软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2015,26(3):491−508 [doi: 10.13328/j.cnki.jos.004764]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 基于 Artifact 快照序列的行为一致性检测方法*

刘海滨[1], 刘国华[2], 黄立明[1], 宋金玲[1]

[1](河北科技师范学院 工商管理学院,河北 秦皇岛 066004)

[2](东华大学 计算机科学与技术学院,上海 201620)

通讯作者: 刘海滨, E-mail: champion_lhb@163.com

**摘 要**: Artifact 行为的一致性检测,是在流程建模、运行之后亟待解决的关键问题之一.针对现有一致性检测技术忽略数据操作方面检测的问题,提出了一种基于 Artifact 快照序列的行为一致性检测方法.首先,利用全序 Artifact 快照序列定义了 Artifact 的行为模式,该行为模式不仅体现了服务的运行轨迹,也描述出了 Artifact 数据属性赋值的状态变化;然后,将 Artifact 行为一致性检测问题转换为语言可判定问题,证明了该问题是一个可判定问题,该过程中,设计一台判定该语言的图灵机作为一致性验证模型,该模型不仅检测了 Artifact 生命周期中服务路径的一致性,同时也检测了生命周期中 Artifact 属性赋值的正确性;进一步地,利用服务-快照关联矩阵的等价转换,给出了行为一致性量化指标中确切度的精确计算方法;最后,通过实例分析及实验对所提出的方法进行了验证.

**关键词**: 业务流程管理;Petri 网;Artifact 生命周期;图灵机;Artifact 行为一致性

**中图法分类号**: TP311

中文引用格式: 刘海滨,刘国华,黄立明,宋金玲.基于 Artifact 快照序列的行为一致性检测方法.软件学报,2015,26(3):491−508. http://www.jos.org.cn/1000-9825/4764.htm

英文引用格式: Liu HB, Liu GH, Huang LM, Song JL. Method for behavior conformance checking based on Artifact snapshot sequences. Ruan Jian Xue Bao/Journal of Software, 2015,26(3):491−508 (in Chinese). http://www.jos.org.cn/1000-9825/4764.htm

## Method for Behavior Conformance Checking Based on Artifact Snapshot Sequences

LIU Hai-Bin[1], LIU Guo-Hua[2], HUANG Li-Ming[1], SONG Jin-Ling[1]

[1](School of Business Administration, Hebei Normal University of Science & Technology, Qinhuangdao 066004, China)

[2](School of Computer Science and Technology, Donghua University, Shanghai 201620, China)

**Abstract**: Behavior conformance checking is a critical issue after process modeling and process running to ensure the correctness and stability of artifact-centric business process model. With the popularization of data-centric design, researches on checking data manipulation of business process becomes more important. This paper proposes a method for checking behavior conformance according to the lifecycle of artifact. First, the artifact behavior can be defined as the total order artifact snapshot sequences. The total order artifact snapshot sequences not only reflect the path of services, but also describe the change of attribute assignment states of the artifact. Next, the problem for behavior conformance checking is proved to be decidable by transforming the problem into the decidability problem of language. A Turing machine is designed as verification model. The model not only measures conformance of service path in the lifecycle, but also evaluates the correctness of attribute assignment of artifact. Further, the calculation method of fitness metric utilizing equivalent conversion of service-snapshot correlation matrix is presented. Lastly, the instantiation analysis and experiments demonstrate the correctness and validity of this method.

**Key words**: business process management; Petri net; Artifact lifecycle; Turing machine; Artifact behavioral conformance

以数据为中心(data-centric)是业务流程管理(business process management,简称 BPM)发展的新趋势.传统业

* 基金项目: 河北省自然科学青年基金(F2015407039); 河北科技师范学院博士基金(2013YB001)
收稿时间: 2014-04-08; 修改时间: 2014-08-08; 定稿时间: 2014-11-21

© 中国科学院软件研究所 http://www.jos.org.cn

务流程模型[1,2]重点是对控制流进行建模,对数据的设计仅处于辅助地位.一些建模方法将数据设计提高到与控制流设计同等重要的位置.其中,以 Artifact[3]为中心的 BPM 是一个典型的代表.近年来,国内外研究学者对以 Artifact 为中心的业务流程建模[4,5]、模型分析[6,7]等展开了深入研究,并取得了显著成果.与传统 BPM 一样,为确保业务流程建模及运行的正确性,Artifact 行为一致性检测是继流程建模、模型分析之后亟待解决的关键问题之一.

一致性检测技术[8,9]是对已知流程模型及其运行后的事件日志进行映射分析,检测流程模型中是否存在违规操作与不一致现象.目前存在很多一致性检测方法[10−12],这些方法都是基于 Petri 网原理进行建模和分析,即:给定一个流程的 Petri 网模型及其执行的事件日志,通过将事件日志还原为 Petri 网模型,然后对两个 Petri 网模型进行一致性分析.基于 Petri 网模型的检测机制存在模型语义过于严格的缺陷,导致很难做出更精确的分析.文献[13]提出一种与模型无关的事件平衡分析方法来避免 Petri 网模型的局限性,但是由于在流程模型中条件设置的多样性,使得进行事件平衡数的计算比较复杂.Gunther 等人[14]通过定义自适应模型(adaptive model)解决了这个问题.基于模糊模型的语义有时会过于松弛,Adriansyah 等人[15]提出一种灵活模型(flexible model)进一步解决了模糊模型存在的不足.在行为一致性检测中,确切度(fitness)[15,16]是对行为一致性进行量化的重要指标,确切度是否精确,直接影响对流程模型的评测结果.Adriansyah 等人[16]提出一种基于成本(cost-based)分析技术来度量流程模型和事件日志之间的确切度,然而该技术也只针对流程中任务执行的成本进行了分析.为了保证检测结果的精确度,同时降低计算复杂度,Munoz-Gama 等人[17]提出一种层次检测方法.该方法先将复杂模型分解成若干可交互的子模型求解,然后再对子模型进行合并.以 Artifact 为中心的业务流程中,Artifact 之间存在多对多(many-to-many)的交互模式[18];并且,一个 Artifact 类型[19]是由其数据对象模式和生命周期模式两部分组成.上述检测方法都着重检测流程模型中的任务(服务)是否一致,而忽略了数据操作方面的一致性检测.文献[20]提出一种 Data-process Graph(DP-graph)模型,有效地实现了面向业务流程的数据模型异常检测.但是该模型只针对流程模型中的数据进行了静态异常检测,并未能实现对运行后的数据进行一致性检测.

在行为一致性检测过程中,忽略流程模型中数据操作方面会降低检测结果的精确度.例如采购审批业务流程中,采购申请单(purchase request,简称 PR)是业务流程中的关键 Artifact.图 1 给出的是 PR 完整生命周期的 Petri 网模型 *M*,图 2 是对 PR 实际运行后的日志进行还原的 Petri 网模型 *L*.

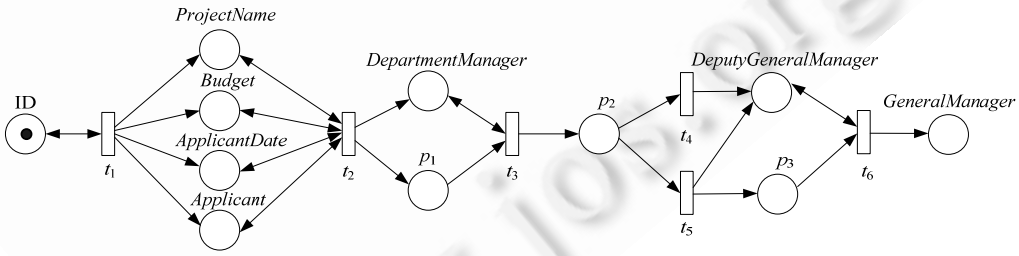



Fig.1 Complete lifecycle of PR represented by Petri net

图 1 PR 完整生命周期的 Petri 网模型

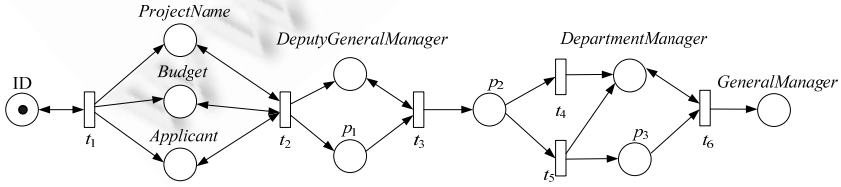



Fig.2 Petri net model replayed by process logs

图 2 PR 执行日志的还原 Petri 网模型

根据已有一致性检测技术分析,*M* 和 *L* 所执行的任务是一致的,即:

© 中国科学院软件研究所 http://www.jos.org.cn

$$M.t_1 \leftrightarrow L.t_1, M.t_2 \leftrightarrow L.t_2, M.t_3 \leftrightarrow L.t_3, M.t_4 \leftrightarrow L.t_4, M.t_5 \leftrightarrow L.t_5, M.t_6 \leftrightarrow L.t_6.$$

然而,当考虑到 Artifact 数据对象属性赋值时,$M$ 和 $L$ 中 PR 属性赋值并不完全一致.从图中可以看出:$L$ 中任务 $t_1$ 执行后未对属性 *ApplicantDate* 进行赋值,而任务 $t_2$ 和 $t_4$ 执行后的属性赋值也与 $M$ 中不一致.实例表明,仅从模型中任务角度考虑一致性问题是有片面性的.

综上所述,在 Artifact 行为一致性检测过程中,不仅要考虑业务流程执行过程中服务路径是否一致,同时还要考虑 Artifact 属性赋值是否正确.为此,首要解决以下几个关键性问题:

(1) 如何形式化描述 Artifact 的行为及 Artifact 行为一致性问题?

(2) 如何设计一种验证模型,使其能够同时对服务路径和 Artifact 属性赋值状态进行检测?

