 232001)

**摘 要** 流程模型挖掘是基于系统运行记录下的事件日志来还原特征对应流程模型的技术。目前已有的挖掘方法多是基于由系统分解出的不同模块之间交互频繁且模块包含特征较少的场景。在挖掘包含较多特征、交互不频繁的流程模型方面,目前的方法存在一定的局限性。鉴于此,文中提出了基于接口变迁的交互流程模型挖掘方法。首先,利用现有的挖掘方法来挖掘模块内部的特征序,确定初始模块网;其次,遍历事件日志以查找疑似接口变迁;然后,通过挖掘特征网来确定接口变迁,并对接口变迁增加接口库所;最后,基于开放 Petri 网,利用合成网的观点将交互模块合成为一个完善的流程模型 Petri 网。通过实例分析,验证了该挖掘方法的有效性。

**关键词** 流程模型挖掘,日志特征,模块网,接口变迁,特征网

中图分类号 TP391.9 文献标识码 A DOI 10.11896/j.issn.1002-137X.2018.03.052

## Interaction Process Model Mining Method Based on Interface Transitions

ZHAI Peng-jun FANG Xian-wen LIU Xiang-wei

(Department of Information and Computing Science, Anhui University of Science and Technology, Huainan, Anhui 232001, China)

**Abstract** Process model mining is a technology based on the event logs recorded by running system to discover process model corresponding to features. At present, most of the mining methods are based on the frequent interaction between different modules which are decomposed by the system, and there are a few features within modules. There are some limitations of the current process mining methods in the aspect of mining process model which includes multiple features and infrequent interaction. This paper provided an interaction models process mining method based on interface transitions. Firstly, the order of features within modules is discovered using existing methods of mining to find the initial module nets. Secondly, the event log is traversed to search the suspect interface transitions. Then, the interface transition is determined by the mining of the feature net, and the interface place is added to it. Finally, based on the view of open Petri net, the interactive modules are synthesized into a complete process model Petri net. The analysis of instance is used to verify the effectiveness of the mining method.

**Keywords** Process model mining, Log feature, Module net, Interface transition, Feature net

## 1 介绍

目前,流程模型管理在许多领域都占据着不可替代的位置,并发挥着至关重要的作用。它不仅确保企业正常运行,还为企业准确、高效地运行提供了保障,对企业业务的运行具有指导意义。业务流程模型是流程模型管理的关键构件,一个企业可以通过一个有效的流程模型来管理其内部的运行情况,提高业务效率,同时尽可能地降低运行成本,从而获得最大利润。因此,根据系统记录的事件日志信息挖掘合理的流程模型 Petri 网已成为国内外关注的课题。

流程挖掘的基本思想是从信息系统记录的事件日志中提

取潜在知识。在流程挖掘方法的研究方面,目前主要是针对 workflow 模型进行挖掘。相对于人为的业务流程建模,从事件日志的信息中挖掘得到的流程模型减少了主观因素,使得模型更具客观性。流程挖掘技术可以产生高质量和适用性强的模型,但日常工作中能满足客户需求的建模工具很少见。为解决这一问题,文献[1]提出了一种基于流程的挖掘方法,该方法涉及建模工具和应用程序的流程建模与软件可用性方面的连续分析,包含用户监控、轨迹聚类、模型推导和模型分析等阶段。文献[2]提出了基于执行日志的业务流程挖掘方法,以及基于  $L^*$  算法及增量日志的业务流程挖掘方法。文献[3]研究了不完整日志对流程挖掘的影响,通过引入概率行为的

收稿日期:2017-01-24 返修日期:2017-05-30 本文受国家自然科学基金项目(61572035,61402011,61272153),安徽省自然科学基金(1508085MF111),安徽省高校自然科学基金重点项目(KJ2014A067),安徽理工大学研究生创新基金(2017CX2048),安徽省优秀青年基金项目(ZY290)资助。

翟鹏珺(1991—),女,硕士生,主要研究方向为 Petri 网;方贤文(1975—),男,博士,教授,主要研究方向为 Petri 网、可信软件,E-mail:280060673@qq.com(通信作者);刘祥伟(1977—),女,硕士,副教授,主要研究方向为业务流程管理,E-mail:877191453@qq.com。

