软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(3):460−473 [doi: 10.13328/j.cnki.jos.004768]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于过程结构树的过程模型变体匹配技术*

凌济民, 张 莉

(北京航空航天大学 计算机学院,北京 100191)

通讯作者: 凌济民, E-mail: lingjimin@buaa.edu.cn

摘 要: 随着过程模型的不断积累和演化,企业组织常常拥有并管理维护成百上千个业务过程模型.由于建模目标和应用场景的不同,参考模型的裁剪和定制以及模型的更新修改等因素,导致过程模型库中可能存在大量相似的过程模型变体.重点研究如何有效管理和识别过程变体之间的共同点和差异性,即自动化地构建过程模型变体之间的匹配关系.为了支持复杂对应关系,保证匹配关系查找效率和结果的有效性,提出了基于过程结构树的模型元素匹配关系构建技术,并进一步给出了基于树编辑距离的过程模型相似性度量方法.通过针对真实的过程模型集合的实验评估表明,该方法在查全率和查准率指标上表现出了良好的效果.

关键词: 过程模型;过程变体管理;过程模型匹配;过程模型相似度;过程块;过程结构树

中图法分类号: TP311

中文引用格式: 凌济民,张莉.基于过程结构树的过程模型变体匹配技术.软件学报,2015,26(3):460−474. http://www.jos.org.cn/1000-9825/4768.htm

英文引用格式: Ling JM, Zhang L. Matching process model variants based on process structure tree. Ruan Jian Xue Bao/Journal of Software, 2015,26(3):460−474 (in Chinese). http://www.jos.org.cn/1000-9825/4768.htm

Matching Process Model Variants Based on Process Structure Tree

LING Ji-Min, ZHANG Li

(School of Computer Science and Engineering, BeiHang University, Beijing 100191, China)

Abstract: It is common for large enterprises or organizations to maintain repositories of process models. A large number of process model variants may exist in these repositories due to the differences of modeling objective or scenario, customization or tailoring reference models, and model updating modification. This paper focuses on the study of identifying commonalities and differences between process variants, i.e. construction of matching relations between process variants automatically. To support the discovery of complex correspondences and ensure the effectiveness and efficiency of matching results, we propose a matching technique based on the traversal of process structure tree and present a process similarity measuring method based on tree-edit distance. The experimental evaluation based on real-world process model collection shows that an effective precision and recall is achieved.

Key words: process model; process variants management; matching process model; process model similarity; process fragments; process structure tree

随着企业对过程重视程度的不断提高,过程模型在不同企事业单位中正变得日益普遍.由于过程模型的不断积累和演化,企业组织有可能拥有并管理维护成百上千个业务过程模型[1].因为建模目标和应用场景的不同,同一业务过程可能被描述为多个相似的业务过程模型.建模人员也可能根据特定的需求和环境定制、裁剪现有的参考模型(如添加、删除或修改模型中的活动),以节约建模的时间和成本.此外,过程模型也可能被不断地修改成一系列的不同版本.上述因素导致企业组织的过程模型库中可能存在大量相似的过程模型变体(以下简称过程变体).如何有效地管理和识别过程变体之间的共同点和差异性,并保持这些过程变体的一致性演化,是过程

* 基金项目: 国家自然科学基金(61170087, 61370058)
收稿时间: 2014-06-27; 修改时间: 2014-09-30; 定稿时间: 2014-11-21

模型管理的关键问题之一[2].而准确有效地建立过程变体之间的元素匹配关系,是解决这一问题的重要基础[3].此外,除了从微观视角对过程模型的元素进行匹配分析,还可以宏观地度量模型之间整体的匹配程度,为过程模型的查询和检索提供有效的技术支持[4].因此,本文重点研究如何有效地建立过程变体之间的模型元素匹配关系,并在此基础上进一步整体地度量过程模型之间的匹配程度.