(3) 如何精确计算确切度?

针对以上问题,本文提出了一种基于 Artifact 快照序列的行为一致性检测方法.该方法的主要思想如下:

- 第一,利用 Artifact 快照序列描述 Artifact 行为.现有流程行为描述模型(如 fuzzy model、flexible model 等)虽然简单明了,但在描述 Artifact 行为上明显不够完整.一个 Artifact 快照是在完整生命周期内,某一时刻 $t$,执行某一服务 $S$ 时 Artifact 数据属性赋值的状态.Artifact 快照序列不仅体现了服务的运行轨迹,也描述出了 Artifact 数据属性赋值的状态变化;
- 第二,将 Artifact 行为一致性检测问题转换为语言可判定问题,证明了该问题是一个可判定问题:首先,利用已知模型推导的 Artifact 快照序列定义语言;然后,构造一台判定该语言的图灵机作为一致性验证模型;最后,将实际执行后的快照序列在该模型上进行模拟,模拟过程中不仅检测了 Artifact 生命周期中服务路径的一致性,也检测了生命周期中 Artifact 属性赋值的正确性;
- 第三,利用 Artifact 的行为模式构造服务-快照关联矩阵.通过服务-快照关联矩阵的等价转换操作计算确切度.服务-快照关联矩阵元素中包含了服务和 Artifact 属性赋值结果,提高了计算确切度指标的精确度.

## 1 相关理论

以 Artifact 为中心的业务流程模型[3]主要包含 Artifact 类型、服务和仓库等基本元素.Artifact[7]是一个可标识、自描述的数据单元实体.一个 Artifact 类型由 Artifact 模式(即数据模式)和生命周期模式两部分组成.服务用来操作业务流程中的 Artifact,仓库用来存储 Artifact 生命周期各个阶段的状态.一个服务根据业务规则对 Artifact 进行属性值的添加、更新和归档等操作.每个服务执行后都产生一个 Artifact 实例,并将其存储到相应的仓库中.下面给出相关的定义.

**定义 1(以 Artifact 为中心的业务流程模型,Artifact-centric busienss process model)**. 以 Artifact 为中心的业务流程模型 $M$ 定义为一个多元组$(S,C,E_p,E_a,O,R_1,R_2,R_3,R_4,R_5,R_6;f_1,f_2,f_3,f_4,f_5;H_0)$,其中,

1) $S=\{s_1,\ldots,s_{ns}\}$为服务元素的有穷集;
2) $C=\{c_1,\ldots,c_{nc}\}$为仓库元素的有穷集;
3) $E_p=\{e_1,\ldots,e_{nep}\}$为产生事件的有穷集;
4) $E_a=\{e_1,\ldots,e_{nea}\}$为接收事件的有穷集;
5) $O=\{o_1,\ldots,o_{no}\}$为外部仓库元素的有穷集;
6) $R_1\subseteq S\times C$:表示服务元素和输出 Artifact 之间的关系,其中 $R_1\neq\varnothing$;
7) $R_2\subseteq C\times S$:表示服务元素和输入 Artifact 之间的关系;
8) $R_3\subseteq E_p\times S$:表示服务元素和调用该服务的事件之间的关系;
9) $R_4\subseteq S\times E_a$:表示服务元素和其产生事件之间的关系;
10) $R_5\subseteq S\times O$:表示服务元素和外部 Artifact 之间的关系;
11) $R_6\subseteq S\times E_a$:表示仓库元素和其产生事件之间的关系;
12) $f_1:R_1\rightarrow D_1$,其中,$D_1$ 为前提条件表达式的集合;

© 中国科学院软件研究所 http://www.jos.org.cn

13) $f_2$:$R_2 \to D_2$,其中,$D_2$ 为作用表达式的集合;
14) $f_3$:$C \cup O \to G$,$G$ 为 Artifact 类型的集合;
15) $f_4$:$E_p \cup E_a \to F$,$F$ 为事件类型集合;
16) $f_5$:$R_3 \to K$,$K$ 为消息类型集合;
17) $H_0$ 为初始快照.

例如,根据定义,以 Artifact 为中心的餐馆业务流程模型如图 3 所示.在餐馆业务流程中,客户点菜单(guest check,简称 GC)是该业务流程的关键 Artifact 类,整个业务流程将围绕关键 Artifact 类 GC 展开工作.

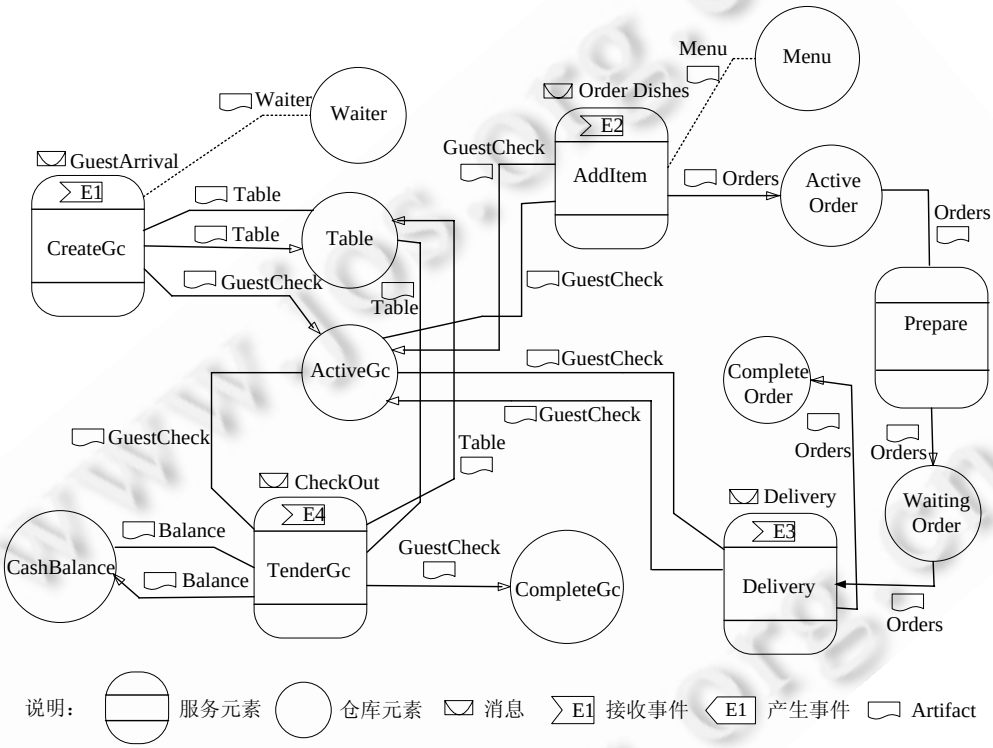



Fig.3 An Artifact-centric business process model of restaurant

图 3 以 Artifact 为中心的餐馆业务流程模型实例

**定义 2(服务[21],service)**. 服务 $S$ 定义为一个 8 元组($n$,$E_T$,$E_S$,$E_P$,$A_r$,$A_w$,$Q$,$\Sigma$),其中,

1) $n$ 是服务的名称描述;
2) $E_T$ 代表触发事件类型的有穷集;
3) $E_S$ 代表挂起事件类型的有穷集;
4) $E_P$ 代表产生事件类型的有穷集;
5) $A_r$ 代表服务读取 Artifact 的有穷集;
6) $A_w$ 代表服务修改或新产生的 Artifact 有穷集;
7) $Q$ 为条件表达式的有穷集,服务读取的 Artifact 必须满足 $Q$ 中的条件;
8) $\Sigma$是由以下语句组成的操作序列:$create(x)$,$def(x,P)$,$x.P=y.R$.其中,(a) $x,y$ 是 Artifact 类型或消息类型的变量;(b) $P,R$ 是 Artifact 类型或消息类型的属性;(c) $create(x)$表示创建一个新的 Artifact;(d) $def(x,P)$表示定义 Artifact 变量 $x$ 的一个属性;(e) $x.P=y.R$ 表示将一个 Artifact 的属性值赋给另一个 Artifact 的属性.

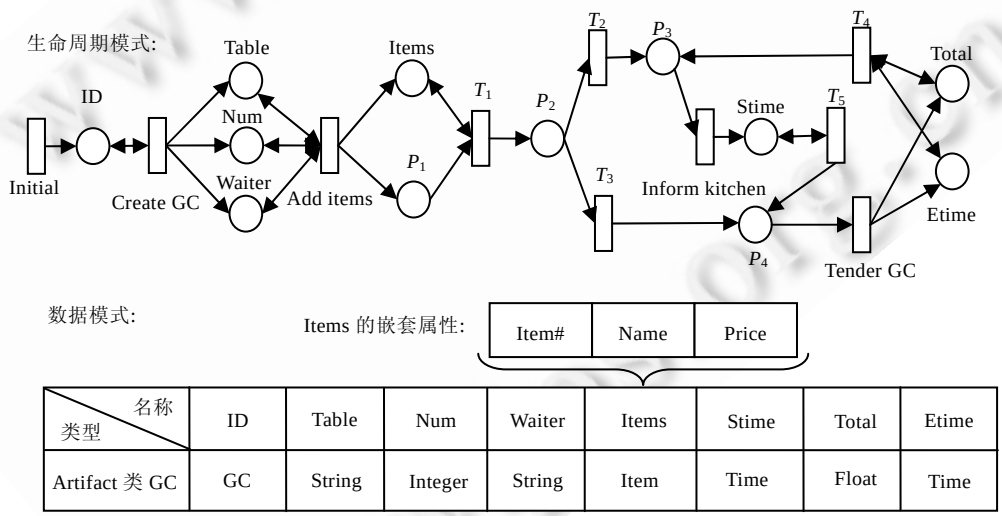
© 中国科学院软件研究所 http://www.jos.org.cn

例如,在餐馆的业务流程模型中的服务有:(1) Create GC,当顾客到达时创建一个新的点菜单 GC(guest check);(2) Add Items,当顾客点菜时为 GC 添加菜品,并创建厨房做菜单 KO(kitchen order);(3) Prepare,取当前没有做的 KO 进行准备;(4) Deliver,将准备好的 KO 送到顾客餐桌,并读取相应的 GC 对照;(5) Tender GC,当顾客结账时计算 GC 的总金额,将现金收取情况写入库 CB(cash balance),一个 GC 最终完成.