关系,提出了一种更为强大的流程挖掘算法,并验证了该挖掘算法需要的模型比其他挖掘算法所需模型小。文献[4]提出了一种模型修复的流程挖掘方法,当事件日志可以在模型中重放时,不需要改变模型,否则需要利用模型修复技术对模型的该部分进行修复。文献[5]描述了一种挖掘事件日志中的频繁行为模式的方法,这些模式为局部流程模型,局部流程模型的挖掘可以定位在进程发现和事件/序列模式挖掘之间。文献[6]提出了从事件日志中挖掘过程树的方法,由于树的形式确保了模型的合理性,因此该方法很好地权衡了模型的质量维度,而且在一定程度上是省时的。由于系统中存在大量的活动,在计算方面对模型质量的检测提出了很大的挑战性,为此文献[7-8]提出了不同的方法,通过把流程挖掘问题分割成较小的问题,使得挖掘流程模型的时间大大减少。文献[9]提出基于日志特征挖掘流程模型的新方法,即基于事件日志挖掘功能结构,在功能结构的背景下分步挖掘流程模型。已有的挖掘方法大多是基于模块少、交互频繁的模式,在实际应用中存在一定的局限性,对不同模块交互不频繁且模块内部包含特征较多的场景并不适用。针对这一缺陷,本文提出基于疑似接口变迁挖掘流程模型的方法,以弥补单纯基于事件日志挖掘流程模型过程中出现的计算复杂度高、适用范围小等缺陷。

本文第2节提出一个动机例子,引出基于接口变迁挖掘交互流程模型的方法;第3节介绍关于Petri网的基础知识;第4节重点阐述基于疑似接口变迁挖掘流程模型的方法,并给出相应的算法步骤;第5节通过实例分析来说明方法的有效性;最后总结全文并展望未来工作。

## 2 动机例子

随着网络技术的发展,在网上订购外卖越来越受欢迎。基于外卖系统,本文考虑3个模块:买家中心、支付中心和卖家中心。买家登录买家中心,挑选商品并支付费用;支付中心收到支付信息后,向卖家中心递交订单信息;卖家中心接收到订单信息后,为买家准备订单并联系买家完成购物。在实际情景中,买家在提交订单后可能会取消订单,或者卖家拒绝接收订单,此时支付中心需要向买家退还已支付的费用。

当前系统所涉及的运行软件并不存在精确的模型来说明系统中的行为。相反,从系统中只能得到一些由于系统运行而被记录的事件日志(见表1),部分事件日志并没有完全重述系统的运行。基于这些事件日志,软件构造者可以推得由系统分解出的模块所提供的特征,如下所示:

- 老用户(A)
- 新用户(B)
- 注册(C)
- 填写个人信息(D)
- 登录(E)
- 修改信息(F)
- 挑选商品(G)
- 提交订单(H)
- 删除订单(I)
- 接收外卖(O)
- 结束购物(P)
- 卖家登录(Q)
- 拒绝订单(R)
- 接受订单(S)
- 加工制作(T)
- 配送外卖(U)
- 联系买家(V)
- 用户登录(W)

- 确认订单(J)
- 新用户5折(K)
- 老用户原价(L)
- 支付成功(M)
- 等待外卖(N)
- 扣除费用(X)
- 退还费用(Y)
- 收款反馈(Z)
- 支付失败(a)

表1 外卖系统的事件日志  
Table 1 Event logs of delivery system

案例	事件日志
1	A, E, G, H, I, P
2	A, E, F, G, H, I, P
3	B, C, D, E, F, G, H, I, P
4	A, E, G, H, J, L, M, N, O, P
5	A, E, H, K, M, W, Z, Q, S, V, O, P
6	A, E, G, H, J, L, W, X, Z, M, N, O, P
7	B, E, G, H, J, L, W, X, Z, M, Q, R, Y, a, P
8	B, C, D, E, F, G, H, J, K, W, X, Z, M, N, O, P
9	A, E, F, G, H, J, L, W, X, Z, M, N, Q, R, Y, a, P
10	A, E, F, G, H, J, L, W, X, Z, M, N, Q, S, T, U, V, O, P
11	B, C, D, E, G, F, H, J, K, W, X, Z, M, N, Q, R, Y, a, P
12	B, C, D, E, F, G, H, J, K, W, X, Z, M, N, Q, S, T, U, V, O, P