目前,对该领域的研究工作通常首先建立过程模型之间的节点匹配关系,然后基于这些节点对应关系和匹配程度,从结构[5−7]或行为[8−10]的视角度量模型整体的相似性程度.现有研究中,模型元素匹配关系的建立通常依赖于唯一的活动命名,或者利用活动名称之间的文本相似度计算来解决不同模型间的语义差异.然而在不同的过程模型之间,除了活动命名的差异性外,由于建模目标和习惯的不同,一个相同的业务逻辑可能被不同的建模人员用不同数量的活动甚至不同的模型结构来实现.因此,仅仅简单地考虑模型元素之间的原子对应关系是不够的,在分析过程变体之间的共性和差异性时,还需要建立并考虑节点组合之间的复杂对应关系.更加全面准确地建立模型元素匹配关系,也将有助于提高模型整体匹配度量的效果[11].

建立过程变体之间复杂对应关系时,主要面临以下挑战:首先,复杂对应关系可能并非完全由原子对应关系组成,若干相似性较低的节点对有可能组合出相似程度较高的复杂对应关系;此外,节点组合的选择显然具有指数级的时间复杂度,因此需要一种有效的组合策略来保证效率.通过引入单入口单出口过程模型片段(single-entry-single-exit process fragment,简称 SESE 过程块)的概念[12],在过程模块能够很大程度上地表达相对完整独立业务逻辑的假设基础上,提出了基于过程结构树(process structure tree,简称 PST)[12]的模型元素对应关系的构建方法,高效率地建立了过程模型元素之间的复杂对应关系;然后,基于过程模型相应的过程结构树之间的节点匹配关系,提出了基于树编辑距离[13]的过程模型相似性度量方法;最后,利用从 BPM AI 过程模型库(BPM academic initiative repository)[14]中选取的过程变体集合,对本文提出的方法进行了实验评估.

本文的贡献可以分为两个方面:1) 提出了基于 SESE 过程块的过程模型复杂对应关系的构建方法,相比现有支持原子对应关系[6,15]或者单对多复杂对应关系[5,16]的方法,在支持更全面合理对应关系构建的同时又保证了查找效率;2) 提出了基于 SESE 过程块对应关系度量过程模型相似度的方法,探索了从树编辑距离的视角度量模型之间的匹配程度,并且实验评估表明,该方法能够较好地应用于过程模型查询和相似性分析等场景中.

本文第 1 节通过模型示例对研究问题进行分析,并介绍本文工作的相关背景和概念.第 2 节介绍基于过程结构树的模型片段对应关系的构建方法.第 3 节阐述基于树编辑距离的过程模型相似性度量算法.第 4 节讨论本方法实验评估的结果和存在问题的.第 5 节讨论并比较本文的相关工作.第 6 节对全文的研究工作进行总结,并提出接下来的研究方向.

1 问题背景

本节首先通过一组过程变体的示例对本文的研究问题进行了解释说明,然后介绍了本文方法有关的背景和概念,包括过程模型、SESE 过程块以及过程结构树等.

1.1 一组过程变体的示例

图 1 给出了两个使用 BPMN 描述的请求处理过程.可以看出:两个模型表达的业务流程是相似的,都是首先创建或获取一个请求,然后对请求进行处理和状态的跟踪.但是仍然不难发现,两个模型之间存在不少差异,不仅包括活动命名的不同(如,模型(a)中的活动 track request status 和模型(b)中的活动 tracking state of request),而且还存在由于建模目标和粒度的不同引起的模型结构的差异(如,模型(a)中的活动 pull request 在模型(b)中被分解为两个活动操作 search request 和 load request 来表达).因此,在模型间建立元素对应关系时,不仅需要匹配具有相同命名的模型元素,也需要考虑这些具有差异但又相似的地方.为了定量地表示他们的相似程度,每组对应关系可以被分配一个匹配值(match value),匹配值的计算和阈值的选取是对应关系建立的关键.我们在图 1 中通过活动的颜色和标记给出了所建立的活动对应关系,包括原子对应关系(例如,模型(a)中的活动 A 对应模型(b)中的 A),单对多复杂对应关系(例如,模型(a)中的活动 C 对应模型(b)中的{$C1$,$C2$})以及多对多复杂对应关系(例如,模型(a)中的活动{$B1$,$B2$}对应模型(b)中的{$B3$,$B4$}).