**定义 3(Artifact 类型,Artifact class)**. Artifact 类型 $C_A$ 为 2 元组($\mathcal{A}$,$L$),其中,

1) $\mathcal{A}$为 Artifact 模式,它是一个 2 元组($U,\tau$),其中:
   (a) $U$ 是属性的有限集,有一个特殊的属性 $I\in U$ 称为标识符属性(identifier attribute);
   (b) $\tau:U\to D$ 是一个完全映射,$D$ 是域(domain)的集合,$D$ 中至少包含一个标识符域;
2) $L$ 为$\mathcal{A}$的生命周期模式,$L$ 是 2 元组($PN,f$),其中:
   (a) $PN=(P,T,F,K,W,M_0)$是 Petri 网,其中,$P=\{p_1,p_2,\ldots,p_m\}$是库所集,对应 Artifact 类型的属性集合;$T=\{t_1,t_2,\ldots,t_n\}$是变迁集,$T=T_U\cup T_{asst}$,$T_U$ 对应流程模型中的服务集合,$T_{asst}$ 为辅助变迁集合,不对$\mathcal{A}$的属性进行赋值操作;$F\subseteq(P\times T)\cup(T\times P)$是弧集;$K$ 是容量函数,规定 $K(P)\equiv1$,即每个库所最多容纳的托肯数为 1;$W$ 是权函数,规定 $W(F)\equiv1$,即每个变迁发生一次引起相关库所托肯数量的变化为 1;$M_0:P\to\{0,1\}$是初始标识;
   (b) $f:U\to P$ 是完全映射,即$\mathcal{A}$中每个属性都映射为 Petri 网的一个库所,$P=P_U\cup P_{asst}$,$P_U=\{f(u)|u\in U\}$是由$\mathcal{A}$的属性映射的库所集合,$P_{asst}$ 是辅助库所集合,$P_{asst}$ 中的库所与$\mathcal{A}$的属性间没有映射关系.

例如,关键 Artifact 类 GC 的生命周期模式和数据对象模式如图 4 所示.在生命周期模式中,$P_1$,$P_2$,$P_3$ 和 $P_4$ 为辅助库所,它们与 GC 的属性没有映射关系;$T_1$,$T_2$,$T_3$ 和 $T_4$ 为辅助变迁,它们不对 GC 的属性进行赋值操作.




| 名称 / 类型 | ID | Table | Num | Waiter | Items | Stime | Total | Etime |
|---|---|---|---|---|---|---|---|---|
| Artifact 类 GC | GC | String | Integer | String | Item | Time | Float | Time |

Fig.4 Data schema and lifecycle mode of key Artifact GC

图 4 关键 Artifact 类 GC 的数据模式和生命周期模式

**定义 4(Artifact 实例,Aritfact instance)**. 设$\mathcal{A}(U,\tau)$是一个 Artifact 模式,$\mathcal{A}$的实例 $a_{\mathcal{A}}$ 是一个 2 元组($i,\mu$),其中,

1) $\mu$是一个部分映射,为 $U$ 中的属性 $u$ 分配域 $\tau(u)$中一个的元素;
2) $i=\mu(I)$,且 $i\neq NULL$,称为标识符.

流程执行过程中,Artifact 实例是指服务对 Artifact 类型的数据对象模式具体的赋值操作.例如,关键 Artifact 类 GC 完整生命周期中的一个实例见表 1.

© 中国科学院软件研究所 http://www.jos.org.cn

**Table 1** An aritfact instance of Gc in its lifecycle

**表 1** 关键 Artifact 类 GC 的一个实例

| ID | Table | Number | Waiter | Items | *STime* | Total | *ETime* |
|---|---|---|---|---|---|---|---|
| 20120402001 | Rooms 1 | 12 | YangYan | 10 | 11:08 | 210.00 | 13:08 |

## 2 Aritfact 行为一致性问题描述

一个 Artifact 快照反映了生命周期内某一时刻 $t$,执行某一服务 $S$ 后生成的 Artifact 实例,也就是 Artifact 的属性赋值状态.Artifact 的生命周期描述了一类 Artifact 从创建到完成各操作直至归档的整个过程.因此在完整的生命周期内,Artifact 快照序列构成了 Artifact 行为.下面给出相关定义.

**定义 5(Artifact 快照,snapshot)**. 设$\mathcal{A}(U,\tau)$是一个 Artifact 模式,6 元组$(i,S,I_{\mathcal{A}},TS,\lambda,\zeta)$称为$\mathcal{A}$的一个快照,其中,

1) $i$ 为唯一标识符;
2) $I_{\mathcal{A}}$ 是该模式下标识符为 $i$ 的实例集合;
3) $S$ 为操作 Artifact 模式$\mathcal{A}$的服务集合;
4) $TS$ 为服务执行的时间戳集合;
5) $\lambda:S\to I_{\mathcal{A}}$ 为服务集合到实例集合的映射函数;
6) $\zeta:S\to TS$ 为服务集合到时间戳的映射函数.

根据定义 5,关键 Artifact 类 GC 在执行服务 Create GC 时产生一个快照,见表 2.

**Table 2** An aritfact snapshot of GC

**表 2** 关键 Artifact 类 GC 的一个快照

| Service | ID | Table | Number | Waiter | Items | *STime* | Total | *ETime* | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| Create GC | 20120402001 | Rooms 1 | 12 | YangYan | *Null* | *Null* | *Null* | *Null* | 2012-04-02 |

**定义 6(Artifact 行为,Artifact behavior)**. 在 Artifact 的完整生命周期内,Artifact 的行为可以描述为一个满足全序关系的 Artifact 快照序列$(\{H_0,H_1,\ldots,H_{n-1}\},\prec)$.其中,$\{H_0,H_1,\ldots,H_{n-1}\}$为 Artifact 生命周期内的快照集合,$H_0$为初始快照,$n-1$ 为服务执行路径中服务的数量,$\prec$表示快照集合满足全序关系,由生命周期内服务路径和时间戳决定.

例如,表 3~表 7 给出了关键 Artifact 类 GC 在完整生命周期中产生的一个快照序列.快照序列$\{H_0,H_1,H_2,H_3,H_4\}$满足全序关系,由服务执行路径“Initial”$\prec$“Create GC”$\prec$“ADD Items”$\prec$“Inform Kitchen”$\prec$“Tender GC”和时间戳“2012-04-02,08:30:15”$\prec$“2012-04-02, 08:32:17”$\prec$“2012-04-02,08:50:50”$\prec$“2012-04-02, 09:18:32”$\prec$“2012-04-02, 11:51:30”决定.

**Table 3** Initial snapshot $H_0$ of GC

**表 3** Artifact 类 GC 的初始快照 $H_0$

| Service | ID | Table | Number | Waiter | Items | *STime* | Total | *ETime* | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| Initial | 20120402001 | *Null* | *Null* | *Null* | *Null* | *Null* | *Null* | *Null* | 2012-04-02 08:30:15 |

**Table 4** Snapshot $H_1$ of GC

**表 4** Artifact 类 GC 的快照 $H_1$

| Service | ID | Table | Number | Waiter | Items | *STime* | Total | *ETime* | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| Create GC | 20120402001 | Table#1 | 10 | WangYu | *Null* | *Null* | *Null* | *Null* | 2012-04-02 08:32:17 |

**Table 5** Snapshot $H_2$ of GC

**表 5** Artifact 类 GC 的快照 $H_2$

| Service | ID | Table | Number | Waiter | Items | *STime* | Total | *ETime* | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| ADD items | 20120402001 | Table#1 | 10 | WangYu | Item#1 | *Null* | *Null* | *Null* | 2012-04-02 08:50:50 |

© 中国科学院软件研究所 http://www.jos.org.cn

**Table 6** Snapshot $H_3$ of GC

**表 6** Artifact 类 GC 的快照 $H_3$

| Service | ID | Table | Number | Waiter | Items | *STime* | Total | *ETime* | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| Inform kitchen | 20120402001 | Table#1 | 10 | WangYu | Item#1 | AM 09:18 | *Null* | *Null* | 2012-04-02 09:18:32 |

**Table 7** Snapshot $H_4$ of GC

**表 7** Artifact 类 GC 的快照 $H_4$

| Service | ID | Table | Number | Waiter | Items | *STime* | Total | *ETime* | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| Tender GC | 20120402001 | Table#1 | 10 | WangYu | Item#1 | AM 09:18 | 102.00 | AM 11:50 | 2012-04-02 11:51:30 |

为简化问题,本文只对 Artifact 数据属性是否赋值进行一致性检测,并不检测具体属性值的正确性和有效性.为此,下面给出简化 Artifact 快照的定义.

**定义 7(简化 Artifact 快照,simplify snapshot)**. 设$\mathcal{A}$是一个 Artifact 模式,$H$ 是该模式下完整生命周期内的一个快照序列,对 $H$ 进行一个简化变换$\varepsilon$得到$\mathcal{A}$的简化 Artifact 快照 $H^*$,变化方法是:当 $H$ 中属性值为空时,将它的值变换为 0;否则,将它的值变换为 1.

例如,对表 2 给出的关键 Artifact 类 GC 的一个快照进行简化后,得到表 8 的形式.