根据事件日志信息以及特征表达语义,外卖系统的3个模块如图1所示。已有研究<sup>[1-5,9-10]</sup>提出的基于事件日志挖掘流程模型的方法适用于特征数量少且模块交互频繁的情境,对于本文的动机例子并不适用。下面主要介绍一种基于事件日志挖掘流程模型的方法,该方法不仅可以基于特征数量少、模块交互频繁的事件日志挖掘交互流程模型,而且在特征数量多且模块交互并不频繁的情况下也具有一定的优越性。

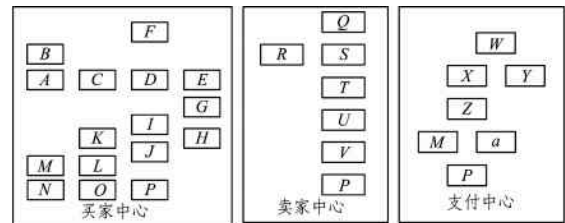


图1 动机例子的源特征模块

Fig. 1 Initial features modules of motivation example

## 3 基础知识

本节介绍与本文密切相关的概念,其他概念及术语请参考文献[11]。

**定义 1<sup>[12]</sup>**(流程模型 Petri 网) 设一个流程模型 Petri 网  $PN=(P, T, F, C)$  是一个四元组:

- 1)  $P$  和  $T$  分别是有限的库所集和变迁集;
- 2)  $P \neq \emptyset, T \neq \emptyset$  且  $P \cap T = \emptyset$ ;
- 3)  $F=(P \times T) \cup (T \times P)$  表示  $PN$  的流关系,  $(P \cup T, F)$  是强连通图;
- 4)  $dom(F) \cup cod(F) = P \cup T$ , 其中  $dom(F) = \{x \in P \cup T \mid \exists y \in P \cup T, (x, y) \in F\}$ ,  $cod(F) = \{x \in P \cup T \mid \exists y \in P \cup T, (y, x) \in F\}$ ;
- 5)  $C = \{\text{and}, \text{xor}, \text{or}\}$  是流程网的结构类型。

对于一个流程模型 Petri 网  $PN=(P, T, F, C)$ , 给定一个节点  $n \in (P \cup T)$ , 则有  $n$  的前集  $\cdot n = \{n' \mid (n', n) \in F\}$ ,  $n$  的后

集  $n' = \{n' | (n, n') \in F\}$ 。另外,一个流程模型 Petri 网  $PN$  和一个初始标识  $M_0$  就确定了一个标识网,变迁  $t$  在标识  $M_0$  下可以发生( $M_0[t >]$ ),如果  $\forall p \in P: p \in \cdot t \rightarrow M(p) \geq 1$ 。称  $M_1$  为从  $M_0$  可达的,若存在  $t \in T$ ,使得  $M_0[t >] M_1$ 。

**定义 2**<sup>[13]</sup>(开放 Petri 网)  $OPN=(P, I, O, T, F, i, f)$  为开放 Petri 网,当且仅当:

- 1)  $(PUIUO, T, F)$  是一个 Petri 网;
- 2)  $P$  是内部库所集,  $T$  是变迁集,  $F$  是库所和变迁之间弧线的集合;
- 3)  $I$  是输入库所集,且  $\cdot I = \emptyset$ ;  $O$  是输出库所集且  $O' = \emptyset$ ;
- 4)  $P, I, O$  两两不相交;
- 5)  $i$  是初始标识,  $f$  是终止标识。

一个  $OPN$  是封闭的,如果  $I=O=\emptyset$ 。  $I \cup O$  为  $OPN$  的接口库所。

**定义 3**(接口库所)  $OPN=(P, I, O, T, F, i, f)$  为开放 Petri 网,不同  $OPN$  之间的信息交流通过一个特殊的库所表示。输入库所  $I$  从外部接收信息,只有一个输出弧;输出库所  $O$  向外界发送信息,只有一个输入弧。

**定义 4**<sup>[14]</sup>(合成网) 设  $N_1$  和  $N_2$  是两个  $OPN$ ,  $N_1 \oplus N_2$  表示  $N_1$  和  $N_2$  的合成网,  $N_1 \oplus N_2$  也是一个  $OPN$ , 当且仅当:

- 1)  $P = P_{N_1} \cup P_{N_2} \cup (I_{N_1} \cup O_{N_1}) \cup (I_{N_2} \cup O_{N_2})$  为库所集;
- 2)  $T = T_{N_1} \cup T_{N_2}$  为变迁集;
- 3)  $F = F_{N_1} \cup F_{N_2}$  为流关系;
- 4)  $I = (I_{N_1} \setminus O_{N_2}) \cup (I_{N_2} \setminus O_{N_1})$  为输入库所集;
- 5)  $O = (O_{N_1} \setminus I_{N_2}) \cup (O_{N_2} \setminus I_{N_1})$  为输出库所集;
- 6)  $i = i_{N_1} \cup i_{N_2}$  是初始标识;
- 7)  $f = f_{N_1} \cup f_{N_2}$  是终止标识。

两个  $OPN$  的合成通过接口库所连接,  $N_1$  的一个输入(输出)库所必然对应  $N_2$  的一个输出(输入)库所。

#### 4 基于接口变迁挖掘流程模型

本节主要介绍基于接口变迁挖掘交互流程模型的新方法。首先,利用现有的挖掘方法挖掘模块内部的特征序,确定初始模块网;其次,通过遍历事件日志查找疑似接口变迁;然后,通过挖掘特征网确定接口变迁,并对接口变迁增加接口库所;最后,在模块网和特征网的基础上,运用合成网的观点将交互模块合成为流程模型 Petri 网。

##### 4.1 基于通讯行为轮廓挖掘模块网

事件日志可以描述系统执行特征的顺序,而不同模块之间的通讯通过特征来传送。规定每个特征只能属于一个模块,且通讯只能在同一个方向上。例如,给定两个通讯特征  $A$  和  $B$ ,则特征  $A$  发送信息给特征  $B$ ,或特征  $B$  发送信息给特征  $A$ ,但是两者不能同时发生。在一个事件日志中,需要考虑事件或者特征发生的顺序。为此,基于通讯后继关系可以定义通讯行为轮廓。

**定义 5**<sup>[9]</sup>(通讯行为轮廓) 设  $L \subseteq T^*$  是一个事件日志,  $\mathcal{R}: \bar{T} \rightarrow M, \bar{T}$  是一个特征集,  $M \in Rng(\mathcal{R})$  是一个模块。通过

$a \ll_L b$  定义  $\ll_L \subseteq \bar{T} \times \bar{T}$  为通讯后继,当且仅当对一些迹  $\sigma, \mathcal{R}(a) \neq \mathcal{R}(b), \sigma(i) = a$  且  $\sigma(i+1) = b$ , 其中  $1 \leq i \leq |\sigma|$ 。则通讯行为轮廓是一个三元组  $(\rightarrow_c, \parallel_c, +_c)_L^{Com}$ :

- 1)  $A \rightarrow_c B$  当且仅当  $A \ll_L B$  且  $B \not\ll_L A$ ;
- 2)  $A \parallel_c B$  当且仅当  $A \ll_L B$  且  $B \ll_L A$ ;
- 3)  $A +_c B$  当且仅当  $A \not\ll_L B$  且  $B \not\ll_L A$ 。

**定义 6**<sup>[9]</sup>(模块日志)  $L \subseteq T^*$  是一个事件日志,  $\mathcal{R}: \bar{T} \rightarrow M, \bar{T}$  是一个特征集,  $M \in Rng(\mathcal{R})$  是一个模块。设  $(\rightarrow_c, \parallel_c, +_c)$  是其相应的通讯行为轮廓,则有模块日志  $L_M = \{\sigma | (F^* \upharpoonright_{\mathcal{R}(F^*)} = M) \mid \sigma \in L\}$ 。

第 2 节的动机例子中包含 3 个模块,基于定义 6 可以确定各个模块所对应的日志的迹(模块日志)。例如,对于卖家中心模块,有模块日志  $L_1 = \{\langle Q, R, P \rangle, \langle Q, S, T, U, V, P \rangle\}$ ; 对于支付中心模块,有模块日志  $L_2 = \{\langle W, X, Z, M, P \rangle, \langle W, X, Z, a, P \rangle\}$ 。通过运用文献[15]中提到的归纳挖掘法,可以得到如图 2 所示的 3 个模块网。