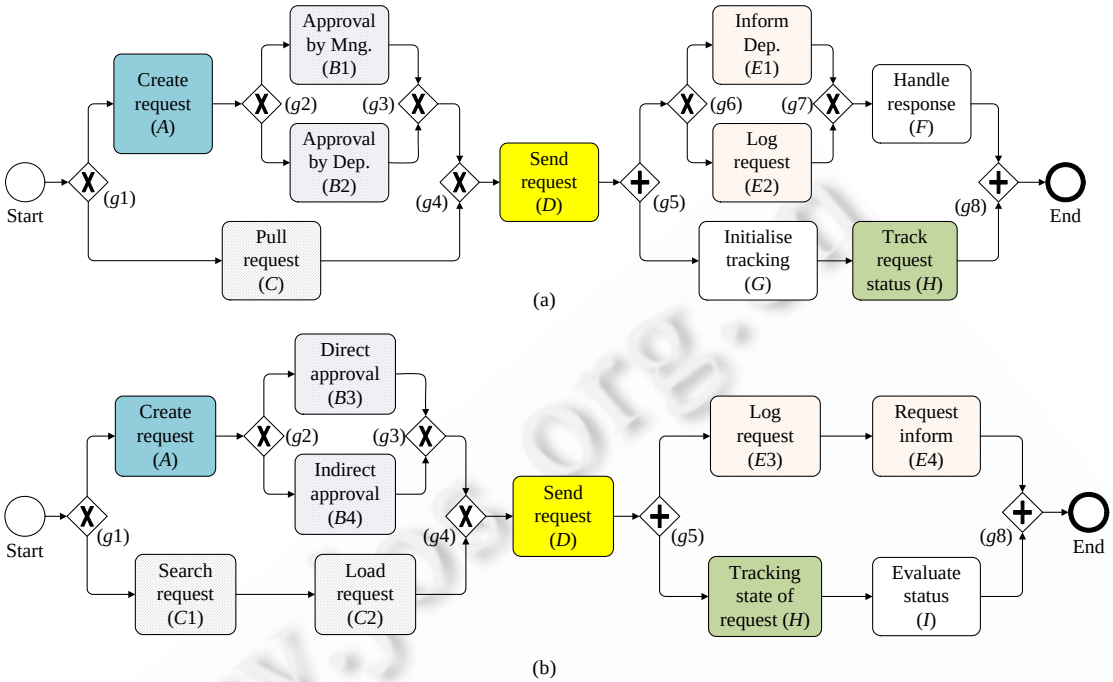


Fig.1 Two sample BPMN process models marked by matching relations

图 1 两个 BPMN 过程模型示例及其之间的对应关系

1.2 过程模型描述和假设

目前,在实践中已有多种过程建模语言用于描述业务过程,包括业务过程建模与标注(business process model and notation,简称 BPMN)、事件驱动过程链 (event-driven process chains,简称 EPC)以及 UML 活动图(UML activity diagram)等.这些过程建模语言至少包括两类基本节点:活动节点和控制节点.活动节点描述由人工或软件程序执行的任务,而控制节点描述了活动节点间的行为顺序.我们希望本文提出的方法能够应用于不同建模语言,因此,后文中将使用过程模型图 PMG(process model graph)而非某种特定的建模语言来阐述过程模型匹配的过程.一个 PMG 是由节点和边的集合组成的连通有向图,其中,节点可能包含命名标签、类型、资源使用等属性.

定义 1(过程模型图 PMG). 过程模型图 PMG 是一个五元组(N,E,T,Ω,α),其中,

- N 表示模型中所有节点的集合;
- $E\subseteq N\times N$ 表示节点间有向边的集合;
- T 表示一组属性名的集合,例如 TYPE,LABEL,RESOURCE,INPUT,OUTPUT 等属性;
- Ω表示由若干个字符串组成的集合;
- $\alpha:N\to(T\to\Omega)$表示节点和属性的映射关系,其中,属性由属性名和字符串属性值的映射构成.

图 2 给出了图 1 中过程模型(b)相应的过程模型图 PMG 的描述.可以看出:不同类型的节点在 BPMN 中用不同的图符描述,但在 PMG 中都统一表示为圆型的节点.节点的类型和名称等均作为节点拥有的属性,根据过程模型的需要,也可以添加更多的附加属性,如资源使用、输入/输出数据等.