**Table 8** A simplify snapshot of GC

**表 8** 关键 Artifact 类 GC 的一个简化快照

| Service | ID | Table | Number | Waiter | Items | *STime* | Total | *ETime* | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| Create GC | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

根据简化 Artifact 快照的定义,下面给出 Artifact 行为一致性检测问题的形式化描述.

**定义 8(行为一致性检测)**. Artifact 行为一致性检测形式化定义为 6 元组$(M,L,H_{Simplify},N_1,N_2,f(N_1,N_2))$,其中,

1) $M$ 为已知流程模型集合;
2) $L$ 为流程日志集合;
3) $H_{Simplify}$ 为简化 Artifact 快照集合;
4) $N_1:M\rightarrow H_{Simplify}$ 为流程模型 $M$ 到简化 Artifact 快照序列的映射;
5) $N_2:L\rightarrow H_{Simplify}$ 为流程日志 $L$ 到简化 Artifact 快照序列的映射;
6) $f(N_1,N_2)$为两个简化 Artifact 快照序列的确切度,且 $f(N_1,N_2)\in[0,1]$.

从定义可以得出:Artifact 行为一致性检测就是利用已知流程模型推导出的简化 Artifact 快照序列,与实际运行后产生的简化 Artifact 快照序列进行映射分析,进而判断流程模型中是否存在违规操作和不一致现象.这里的违规操作和不一致现象主要包括服务执行是否正确、Artifact 类型的属性是否赋值.

根据定义,当 $f(N_1,N_2)=1$ 时,说明业务流程中不存在违规操作或不一致的现象;当 $0\leqslant f(N_1,N_2)<1$ 时,表明业务流程中存在违规操作或不一致的现象.据此,我们将检测到业务流程中具体的违规操作.

## 3 Artifact 行为一致性检测方法

根据行为一致性检测的定义,本节给出了 Artifact 行为一致性检测的方法,该方法的执行框架如图 5 所示.

在该框架中:首先,根据已知流程模型推导 Artifact 快照序列;然后,将 Artifact 行为一致性检测问题转换为语言可判定问题,利用推导出的 Artifact 序列定义语言,构造图灵机作为一致性验证模型;最后,将实际运行后得到的快照序列放入图灵机中进行模拟来判定是否一致.进一步,利用 Artifact 快照序列构造服务-快照关联矩阵,通过矩阵等价转换操作计算确切度.

© 中国科学院软件研究所 http://www.jos.org.cn

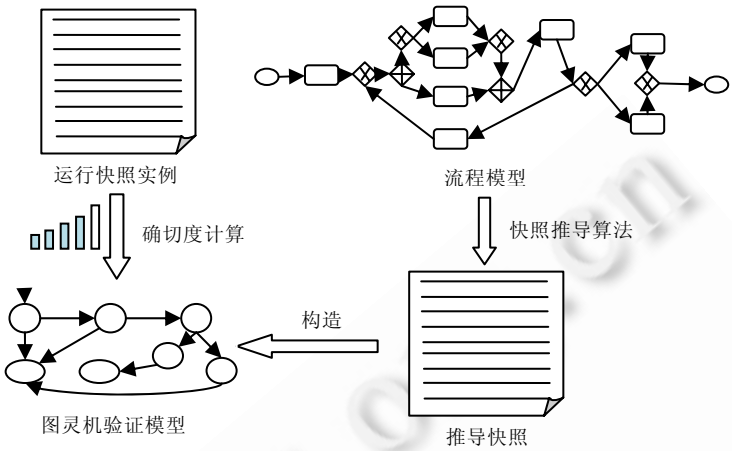



Fig.5 Execution framework of the method

图 5 检测方法的执行框架

### 3.1 从已知流程模型中推导Artifact快照序列

Artifact 快照体现 Artifact 生命周期的运行轨迹.经过不同的服务,Artifact 的属性赋值状态也会产生相应的变化.给定一个业务流程模型 $M$ 和其 Artifact 类型的生命周期模型 $L$,可以推导出简化 Artifact 快照序列.下面给出获取简化 Artifact 快照序列的算法描述.

**算法 1**. 获取 Artifact 简化快照序列算法 *GetArtifactSnapshotSquences*($M,L,h_0$).

输入:流程模型 $M$,Artifact 生命周期模型 $L$;简化初始快照 $h_0$;

输出:简化 Artifact 快照序列 $Q$.

*GetArtifactSnapshotSquences*($M,L,h_0$)

Begin

(1) 初始化 $M,L,Q,h,h',h_0$;

(2) 将简化 Artifact 初始快照 $h_0$ 赋给 $h$;

(3) 验证 $h$ 的正确性:

(4) 如果 $h$ 是无效快照,转步骤(1);否则,转步骤(5);

(5) 添加 Artifact 快照 $h$ 到集合 $Q$ 中;

(6) 调用快照推导子算法 *InferArtifactSnapshot*($h$);

(7) 如果 $h$ 可推导出一个新的简化 Artifact 快照 $h'$,则 $h \leftarrow h'$,转步骤(5);

(8) 否则,return $Q$;

End

整个算法中,子算法 *InferArtifactSnapshot* 是获取简化 Artifact 快照序列过程中最核心部分,它的功能是从已知的简化 Artifact 快照 $h$ 推导出下一个简化 Artifact 快照 $h'$.

**算法 2**. Artifact 快照推导子算法 *InferArtifactSnapshot*($h$).

输入:简化 Artifact 快照 $h$;

输出:新的简化 Artifact 快照 $h'$.

*InferArtifactSnapshot*($h$)

Begin

(1) 获取简化 Artifact 快照 $h$ 的执行服务 $S_h$;

(2) 从流程模型 $M$ 中获取操作该 Artifact 的服务路径 *Path*;

© 中国科学院软件研究所 http://www.jos.org.cn

(3) 从 *Path* 中找到与 $S_h$ 相对应的服务 $S_{mapping}$;

(4) 如果 $S_{mapping}$ 是有效的服务,转步骤(5);否则,转步骤(1);

(5) $S_h \leftarrow S_{mapping}.next$;

(6) 从该 Artifact 生命周期模型 $L$ 中找到与 $S_h$ 映射的变迁 $T$;

(7) 根据 $S_h$ 和 $T$ 对应的库所集合构造一个简化 Artifact 快照 $h'$;

(8) 如果 $h'$不是有效的简化 Artifact 快照,则转步骤(1);

(9) 否则,return $h'$;

End

例如,根据图 3 给定的流程模型和图 4 给定的生命周期模型,利用该算法可以得到关键 Artifact 类 GC 完整生命周期的简化 Artifact 快照序列,见表 9.

**Table 9** Deduced Artifact snapshot sequence of GC

**表 9** 关键 Artifact 类 GC 的推导快照序列

| Service | ID | Table | Num | Waiter | Items | Stime | Total | Etime | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| Initial | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Create GC | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| ADD items | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| Inform kitchen | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| Tender GC | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**定理 1**. *GetArtifactSnapshotSquences* 算法的时间复杂度为 $O(|n|\cdot|m|\cdot p)$.

证明:步骤(7)在整个子算法中花费主要时间.在生命周期模型 $L$ 的 Petri 网模型中,变迁 $T$ 对应于流程模型中的服务元素,库所对应于 Artifact 的属性,库所中的标记(token)表示该属性是否赋值.如果库所中存在一个标记,则表示当前属性已经赋值,用 1 表示;如果库所中不存在标记,表明当前属性尚未被赋值,用 0 表示.设变迁 $T$ 的个数为 $n$(即 Artifact 生命周期中服务的个数),库所的个数为 $m$(即 Artifact 的属性个数).$p$ 是用来维护不同标记状态的次数.因为完成一次 Artifact 快照构造只需单遍扫描库所标记的状态,所以需要的最坏时间复杂度为 $O(|m|\cdot p)$.要完成整个生命周期内的 Artifact 快照序列的构造,需要遍历 $n$ 个变迁对库所标记状态.因此,*GetArtifactSnapshotSquences* 算法的时间复杂度为 $O(|n|\cdot|m|\cdot p)$. □

**定理 2**. *GetArtifactSnapshotSquences* 算法的空间复杂度为 $O(|m^2|)$.

证明:在快照推导过程中,算法需要维护每一次生成的 Artifact 快照属性赋值的状态.设库所的个数为 $m$,一次生成的 Artifact 快照属性赋值的状态中,属性的个数小于等于 $m$,所以 *GetArtifactSnapshotSquences* 算法的空间复杂度为 $O(|m^2|)$. □

### 3.2 行为一致性检测问题转换为语言可判定问题

为了实现同时检测服务路径和 Artifact 数据属性赋值的一致性,本节通过语言判定原理来解决 Artifact 行为一致性问题.根据 Artifact 类型及快照的定义,在进行 Artifact 行为一致性检测时,一个关键的问题是将 Artifact 行为一致性检测问题转换为语言判定问题.下面给出问题转换后的形式化定义.

**定义 9(Artifact 行为一致性语言判定问题)**. Artifact 行为一致性语言判定问题的形式化定义为:$\langle M,H\rangle \in B=\{\langle M,H\rangle | M$ 推导出 $H\}$.其中,

1) $M$ 为一个以 Artifact 为中心的业务流程模型;

2) $\langle M,H\rangle$表示 $M$ 实际运行过程中产生的一个 Artifact 快照序列;

3) $B$ 为一个语言,即该流程模型 $M$ 下可推导出的所有简化 Artifact 快照序列集合.

判定$\langle M,H\rangle$是否属于语言 $B$,如果$\langle M,H\rangle \in B$,即可判定,表示 Artifact 的行为一致;如果$\langle M,H\rangle \notin B$,即不可判定,表示 Artifact 行为不一致.

根据 Artifact 行为一致性语言判定问题的形式化定义,可将 Artifact 行为一致性问题描述成检测语言的成员隶属关系问题.下面的定理将证明语言 $B$ 是可判定语言.