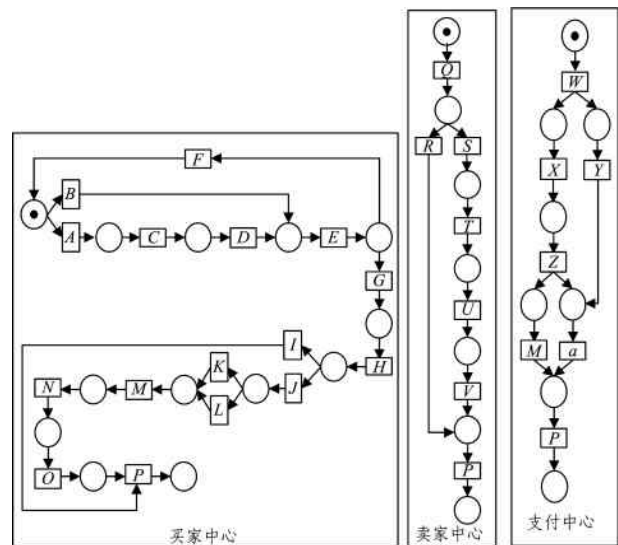


图 2 动机例子模块网

Fig. 2 Module nets for motivation example

##### 4.2 查找模块网中的疑似接口变迁

在模块网中,每个特征由一个对应的变迁表示。根据模块之间的交互方式,为了降低挖掘交互模块流程模型 Petri 网的计算复杂度,下面定义接口变迁。

**定义 7**(接口变迁)  $M_i \in Rng(\mathcal{R})$  为一个模块,其对应的模块网为  $MN_i$ 。  $T_i$  为模块网  $MN_i$  的变迁集,其中  $1 \leq i < |Rng(\mathcal{R})|$ 。  $\tilde{t} \in T_i$  为  $MN_i$  的接口变迁,若  $\tilde{t}$  满足以下条件之一:

- 1)  $\tilde{t} \in T_m, \cdot \tilde{t} \in MN_m$  但  $\tilde{t}' \notin MN_m$ ;
- 2)  $\tilde{t} \in T_m, \cdot \tilde{t} \notin MN_m$  但  $\tilde{t}' \in MN_m$ ;
- 3)  $\tilde{t} \in T_m, \cdot \tilde{t} \in MN_m$  或  $\cdot \tilde{t} \notin MN_m, \tilde{t}' \in MN_m$  或  $\tilde{t}' \notin MN_m (1 \leq m \leq i)$ 。

**算法 1** 查找模块网的疑似接口变迁

输入:事件日志  $L_i \subseteq T_i^*$

输出:疑似接口变迁集  $\tilde{T}'$

1. 遍历事件日志,将其按照包含特征数的多少排列,并找出包含特征

数较多的日志(日志包含的特征数多,说明它是完整迹的可能性较大),转入步骤 2。

2. 根据定义 6,由事件日志  $L_i$  的通讯行为轮廓  $(\rightarrow_c, \parallel_c, +_c)_L^{Com}$  确定相应的模块日志  $L_M = \{\sigma_{|(F^* \mid \exists(F^*)=M)} \mid \sigma \in L\}$ , 据此得到模块网  $MN_i$ , 转入步骤 3。
3. 将步骤 1 查找出的事件日志  $L \in L_j$  放入步骤 2 所得的模块网  $MN_i$  中, 日志  $L$  在  $MN_i$  中可以重放, 则  $L$  为有效日志, 转入步骤 4, 否则舍弃。
4. 对事件日志中的特征进行模块区域划分, 有  $L = M_1 M_2 \dots M_n$ 。根据定义 7, 存在由模块区域  $M_1$  中的最后一个特征、模块区域  $M_n$  中的第一个特征以及模块区域  $M_2, M_3, \dots, M_{n-1}$  中的首尾特征共同组成的疑似接口变迁集  $\tilde{T}$ , 即  $\tilde{T} = \{\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_1 \mid \tilde{t}_1 \in M_1, \tilde{t}_2, \dots, \tilde{t}_{n-1} \in M_2 \cup \dots \cup M_{n-1}, \tilde{t}_n \in M_n\}$ , 转入步骤 5。
5. 结合模块, 通过比较特征语义之间的紧密程度即特征语义之间的逻辑关系, 删除步骤 4 所得疑似接口变迁中极小可能成为接口变迁的变迁, 从而得到最终的疑似接口变迁。