为了限制问题的范围,本文假设所有涉及的过程模型是块结构化的(block-structured).当一个过程模型中的顺序关系、分支和循环都是由具有明确开始和结束节点的块表示时,我们称该过程模型是块结构化的[17].这些块可以相互嵌套但不能重叠.相比于非块结构化的模型,块结构化的模型更易于用户理解,并且出现结构错误的可能性更小[18].文献[19]的案例研究表明:块结构化的过程模型在模型库中所占比例极高,并且一个非块结构化的过程模型在通常情况下能够被转换为块结构化的模型[17].因此,我们使用这一假设保证本文基于过程块方法

的有效性,并且该限制并不会对本文方法的适用性产生较大影响.

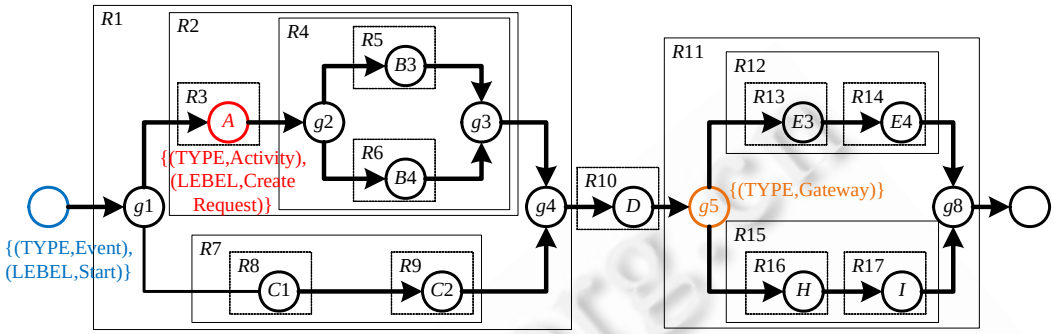


Fig.2 Corresponding PMG and fragments dividing result of process (b) in Fig.1

图 2 图 1 中过程模型(b)相应 PMG 的描述以及包含的 SESE 过程块

1.3 SESE过程块和过程结构树

一个块结构化的过程模型可以层次分解为一组模型模块组成的集合,本文引入 SESE 过程块的概念[12]对过程模型进行层次化分解.文献[18]分析指出,过程模块通常能够表达一个相对完整独立业务逻辑.因此,SESE 过程块可以作为复杂对应关系识别的理论基础.

定义 2(SESE 过程块). 令 x,y 为过程模型图 G 中的两个节点:如果每条从起始节点到 y 的通路上都包含 x,则称 x 控制 y;如果每条从 y 到终止节点的通路上都包含 x,则称 y 受控于 x.同理,边之间的控制与受控关系也可以类似地定义.过程模型图 G 中的 SESE 过程块是一个由两条相异的边(a,b)定义的过程块:

- a 控制 b;
- b 受控于 a;
- 每条包含 a 的回路中均包含 b,每条包含 b 的回路中均包含 a.

根据上述形式化的定义,过程块就是过程模型图中由单一入口开始并且由单一出口结束的任何区域.图 3 给出了对示例过程模型图的过程块划分结果,所有的过程块用虚线方框标出.我们可以认为一个活动节点本身是一个过程块,并且整个过程模型图也可以当作是一个过程块.