© 中国科学院软件研究所 http://www.jos.org.cn

**定理 3**. $B$ 是一个可判定的语言.

证明:证明思路是构造一个判定语言 $B$ 的图灵机[22]$TM_A=(H,\Sigma,\tau,\delta,h_0,h_{accept},h_{reject})$,其中,

1) $H$ 是该 Artifact 类型的简化快照序列集合;
2) $\Sigma$为字母表={0,1 组成的字符串集合,每个字符串的长度为 Artifact 快照的属性个数};
3) $\tau$为带字母表={$\Sigma,\sqcup$};
4) $\delta$为转移函数,即流程模型中的服务 $S$;
5) $h_0\in H$ 为该 Artifact 类型的初始快照;
6) $h_{accept}\in H$ 为该 Artifact 类型的可接受快照;
7) $h_{reject}\in H$ 为该 Artifact 类型的拒绝快照.

在该验证模型中,首先检测输入$\langle M,H\rangle$,它表示一个流程模型和其执行的 Artifact 快照实例.当 $TM_A$ 收到这个输入时,根据 Artifact 快照的定义,检测该输入是否正确表示流程模型 $M$ 中 Artifact 快照:如果是,则先将其转换为简化 Artifact 快照;如果不是,则拒绝.

然后,$TM_A$ 执行模拟.运行开始时,$TM_A$ 的初始状态是 Artifact 初始快照 $h_0$,状态和位置的更新是由转移函数,即流程模型中的服务来操作完成的.当 $TM_A$ 处理完最后一个 Artifact 快照时,如果 $TM_A$ 处于接受状态,则 $TM_A$ 接受这个输入,表明 Artifact 行为一致;如果 $TM_A$ 处于拒绝状态,则 $TM_A$ 不接受这个输入,表明 Artifact 行为不一致.证毕. □

### 3.3 确切度计算

确切度是衡量一致性机制的主要性能指标.确切度计算的精确与否,将直接影响一致性检测的评测结果.为此,本节首先利用服务路径与 Artifact 快照之间的映射关系,构造服务-快照关联矩阵 $R$;然后,根据矩阵等价转换操作计算确切度.

**定义 10**(**服务-快照关联矩阵**,**service-snapshot correlation matrix**). 设向量 $S=\{S_1,S_2,\ldots,S_m\}$为完整 Artifact 生命周期的服务路径,向量 $A=\{A_1,A_2,\ldots,A_n\}$为简化 Artifact 快照的属性集合.构造一个服务-快照关联矩阵 $R$,它满足$\lambda:S\to A$ 的映射,表示服务路径 $S$ 和 Artifact 快照中属性集合 $A$ 赋值之间的关系,其中,$1\leqslant i\leqslant m,1\leqslant j\leqslant n$.

如公式(1)所示,在服务-快照关联矩阵 $R$ 中,当服务 $S_i$ 对属性 $A_j$ 进行了赋值操作,矩阵元素 $a_{ij}$ 的值为 1;当服务 $S_i$ 未对属性 $A_j$ 进行赋值操作,矩阵元素 $a_{ij}$ 的值为 0.

$$R=\begin{array}{c} \\ S_1\\ S_2\\ S_3\\ \cdots\\ S_m\end{array}\begin{array}{c} \begin{array}{ccccc} A_1 & A_2 & A_3 & \cdots & A_n\end{array}\\ \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n}\\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn}\end{pmatrix}\end{array} \tag{1}$$

**定义 11**(**服务路径确切度**,**fitness of service path**). 设 $S_1$ 为流程模型 $M$ 推导的服务路径,$S_2$ 为流程模型 $M$ 实际执行的服务路径,确切度 $f(S_1,S_2)$定义如公式(2):

$$f(S_1,S_2)=\frac{Equal(|S_1|,|S_2|)}{\max(|S_1|,|S_2|)} \tag{2}$$

其中,$Equal(|S_1|,|S_2|)$为两个服务路径中对应服务相等的数量,$\max(|S_1|,|S_2|)$为两个服务路径中服务最大数.

**定义 12**(**服务-快照关联矩阵确切度**,**fitness of service-snapshot correlation matrix**). 设 $R_1$ 为流程模型 $M$ 推导的服务-快照关联矩阵,$R_2$ 为流程模型 $M$ 实际执行时产生的服务-快照关联矩阵,确切度 $f(R_1,R_2)$定义如公式(3):

$$f(R_1,R_2)=\frac{\sum_{i=j=0}^{m,n}\sum_{i=j=0}^{m,n}Equal(|a_{ij}|,|b_{ij}|)}{\max(|R_1|,|R_2|)} \tag{3}$$

© 中国科学院软件研究所 http://www.jos.org.cn

其中, $\sum_{i=j=0}^{m,n}\sum_{i=j=0}^{m,n} Equal(|a_{ij}|,|b_{ij}|)$ 代表 $R_1$ 和 $R_2$ 矩阵中对应元素相等的数量,$a_{ij}\subseteq R_1$,$b_{ij}\subseteq R_2$;$\max(|R_1|,|R_2|)$代表 $R_1$ 和 $R_2$ 矩阵元素最大数.

**定义 13(模型与日志的确切度,fitness of model and its logs)**. 设 $f(S_1,S_2)$为两个服务路径之间的确切度,$f(R_1,R_2)$为两个服务-快照关联矩阵之间的确切度,流程模型 $M$ 与实际运行后的日志 $L$ 之间的确切度 $f(M,L)$定义如公式(4):

$$f(M,L)=w_1\cdot f(S_1,S_2)+w_2\cdot f(R_1,R_2) \tag{4}$$

其中,$w_1$ 和 $w_2$ 是权值,且 $0<w_1<1$,$0<w_2<1$,$w_1+w_2=1$.

**定理 4**. $f(M,L)$计算的时间复杂度为 $O(|R|\cdot|C^2|)$.

证明:矩阵等价转换操作在整个确切度计算过程中占主要的时间代价.在等价转换过程中,算法首先需要遍历矩阵每一行每一个列元素,设服务-快照关联矩阵的行数为$|R|$,服务-快照关联矩阵的列数为$|C|$,整个确切度计算的时间复杂度为 $O(|R|\cdot|C^2|)$. □

**定理 5**. $f(M,L)$计算的空间复杂度为 $O(|R|\cdot|C|)$.

证明:该计算过程中,需要维护矩阵每一行列值等价转换的过程.设服务-快照关联矩阵的行数为$|R|$,服务-快照关联矩阵的列数为$|C|$,整个确切度计算的空间复杂度为 $O(|R|\cdot|C|)$. □

## 4 方法的正确性分析

### 4.1 理论分析

**定理 6**.基于 Artifact 快照序列的行为一致性检测方法是正确的.

证明:根据 Artifact 行为一致性问题的形式化定义(定义 8),为证明本文提出的行为一致性检测方法是正确的,需要从两个方面进行证明:一方面,要证明利用快照序列构造的图灵机验证模型是正确的;另一方面,要证明利用服务-快照关联矩阵等价转换操作计算确切度是正确的.

关于图灵机验证模型的正确性,定理 3 已给出了详细的证明过程. □

确切度应满足 $f(M,L)\in[0,1]$,它精确地反映了一个流程模型实际执行的结果与预期执行结果之间存在的差异程度.为证明 $f(M,L)$是正确的,首先要证明 $f(M,L)$是一个距离函数.根据距离函数[23]的定义,要证明 $f(M,L)$是度量空间的距离函数,需要证明 $f(M,L)$满足非负性(positiveness)、自反性(reflexivity)、对称性(symmetry)和三角不等式(triangle inequality)这 4 个特性.根据定义 13,$f(M,L)$是由 $f(S_1,S_2)$和 $f(R_1,R_2)$两部分组成.首先,证明 $f(S_1,S_2)$是一个距离函数.假设 $S_1,S_2$ 分别为同一 Artifact 生命周期内两条不同的服务路径,下面分别证明 $f(S_1,S_2)$满足非负性、自反性、对称性和三角不等式特性.

(a) 非负性,即,$f(S_1,S_2)\geqslant 0$.因为 $S_1\cap S_2\subseteq S_1\cup S_2$,所以$|S_1\cap S_2|\leqslant|S_1\cup S_2|$.因此:

$$(|S_1\cap S_2|/|S_1\cup S_2|)\leqslant 1, f(S_1,S_2)=1-(|S_1\cap S_2|/|S_1\cup S_2|)\geqslant 0.$$

(b) 自反性,即,$f(S_1,S_2)=0$.$f(S_1,S_2)=1-(|S_1\cap S_2|/|S_1\cup S_2|)=1-1=0$.

(c) 对称性,即,$f(S_1,S_2)=f(S_2,S_1)$,因为 $f(S_1,S_2)=1-(|S_1\cap S_2|/|S_1\cup S_2|)=1-(|S_2\cap S_1|/|S_2\cup S_1|)=f(S_2,S_1)$.

(d) 三角不等性,即,$f(S_1,S_2)\leqslant f(S_1,S_3)+f(S_3,S_2)$.满足不等式的充要条件是:

$$(|S_1\cap S_3|/|S_1\cup S_3|)+(|S_2\cap S_3|/|S_2\cup S_3|)\leqslant 1+(|S_1\cap S_2|/|S_1\cup S_2|).$$

任意两个集合 $S_1$ 和 $S_2$,都有 $S_1\cap S_2\subseteq S_1\cup S_2$.因此:

$$\begin{aligned}(|S_1\cap S_3|/|S_1\cup S_3|)+(|S_2\cap S_3|/|S_2\cup S_3|)&\leqslant(|\{S_1\cap S_3\}\cup S_2|/|S_1\cup S_2\cup S_3|)+(|\{S_2\cap S_3\}\cup S_1|/|S_1\cup S_2\cup S_3|)\\&\leqslant(|S_1\cup S_2\cup S_3|+|S_1\cap S_2\cap S_3|/|S_1\cup S_2\cup S_3|)=1+(|S_1\cap S_2\cap S_3|/|S_1\cup S_2\cup S_3|)\\&\leqslant 1+(|Ss_1\cap S_2|/|S_1\cup S_2|).\end{aligned}$$

因此,$f(S_1,S_2)\leqslant f(S_1,S_3)+f(S_3,S_2)$.