### 4.3 挖掘特征网并确定接口变迁

通过观察动机例子可以得出, 不同的特征之间可以发送和接收一些复杂信息。例如, 特征  $M$  在接收特征  $Z$  的信息后才可以发送信息给特征  $Q$ 。因此, 接下来要重构每个疑似接口变迁所对应特征的内部行为。为了重构特征, 基于每个疑似接口变迁特征产生一个事件日志, 该事件日志包含了特性之间的交流信息, 称为特征日志。

**定义 8**<sup>[9]</sup> (特征日志)  $L \subseteq T^*$  是一个事件日志,  $F^* \subseteq \bar{T}$  是模块中的一部分特征,  $(\rightarrow_c, \parallel_c, +_c)_L^{Com}$  为其相应的通讯行为轮廓, 则有特征日志  $L_{F^*} = \{\sigma_{|C(F^*)} \mid \sigma \in L, F^* \in \sigma\}$ , 其中  $C(F^*) = \{A \mid A \rightarrow_c F^* \vee F^* \rightarrow_c A\}$ 。

基于这些特征日志, 应用文献[15]中的归纳挖掘方法可以挖掘出一个合理的工作流网。将挖掘出的合理工作流网转化成成一个开放 Petri 网  $OPN$ , 用于说明特征发送和信息接收, 由此便对事件日志中的每个疑似接口变迁所对应的特征产生了一个特征网。

**定义 9**<sup>[9]</sup> (特征网)  $L \subseteq T^*$  是一个事件日志,  $F^* \in \bar{T}$  是模块中的一部分特征,  $(\rightarrow_c, \parallel_c, +_c)_L^{Com}$  为其相应的通讯行为轮廓。特征网  $N_F$  是一个开放 Petri 网  $OPN(P, I, O, T, F, i, f)$ , 当:

- 1)  $P = \bar{P}, T = \bar{T}', i = [\bar{i}], f = [\bar{f}]$ ;
- 2)  $I = \{p_{A-F^*} \mid A \rightarrow_c F^*\}$ ;
- 3)  $O = \{p_{F^*-A} \mid F^* \rightarrow_c A\}$ ;
- 4)  $F^* = \{\bar{F} \cup (t, p_{F^*-A}) \mid t \in T, \lambda(t) = A, F^* \rightarrow_c A\} \cup \{(p_{A-F^*}, t) \mid t \in T, \lambda(t) = A, A \rightarrow_c F^*\}$ 。

其中,  $t$  是变迁集  $T$  中的一个变迁;  $\lambda$  是映射函数, 将变迁  $t$  映射到相应的特征  $A$ ;  $\langle \bar{P}, \bar{T}', \bar{F}, \bar{i}, \bar{f} \rangle$  是被挖掘出的工作流网。

由模块日志挖掘出的模块网定义了模块内部所有不同特征之间的执行顺序, 而通过挖掘疑似接口变迁对应特征的特征网确定了不同模块之间的交互方式, 并添加相应的接口库所。最后, 基于挖掘所得的模块网和特征网, 利用合成网的观点可以挖掘出完整的流程模型 Petri 网。

### 算法 2 挖掘流程模型 Petri 网

输入: 模块网  $MN_i$ , 事件日志  $L_j \subseteq T_i^*$ , 疑似接口变迁  $\tilde{T}'$

输出: 流程模型 Petri 网

1. 查找模块网  $MN_i$  中与疑似接口变迁  $\tilde{T}'$  所对应特征  $F^*$  存在通讯的特征集合  $C(F^*)$ , 其中  $C(F^*)$  表示特征  $F^*$  的前驱特征及后继特征的一个特征集合, 转入步骤 2。
2. 检验步骤 1 所得特征集合, 若疑似接口变迁  $\tilde{T}' \in MN_i$ , 且  $C(F^*) \subset MN_i$  (即特征  $F^*$  的前驱特征和后继特征不完全包含于模块网  $MN_i$  中), 则疑似接口变迁  $\tilde{T}'$  为接口变迁, 转入步骤 3; 否则舍弃该变迁, 转入步骤 1。
3. 确定事件日志  $L_j$  在步骤 1 所得特征上的投影, 根据定义 8 查找模块网  $MN_i$  中疑似接口变迁  $\tilde{T}'$  所对应特征的特征日志  $L_{F^*} = \{\sigma_{|C(F^*)} \mid \sigma \in L, F^* \in \sigma\}$ , 转入步骤 4。
4. 基于特征日志  $L_{F^*}$ , 应用已有的归纳挖掘方法产生一个合理的工作流网。根据定义 9 中定义的接口库所  $I$  和  $O$ , 将工作流网转化为一个开放 Petri 网  $OPN(P, I, O, T, F, i, f)$ , 即对接口变迁  $\tilde{T}'$  添加相应的输入库所  $I$  和输出库所  $O$ , 由此模块网  $MN_i$  便成为一个开放 Petri 网, 转入步骤 5。
5. 重复以上步骤, 直至遍历完所有接口变迁  $\tilde{T}'$ , 以及模块网  $MN_i$  都被准确地转换为开放 Petri 网  $N_i$ , 转入步骤 6。
6. 根据定义 4, 利用开放 Petri 网合成网的观点, 通过步骤 5 所得每个开放 Petri 网  $N_i$  所对应的接口库所将  $N_i$  进行两两组合, 从而得到最终挖掘出的流程模型 Petri 网(合成网)。