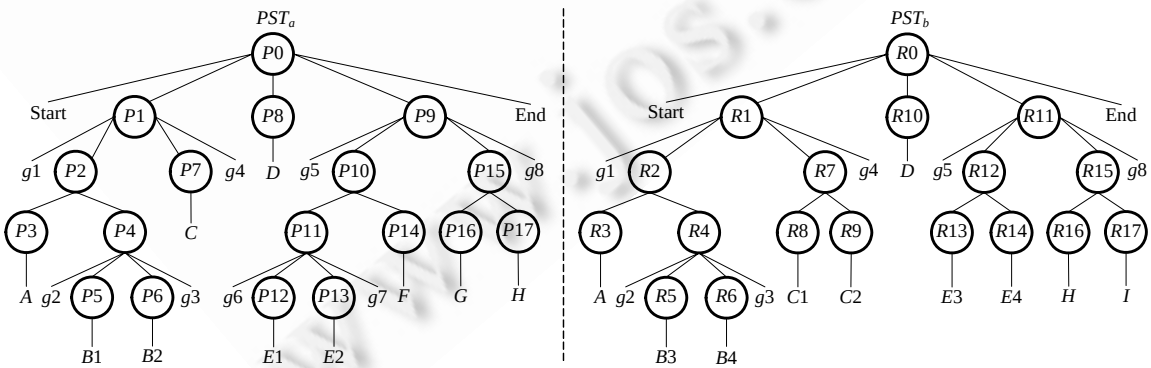


Fig.3 Corresponding process structure trees of two processes in Fig.1

图 3 图 1 中过程模型(a)和过程模型(b)对应的过程结构树 PST

事实上,图 3 所示的过程模型图包含的过程块比图中标识出来的要更多.例如,由 $R10$ 和 $R11$ 合并组成的模型片段(记作 $R10\cup R11$)同样也是 SESE 过程块,但仅有图中虚线框标出的过程块才是规则的(canonical).该概念的解释是:两个过程块 $F1$ 和 $F2$ 是顺序相邻的当且仅当 $F1$ 的出口边与 $F2$ 的入口边相同.一个过程块 F 不是规

则的当且仅当存在过程块 X,Y,Z,其中,$F=X\cup Y$,并且 F 和 Z 顺序相邻;否则,我们称 F 为规则的过程块.我们约定,下文所有提到的过程块均是指规则的过程块.

已有研究证明,两个过程块之间要么相互嵌套要么不相交[12].因此,一个过程模型的过程块可以构成一个树形结构,称为过程结构树 PST(process structure tree)[12],它与经典的程序结构树(program structure tree)[20]的概念类似.图 3 给出了图 1 中两个过程模型相应的过程结构树,图中圆圈表示过程块,根节点表示整个过程模型,叶子节点表示模型节点.利用环路等价算法[20]对过程模型图进行处理,可以在线性时间复杂度内生成与其对应的过程结构树.这一理论基础,保证了我们方法的有效性和效率.

2 基于 PST 的过程变体元素匹配

建立过程变体之间准确的元素对应关系,是识别模型差异性、保证变体演化一致性的重要前提.为了支持模型间复杂对应关系的建立并保证方法的效率,避免节点间无意义的随意组合,我们在能够表达相对独立完整业务逻辑的过程块之间寻找对应关系,并度量过程块之间的匹配程度.我们将过程模型元素对应关系的构建问题转化为两棵过程结构树之间节点映射关系的查找过程,模型间的元素对应关系包括:

- 原子对应关系:两个叶子节点之间的映射关系;两个包含单一子节点的非叶子节点之间的映射关系;
- 单对多对应关系:包含单一子节点的非叶子节点与包含多个子节点的非叶子节点之间的映射关系;
- 多对多对应关系:两个包含多个子节点的非叶子节点之间的映射关系.

本节定义了过程结构树之间两个节点对应关系成立的条件,并分别提出叶子节点之间以及非叶子节点之间匹配程度的度量方法.

2.1 PST节点之间映射关系的判别依据

PST 节点之间映射关系的判别依据,决定了两个过程模型中哪些过程块之间存在对应关系.我们根据 PST 节点类型的不同,基于节点之间的相似性程度,提出叶子节点之间以及非叶子节点之间映射关系的判别依据.

定义 3(PST 叶子节点之间映射关系的判别依据). 令 n 和 m 分别为过程结构树 PST_1 和 PST_2 中的叶子节点,分别代表过程模型图 PMG_1 和 PMG_2 中的节点元素 $node_1$ 和 $node_2$,它们之间存在候选映射关系当且仅当它们之间的匹配值 $Match(n,m)$不低于某给定阈值:

$$Match(n,m)=SimN(node_1,node_2)\geqslant cutoff_n \tag{1}$$

其中,叶子节点 n 和 m 之间的匹配值可以由节点 $node_1$ 和 $node_2$ 之间的节点相似度 $SimN(node_1,node_2)$表示,其值域为[0,1],具体计算方法在第 2.3 节中的定义 11 中说明.参数 $cutoff_n$ 表示叶子节点间映射关系成立的判别阈值,取值范围为[0,1],通过实验评估将该值设置为 0.55.