因为 $f(S_1,S_2)$满足非负性、自反性、对称性和三角不等式这 4 个基本特性,所以 $f(S_1,S_2)$是一个正确的距离函数.$f(R_1,R_2)$的证明过程与 $f(S_1,S_2)$相似,同理,$f(R_1,R_2)$也是一个正确的距离函数.因为 $f(S_1,S_2)$和 $f(R_1,R_2)$都是正确

© 中国科学院软件研究所 http://www.jos.org.cn

的距离函数,且 $f(M,L)=w_1\cdot f(S_1,S_2)+w_2\cdot f(R_1,R_2)$,$w_1$ 和 $w_2$ 是权值,$0<w_1<1$,$0<w_2<1$,$w_1+w_2=1$,所以 $f(M,L)$是正确的距离函数.证毕.因此,基于 Artifact 快照序列的行为一致性检测方法是正确的.

### 4.2 实例分析

以餐馆业务流程为例,假设$\{S_1,S_2,S_3,S_4,S_5\}$为服务集合{Initial,Create GC,ADD Items,Inform Kitchen,Tender GC},关键 Artifact 类 GC 完整的生命周期有两条不同的服务路径,分别为 $P_1=\{S_0,S_1,S_2,S_3,S_4,S_5\}$和 $P_2=\{S_1,S_2,S_3,S_5,S_4\}$.根据算法 1 可推导出两个简化 Artifact 快照序列,见表 10 和表 11.表 12 和表 13 分别为业务流程实际运行后关键 Artifact 类 GC 产生的快照实例.

**Table 10** Deduced Artifact snapshot sequence 1 of GC

**表 10** 关键 Artifact 类 GC 推导快照序列 1

| E-Service | ID | Table | Num | Waiter | Items | Stime | Total | Etime | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $S_2$ | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| $S_3$ | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| $S_4$ | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| $S_5$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Table 11** Deduced Artifact snapshot sequence 2 of GC

**表 11** 关键 Artifact 类 GC 推导快照序列 2

| E-Service | ID | Table | Num | Waiter | Items | Stime | Total | Etime | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $S_2$ | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| $S_3$ | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| $S_5$ | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| $S_4$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Table 12** Running Artifact snapshot instance 1 of GC

**表 12** 关键 Artifact 类 GC 实际运行的快照实例 1

| E-Service | ID | Table | Num | Waiter | Items | Stime | Total | Etime | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | 20120402001 | *NULL* | *NULL* | *NULL* | *NULL* | *NULL* | *NULL* | *NULL* | 2012-04-02 |
| $S_2$ | 20120402001 | Table#1 | 10 | WangYu | *NULL* | *NULL* | *NULL* | *NULL* | 2012-04-02 |
| $S_3$ | 20120402001 | Table#1 | 10 | WangYu | Item#1 | *NULL* | *NULL* | *NULL* | 2012-04-02 |
| $S_4$ | 20120402001 | Table#1 | 10 | WangYu | Item#1 | AM 09:18 | *NULL* | *NULL* | 2012-04-02 |
| $S_5$ | 20120402001 | Table#1 | 10 | WangYu | Item#1 | AM 09:18 | 102.00 | AM 11:50 | 2012-04-02 |

**Table 13** Running Artifact snapshot instance 2 of GC

**表 13** 关键 Artifact 类 GC 实际运行的快照实例 2

| E-Service | ID | Table | Num | Waiter | Items | Stime | Total | Etime | *TimeStamp* |
|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | 20120402001 | *NULL* | *NULL* | *NULL* | *NULL* | *NULL* | *NULL* | *NULL* | 2012-04-02 |
| $S_2$ | 20120402001 | Table#1 | 10 | WangYu | *NULL* | *NULL* | *NULL* | *NULL* | 2012-04-02 |
| $S_3$ | 20120402001 | Table#1 | 10 | WangYu | Item#1 | *NULL* | *NULL* | *NULL* | 2012-04-02 |
| $S_5$ | 20120402001 | Table#1 | 10 | WangYu | Item#1 | AM 09:18 | 102.00 | *NULL* | 2012-04-02 |
| $S_4$ | 20120402001 | Table#1 | 10 | WangYu | Item#1 | AM 09:18 | 102.00 | AM 11:50 | 2012-04-02 |

将关键 Artifact 类 GC 行为一致性问题转换为语言判定问题.根据定义,语言:

$$B=\{(\langle S_1,S_2,S_3,S_4,S_5\rangle,((1,0,0,0,0,0,0,0,1),(1,1,1,1,0,0,0,0,1),(1,1,1,1,1,0,0,0,1),(1,1,1,1,1,1,0,0,1),(1,1,1,1,1,1,1,1,1));(\langle S_1,S_2,S_3,S_5,S_4\rangle,((1,0,0,0,0,0,0,0,1),(1,1,1,1,0,0,0,0,1),(1,1,1,1,1,0,0,0,1),(1,1,1,1,1,0,1,1,1),(1,1,1,1,1,1,1,1,1))\}.$$

首先,根据简化 Artifact 快照的定义,将实际运行的 Artifact 快照实例进行简化,如果 Artifact 属性已有具体值,则将其还原为 1,代表该属性已被赋值;如果该属性值为 *NULL*,将其还原为 0,表示该属性尚未赋值.因此,根据定义,关键 Artifact 类 GC 的两个运行实例分别还原为:

- $H_1=\{\langle S_1,S_2,S_3,S_4,S_5\rangle,((1,0,0,0,0,0,0,0,1),(1,1,1,1,0,0,0,0,1),(1,1,1,1,1,0,0,0,1),$

© 中国科学院软件研究所 http://www.jos.org.cn

(1,1,1,1,1,1,0,0,1),(1,1,1,1,1,1,1,1,1))};和

- $H_2$={⟨$S_1$,$S_2$,$S_3$,$S_5$,$S_4$⟩,((1,0,0,0,0,0,0,0,1),(1,1,1,1,0,0,0,0,1),(1,1,1,1,1,0,0,0,1),(1,1,1,1,1,1,1,0,1),(1,1,1,1,1,1,1,1,1))}.

设计一台识别语言 $B$ 的图灵机 $A_{TM}$ 作为 Artifact 行为一致性验证模型,模型的验证过程如图 6 所示,其中,$h_0$=(1,0,0,0,0,0,0,0,1),$h_1$=(1,1,1,1,0,0,0,0,1),$h_2$=(1,1,1,1,1,0,0,0,1),$h_3$=(1,1,1,1,1,1,0,0,1),$h_4$=(1,1,1,1,1,0,1,1,1),$h_{accept}$=(1,1,1,1,1,1,1,1,1),$h_{reject}$=¬$h_{accept}$.

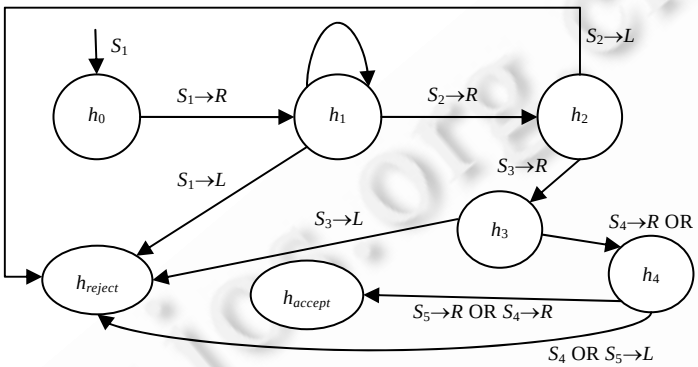



Fig.6 Verification mode for checkng behavioral conformance of Artifact class Gc

图 6 Artifact 类 GC 行为一致性检测的 $A_{TM}$ 验证模型

经过 $A_{TM}$ 验证,Artifact 快照实例 $H_1$ 最终可达到接受状态.因此,$H_1$ 和模型 $M$ 推导的简化 Artifact 快照是一致的,而 $H_2$ 最终是拒绝状态.因此,$H_2$ 和模型 $M$ 推导的简化 Artifact 快照是不一致的.

根据定义 10,可以得到服务-快照关联矩阵.$R_1$ 和 $R_2$ 分别为模型 $M$ 推导的服务-快照关联矩阵,$R_3$ 和 $R_4$ 分别是运行后的快照实例 $H_1$ 和 $H_2$ 得出的服务-快照关联矩阵.

$$R_1=\begin{pmatrix}1&0&0&0&0&0&0&0&1\\1&1&1&1&0&0&0&0&1\\1&1&1&1&1&0&0&0&1\\1&1&1&1&1&1&0&0&1\\1&1&1&1&1&1&1&1&1\end{pmatrix} \tag{5}$$

$$R_2=\begin{pmatrix}1&0&0&0&0&0&0&0&1\\1&1&1&1&0&0&0&0&1\\1&1&1&1&1&0&0&0&1\\1&1&1&1&1&0&1&1&1\\1&1&1&1&1&1&1&1&1\end{pmatrix} \tag{6}$$

$$R_3=\begin{pmatrix}1&0&0&0&0&0&0&0&1\\1&1&1&1&0&0&0&0&1\\1&1&1&1&1&0&0&0&1\\1&1&1&1&1&1&0&0&1\\1&1&1&1&1&1&1&1&1\end{pmatrix} \tag{7}$$

$$R_4=\begin{pmatrix}1&0&0&0&0&0&0&0&1\\1&1&1&1&0&0&0&0&1\\1&1&1&1&1&0&0&0&1\\1&1&1&1&1&1&1&0&1\\1&1&1&1&1&1&1&1&1\end{pmatrix} \tag{8}$$

© 中国科学院软件研究所 http://www.jos.org.cn

根据定义 11~定义 13,计算流程模型和执行日志之间的确切度,设权值:

$$w_1=0.5, w_2=0.5, f(M,H_1)=0.5\times\frac{5}{5}+0.5\times\frac{45}{45}=1, f(M,H_2)=0.5\times\frac{5}{5}+0.5\times\frac{44}{45}\approx 0.99.$$

计算结果显示:如果仅从生命周期中执行服务的角度来考虑,流程模型 $M$ 和快照序列 $H_1$,$H_2$ 完全一致;但是考虑到 Artifact 数据属性赋值情况时,流程模型 $M$ 和快照序列 $H_2$ 并非完全一致.实例分析表明,本文提出的检测方法比已有方法的精确度更高.