## 5 实例分析

本节以挖掘外卖系统(动机例子)的流程模型 Petri 网为例来说明所提方法的可行性。4.1 节已根据事件日志挖掘得到外卖系统的模块网, 下面查找外卖系统的疑似接口变迁集  $\tilde{T}'$  及其对应特征的特征网。

首先遍历系统运行所记录的事件日志, 根据其包含的特征数, 考虑表 1 中的事件日志案例 9—案例 12, 分别如图 3 所示。其中虚线中的特征属于同一个模块网  $MN_3$  (支付中心), 实线中的特征均属于另一个模块网  $MN_2$  (卖家中心), 其余的特征则属于其他模块网  $MN_1$  (买家中心)。根据算法 1, 查找疑似接口变迁集  $\tilde{T}' = \{L, W, Z, M, N, Q, R, Y, a, P, V, O, K\}$ , 由于特征  $P$  为终止特征, 且包含于每个模块内, 因此不考虑特征  $P$ 。另外, 支付中心的登录属于独立行为, 同样不考虑特征  $W$ 。最终得到疑似接口变迁集  $\tilde{T}' = \{L, Z, M, N, Q, R, Y, a, V, O, K\}$ 。

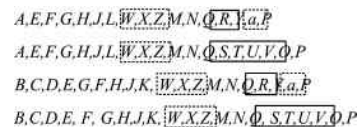


图 3 动机例子事件日志模块划分

Fig. 3 Partition of modules based on event logs for motivation example

分析疑似接口变迁集  $\tilde{T}' = \{L, Z, M, N, Q, R, Y, a, V, O, K\}$ , 其中特征  $L \in MN_1$ , 且  $C(L) = \{J, M\} \subset MN_1$ 。根据算法 2, 特征  $L$  不是接口变迁。同理, 分析得到特征  $K, W$  和  $a$  不是接口变迁, 因此得到接口变迁集  $\tilde{T}'_i = \{Z, M, Q, R, Y, V,$