## 5 实验分析

本实验的案例为某市某铝业公司的业务流程.图 7 是该公司铝锭铸造生产的基本业务流程,当公司收到订单时创建一个相应的 Artifact Sheet,即流程中的关键 Artifact 类型.操作 Artifact Sheet 的服务包括熔铸、热轧、冷轧、精整和包装.熔铸服务通过投入原料和重新利用的废料制造出带有关键属性卷号的铸锭,并将其写入 Sheet 库.热轧、冷轧、精整服务在读取 Artifact Sheet 后进行生产,制造出满足要求的铸锭,并将其改变的属性再次写入 Sheet 库,直至制造出预期的产品包装入库.入库的方式根据客户的需求主要分为两种:一种是热轧后生产出的半成品铝卷进行包装出售;另外一种是依次经过熔铸、热轧、冷轧、精整工艺后制造而成的成品铝卷包装入库.当产品入库后,Artifact Sheet 也完成了其生命周期,将归档于完成 Artifact Sheet 的库中.

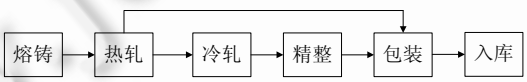
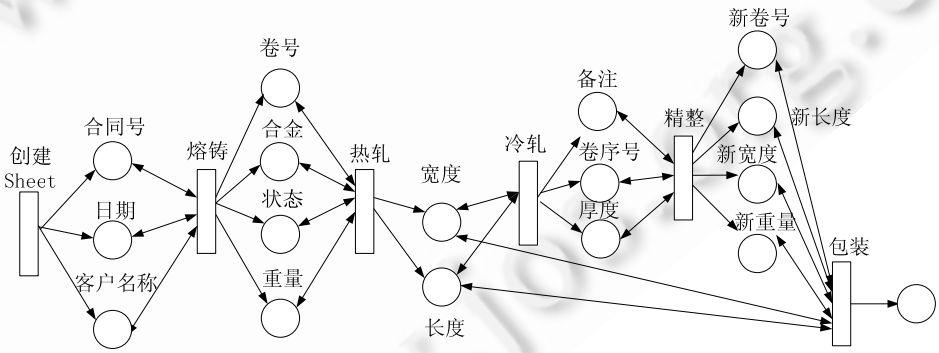



Fig.7 Artiflow model of business process

图 7 业务流程的 ArtiFlow 模型

图 8 中分别给出了 Artifact Sheet 的数据对象模式和生命周期模式的 Petri 网表示.




| 合同编号 | 日期 | 客户名称 | 卷号 | 合金 | 状态 | 重量 | 宽度 | 长度 | 备注 | 卷序号 | 厚度 | 新卷号 | 新长度 | 新宽度 | 新重量 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| string | date | char | char | string | bool | float | float | float | char | string | float | string | float | float | float |

Fig.8 Data schema and lifecycle mode of Artifact Sheet

图 8 Artifact Sheet 的数据对象模式和生命周期模式

实验的硬件环境为 CPU:Inter(R) Core(TM) i3-300;内存为 1GB;软件环境为 Windows XP;数据库为 SQL Server 2000;算法实现的运行环境和语言分别为 Eclipse 3.4 和 Java.ArtiFlow 管理器是本实验室 BPM 研究小组开发的一款以 Artifact 为中心的业务流程建模、运行及检测的原型系统,其中,行为一致性检测器是该系统的一个功能模块.图 9 是 Artifact 行为一致性检测器的运行界面.如图所示,如果服务或 Artifact 属性赋值存在违规操作,检测器将以红色报警显示具体服务和属性赋值的违规位置.

© 中国科学院软件研究所 http://www.jos.org.cn

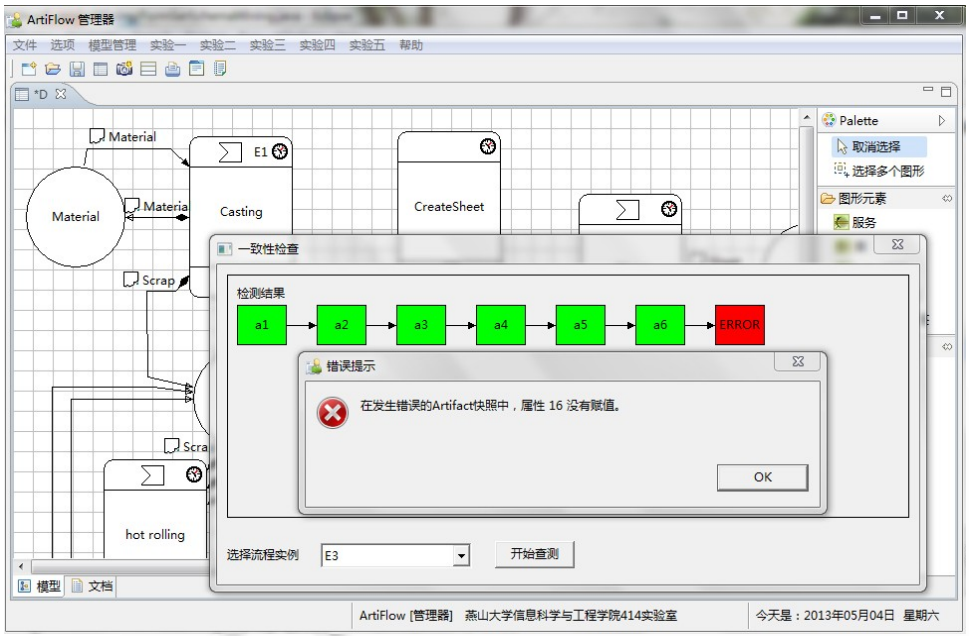



Fig.9 Running interface of behavioral confrmance checker

图 9 Artifact 行为一致性检测器的运行界面

本实验的内容包括两个方面:(1) 对检测方法的确切度进行评测;(2) 评估检测方法的漏检率和误检率.漏检率和误检率[15]是一致性检测性能的两个关键参数:漏检率是指系统或方法未能检测到的违规操作与实际违规操作数的比值;误检率是指系统或方法将非违规操作检测为违规操作的数量与实际违规操作数的比值.从流程日志库中选取 100 个流程实例作为实验分析样本.在 100 个流程实例中,随机修改 10 个流程实例作为违规流程实例.然后,对本文检测方法和传统基于 Petri 网映射方法的确切度、漏检率和误检率进行比较.

服务-快照关联矩阵元素中既包含了服务,又包含了 Artifact 属性赋值结果.因此,利用服务-快照关联矩阵的等价转换操作计算确切度提高了计算确切度指标的精确度.表 14 是不同流程实例数中两种方法的确切度计算结果.

**Table 14** Calculation results of fitness about two kinds of method

**表 14** 两种方法确切度计算结果

| 流程实例数 | 确切度 | | | | | |
|---|---|---|---|---|---|---|
| | 传统 Petri 网映射方法 | 本文检测方法 | | | | |
| | $f(N_1,N_2)$ | $f(S_1,S_2)$ | $f(R_1,R_2)$ | $w_1$ | $w_2$ | $f(M,L)$ |
| 10 | **1.00** | 1.00 | 0.995 | 0.5 | 0.5 | **0.997 5** |
| 20 | **1.00** | 1.00 | 0.996 | 0.5 | 0.5 | **0.998** |
| 30 | **1.00** | 1.00 | 0.997 | 0.5 | 0.5 | **0.998 5** |
| 40 | **1.00** | 1.00 | 0.997 | 0.5 | 0.5 | **0.998 5** |
| 50 | **1.00** | 1.00 | 0.997 | 0.5 | 0.5 | **0.998 5** |
| 60 | **1.00** | 1.00 | 0.997 | 0.5 | 0.5 | **0.998 5** |
| 70 | **1.00** | 1.00 | 0.997 | 0.5 | 0.5 | **0.998 5** |
| 80 | **1.00** | 1.00 | 0.997 | 0.5 | 0.5 | **0.998 5** |
| 90 | **1.00** | 1.00 | 0.998 | 0.5 | 0.5 | **0.999** |
| 100 | **1.00** | 1.00 | 0.998 | 0.5 | 0.5 | **0.999** |

结果发现:

- 基于传统 Petri 网映射分析方法得到的确切度值全为 1,不符合一致性检测预期的结果;
- 而本文提出的检测方法得到更精确的确切度值,与预期结果完全一致.

Artifact 快照序列将业务流程中的数据及服务的执行过程结合在一起,基于 Artifact 快照序列的行为一致性

© 中国科学院软件研究所 http://www.jos.org.cn

检测方法同时考虑到了数据属性赋值状态和服务执行状态.而传统基于 Petri 网映射分析方法仅考虑到服务执行是否一致,忽略了具体属性赋值状态的正确与否.因此,本文提出的检测方法相比传统基于 Petri 网映射分析方法降低了漏检率和误检率:如图 10 所示:本文提出的检测方法平均漏检率为 11.2%,而传统基于 Petri 网映射分析方法的平均漏检率为 55.7%;如图 11 所示:本文提出的检测方法平均误检率为 10.3%,而传统基于 Petri 网映射分析方法的平均误检率为 21.2%.

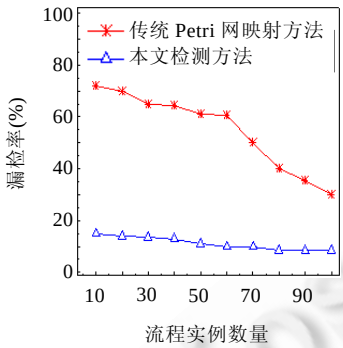



Fig.10 Comparison of miss rate

图 10 漏检率的比较

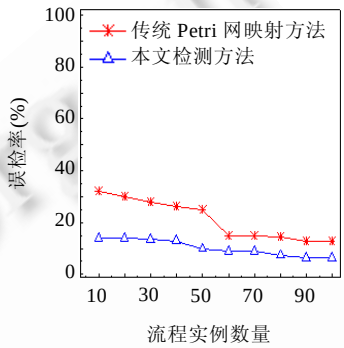



Fig.11 Comparison of error rate

图 11 误检率的比较

## 6 总结与展望

Artifact 行为一致性问题是流程建模、模型分析之后重点解决的关键问题.针对现有一致性检测技术忽略数据操作方面检测的问题,本文提出了一种基于 Artifact 快照序列的行为一致性检测方法.该方法通过对生命周期过程中 Artifact 快照序列进行映射分析,不仅检测流程模型中服务路径是否正确,同时也验证了 Artifact 数据属性赋值状态的正确性,进而提高了一致性检测结果的精确度.

实际上,本文在对 Artifact 数据属性赋值进行检测时,只对 Artifact 属性是否赋值进行检测,并未检测具体属性值的正确性和有效性.下一步的研究目标就是检测在某一时刻下,Artifact 具体属性值的有效性.另一方面,利用服务-快照关联矩阵计算确切度时,随着 Artifact 快照属性个数增多,矩阵也会变得很大,进而会增大矩阵操作的代价.采用属性划分策略,缩小矩阵的列数来优化矩阵操作的时空复杂度,也是下一步研究的重点.

**References**:

[1] Van der Aalst WMP, ter Hofstede AHM, Kiepuszewski B, Barros AP. Workflow patterns. Distributed and Parallel Databases, 2003, 14(1):5−51. [doi: 10.1023/A:1022883727209]

[2] Fan YS, Wu C. Research on a workflow modeling method to improve system flexibility. Ruan Jian Xue Bao/Journal of Software, 2002,13(4):833−839 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/833.htm

[3] Nigam A, Caswell NS. Business Artifacts: An approach to operational specification. IBM Systems Journal, 2003,42(3):428−445. [doi: 10.1147/sj.423.0428]

[4] Bhattacharya K, Caswell NS, Kumaran S, Nigam S, Wu FY. Artifact-Centered operational modeling: Lessons from customer engagements. IBM Systems Journal, 2007,46(4):703−721. [doi: 10.1147/sj.464.0703]

[5] Liu GH, Liu X, Qin HH, Su JW, Yan ZM, Zhang L. Automated realization of business workflow specification. In: Dan A, Gittler F, Toumani F, eds. Proc. of the 1st Int'l Workshop on SOA, Globalization, People, & Work (SG-PAW 2009). Berlin: Springer-Verlag, 2009. 96−108. [doi: 10.1007/978-3-642-16132-2_9]

[6] Bhattacharya K, Gerede C, Hull R, Liu R, Su JW. Towards formal analysis of Artifact-centric business process models. In: Proc. of the Int'l Conf. on BPM 2007. Berlin: Springer-Verlag, 2007. 288−304. [doi: 10.1007/978-3-540-75183-0_21]

© 中国科学院软件研究所 http://www.jos.org.cn

[7] Gerede CE, Bhattacharya K, Su JW. Static analysis of business Artifact-centric operational models. In: Proc. of the IEEE Int'l Conf. on Service-Oriented Computing and Application (SOCA2007). Piscataway: IEEE Computer Society Press, 2007. 133−140. [doi: 10.1109/SOCA.2007.42]

[8] Van der Aalst WMP. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Berlin: Springer-Verlag, 2011.

[9] Leoni MD, Van der Aalst WMP. Data-Aware process mining: Discovering decisions in processes using alignments. In: Shin SY, Maldonado JC, eds. Proc. of the ACM Symp. on Applied Computing (SAC 2013). Portugal: ACM Press, 2013. 1454−1461. [doi: 10.1145/2480362.2480633]

[10] Goedertier S, Martens D, Vanthienen J, Baesens B. Robust process discovery with artificial negative events. The Journal of Machine Learning Research, 2009,10(2):1305−1340.

[11] Van der Aalst WMP, Adriansyah A, Van Dongen B. Replaying history on process models for conformance checking and performance analysis. WIREs Data Mining and Knowledge Discovery, 2012,2(2):182−192. [doi: 10.1002/widm.1045]

[12] Rozinat A, Van der Aalst WMP. Conformance checking of processes based on monitoring real behavior. Information System, 2008, 33(1):64−95. [doi: 10.1016/j.is.2007.07.001]

[13] Luo HB, Fan YS, Wu C. Analysis of event balance in the verification of workflow soundness. Ruan Jian Xue Bao/Journal of Software, 2002,13(8):1686−1691 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/13/1686.htm

[14] Gunther CW, Reichert M, Van der Aalst WMP. Supporting flexible processes with adaptive workflow and case handling. In: Proc. of the 17th Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2008). New York: IEEE Computer Society Press, 2008. 229−234. [doi: 10.1109/WETICE.2008.15]

[15] Adriansyah A, Van Dongen BF, van der Aalst WMP. Towards robust conformance checking. In: zur Muehlen M, Su J, eds. Proc. of the 6th Workshop on Business Process Intelligence (BPI 2010). Hoboken, 2010. 122−133. [doi: 10.1007/978-3-642-20511-8_11]

[16] Adriansyah A, Van Dongen BF, Van der Aalst WMP. Conformance checking using cost-based fitness analysis. In: Chi CH, Johnson P, eds. Proc. of the IEEE Int'l Enterprise Computing Conf. (EDOC 2011). New York: IEEE Computer Society, 2011. 55−64. [doi: 10.1109/EDOC.2011.12]

[17] Munoz-Gama J, Carmona J, Van der Aalst WMP. Hierarchical conformance checking of process models based on event logs. In: Colom JM, Desel J, eds. Proc. of the Applications and Theory of Petri Nets 2013. Berlin: Springer-Verlag, 2013. 291−310. [doi: 10.1007/978-3-642-38697-8_16]

[18] De Leoni M, Van der Aalst WMP, Van Dongen B. Data- and resource- aware conformance checking of business processes. In: Abramowicz W, Kriksciuniene D, Sakalauskas V, eds. Proc. of the Business Information Systems (BIS 2012). Berlin: Springer-Verlag, 2012. 48−59. [doi: 10.1007/978-3-642-30359-3_5]

[19] Hull R, Narendra NC, Nigam A. Facilitating workflow interoperation using Artifact-centric hubs. In: Baresi L, Chi CH, Suzuki J, eds. Proc. of the Int'l Conf. on Service Oriented Computing (ICSOC 2009). Berlin: Springer-Verlag, 2009. 1−18. [doi: 10.1007/978-3-642-10383-4_1]

[20] Liu ZQ, Li HY, Wang L, Qu Q. A data model anomalies detection method for business process model. Chinses Journal of Computers, 2010,33(8):1349−1358 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2010.01349]

[21] Liu HB, Liu GH, Huang LM, Song JL. Artifact-Centric business process models and similarity search method. Computer Integrated Manufacturing Systems, 2013,19(8):1810−1821 (in Chinese with English abstract).

[22] Sipser M. Introduction to the Theory of Computation. 3rd ed., Cengage Learning, 2006. 167−170.

[23] Chavez E, Navarro G, Baeza-Yates R, Marroquin JL. Searching in metric spaces. ACM Computing Surveys, 2001,33(3):273−321. [doi: 10.1145/502807.502808]

**附中文参考文献:**

[2] 范玉顺,吴澄.一种提高系统柔性的工作流建模方法研究.软件学报,2002,13(4):833−839. http://www.jos.org.cn/1000-9825/13/833.htm

[13] 罗海滨,范玉顺,吴澄.工作流合理性验证中的事件平衡分析.软件学报,2002,13(8):1686−1691. http://www.jos.org.cn/1000-9825/13/1686.htm

© 中国科学院软件研究所 http://www.jos.org.cn

[20] 刘之强,李红燕,王磊,曲强.面向业务流程的数据模型异常检测方法.计算机学报,2010,33(8):1349−1358. [doi: 10.3724/SP.J.1016.2010.01349]

[21] 刘海滨,刘国华,黄立明,宋金玲.以业务单据为中心的业务流程模型聚类及相似性查询方法.计算机集成制造系统,2013,19(8):1810−1821.

**刘海滨**(1982－),男,河北承德人,博士,讲师,主要研究领域为业务流程管理,数据库原理,流程挖掘.

**黄立明**(1972－),男,副教授,主要研究领域为信息安全,商务智能.

**刘国华**(1966－),男,博士,教授,博士生导师,主要研究领域为业务流程管理,数据库理论,半结构化数据,数据挖掘.

**宋金玲**(1973－),女,博士,副教授,主要研究领域为信息安全,数据隐私保护.

© 中国科学院软件研究所 http://www.jos.org.cn