O)。根据算法 2,利用文献[15]中的归纳挖掘方法得到如图 4 所示的特征网。

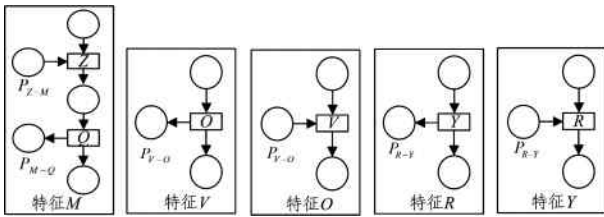


图 4 动机例子疑似接口变迁特征网

Fig. 4 Feature nets of suspect interface transitions for motivation example

最终,基于外卖系统的模块网和特征网,利用开放 Petri 网合成网的观点挖掘得到外卖系统合理的流程模型 Petri 网,如图 5 所示。

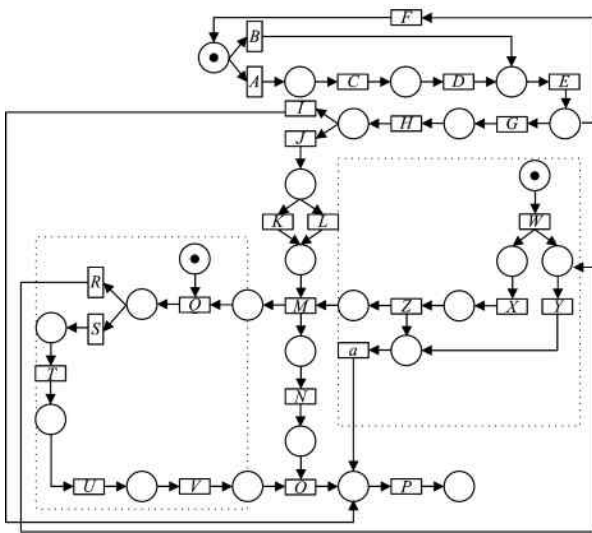


图 5 动机例子流程模型 Petri 网

Fig. 5 Process model Petri net for motivation example

**结束语** 目前,流程模型挖掘是业务流程管理的核心内容之一,本文在已有研究的基础上,提出基于接口变迁挖掘交互流程模型的新方法,弥补了单纯地基于事件日志挖掘流程模型过程中出现的计算复杂度高、适用范围小等缺陷。

首先,本文提出的交互流程模型挖掘方法优先确定疑似接口变迁(见算法 1),不仅可以极大程度地降低挖掘过程中的计算复杂度,而且能够使得流程模型挖掘更广泛地应用于各行各业中;其次,本文算法 2 结合已有的挖掘方法,检验并减少了算法 1 所确定的疑似接口变迁,这使得所提的挖掘方法更加精确、有效。

对于流程模型挖掘,下一步的研究方向是对挖掘出的流程模型进行一致性检测评估。

参 考 文 献

[1] THALER T, MAURER D, ANGELIS V D, et al. Mining the

Usability of Business Process Modeling Tools; Concept and Case Study[C]// Business Process Management. 2015:152-166.

[2] FANG X, WU J, LIU X. An Optimized Method of Business Process Mining Based on the Behavior Profile of Petri Nets[J]. Information Technology Journal, 2007, 13(1): 86.

[3] LEEMANS S J J, FAHLAND D, AALST W. Discovering Block-Structured Process Models from Incomplete Event Logs[C]// International Conference on Application and Theory of Petri Nets and Concurrency. Springer International Publishing, 2014: 91-110.

[4] FAHLAND D, AALST W. Model repair - aligning process models to reality[J]. Information Systems, 2013, 47(1): 220-243.

[5] TAX N, SIDOROVA N, HAAKMA R, et al. Mining Local Process Models[J]. Journal of Innovation in Digital Ecosystems, 2016, 3(2): 183-196.

[6] BUIJS J C A M, VAN DONGEN B F, AALST W. A genetic algorithm for discovering process trees[C]// IEEE Congress on Evolutionary Computation. IEEE, 2012: 1-8.

[7] AALST W. Decomposing Petri nets for process mining: A generic approach[J]. Distributed and Parallel Database, 2013, 4(31): 471-507.

[8] AALST W M P. A General Divide and Conquer Approach for Process Mining[J]. Computer Science and Information Systems, 2011, 12(2): 1-10.

[9] VAN DER WERF J M E M, KAATS E. Discovery of Functional Architectures From Event Logs[C]// International Workshop on Petri Nets and Software Engineering. 2015: 227-243.

[10] SHEJALE A, GANGAWANE V. Tree Based Mining for Discovering Patterns of Human Interactions in Meetings[J]. International Journal of Engineering Research & Applications, 2014, 4(7): 78-83.

[11] 吴哲辉. Petri 网理论[M]. 北京:机械工业出版社, 2006: 6-22.

[12] SMIRNOV S, WEIDLICH M, MENDLING J. Business process model abstraction based on behavioral profiles[M]// Service-Oriented Computing. Heidelberg: Springer Berlin Heidelberg, 2010: 1-16.

[13] BERA D, VAN HEE K M, VAN DER WERF J M E M. Designing weakly terminating ROS systems[C]// International Conference on Applications and Theory of Petri Nets and Concurrency. Springer Berlin Heidelberg, 2012: 328-347.

[14] AALST W, LOHMANN N, MASSUTHE P, et al. From Public Views to Private Views: Correctness-by-Design for Services[C]// International Conference on Web Services and Formal Methods. 2007: 139-153.

[15] LEEMANS S J J, FAHLAND D, AALST W. Discovering block-structured process models from event logs-a constructive approach[C]// International Conference on Application and Theory of Petri Nets and Concurrency. 2013: 311-329.