候选映射关系之间可能出现重叠,即一个节点与多个节点之间存在映射关系.这种情况下,仅保留其中匹配值最高的映射关系(若存在多个相同的最高值,任意选取其中之一),将其他候选映射关系删除,从而最终得到节点之间的映射关系.另外,该规则在下面提出的非叶子节点之间映射关系中同样适用.

定义 4(PST 非叶子节点之间映射关系的判别依据). 令 P 和 R 分别为过程结构树 PST_1 和 PST_2 中的非叶子节点,若 P 和 R 均有唯一的子节点(分别记作 n 和 m),则 P 和 R 之间存在候选映射关系当且仅当它们的子节点 n 和 m 之间存在候选映射关系,且 P 和 R 之间的匹配值 $Match(P,R)=Match(n,m)$;否则,P 和 R 之间存在候选映射关系当且仅当它们之间的匹配值 $Match(P,R)$不低于某给定阈值:

$$Match(P,R)=\omega_1\cdot\frac{Num(P,R)}{\mathrm{Min}(P,R)}+\omega_2\cdot SimT(Str(P),Str(R))+\omega_3\cdot\frac{SimN(^{\circ}P,^{\circ}R)+SimN(P^{\circ},R^{\circ})}{2}\geqslant cutoff_p \tag{2}$$

上述表达式定义了 PST 非叶子节点 P 和 R 之间的匹配值的计算方法,公式中考虑了 3 种因素对相似程度的影响:两者的子孙叶子节点之间存在的对应关系的覆盖率、两者的子孙叶子节点属性的综合文本相似度以及两者相应的上下文模型节点的相似度,具体地说:

- $Num(P,R)$表示 P 和 R 各自的子孙叶子节点之间存在的映射关系数量;
- $\mathrm{Min}(P,R)$表示 P 和 R 中包含子孙叶子节点较少者的子孙叶子节点数量;

- $Str(P)$表示 P 中子孙叶子节点对应的模型元素所有属性值串联构成的字符串.例如,对于图 3 中的 PST_b,$R4$ 的串联属性串 $Str(R4)$="*GTW Direct Approval ACT Indirect Approval ACT GTW*"(即节点按深度优先顺序排列).如果过程模型中节点包含其他更多的属性,也可以类似地添加进该字符串中;
- $SimT(Str_1,Str_2)$表示字符串 Str_1 和 Str_2 之间的文本相似度,其值域为[0,1],具体计算方法在第 2.3 节中的定义 7 中详细说明;
- $^\circ P$ 和 P° 分别表示 P 对应的过程块的前后顺序相邻的模型节点;
- $\omega_1,\omega_2,\omega_3$ 分别表示赋予对应关系覆盖率、文本相似度以及上下文节点相似度的权重,并且三者之和为 1.通过实验分析,本文设定 ω_1=0.2,ω_2=0.7,ω_3=0.1;
- $cutoff_p$ 表示非叶子节点间的判别阈值,取值范围为[0,1],通过实验评估,我们将该值设置为 0.6.

在过程块包含的节点之间存在更多的对应关系,意味着过程块之间相似程度很可能更高.因此,在从上述表达式中,两个过程块之间的相似程度除了考虑过程块包含的所有节点属性的文本相似度之外,也考虑了过程块中所包含节点的对应关系的覆盖率.另外,过程块在模型中的前后相邻节点的匹配程度也会对其相似度产生影响,因此,我们将过程块的匹配程度定义为上述 3 项因素的加权平均值.

2.2 PST节点之间映射关系的构建方法

由于非叶子节点的映射关系的判别条件依赖于叶子节点之间的对应关系,因此我们首先需要构建叶子节点之间的映射关系,然后再构建非叶子节点之间的映射关系.首先,按照深度优先顺序对第一棵过程结构树 PST_1 进行遍历,遍历到叶子节点时,在另一棵 PST_2 中遍历寻找符合判别依据的叶子节点,并将该对应关系及其匹配值保存.全部遍历完成之后,需要根据匹配值的高低最终构建出不重叠的映射关系.最终建立的叶节点映射关系及其匹配值是:(*Start*,*Start*,1),(*g*1,*g*1,0.83),(*A*,*A*,1),(*g*2,*g*2,0.67),(*g*3,*g*3,0.67),(*C*,*C*2,0.88),(*g*4,*g*4,0.72),(*D*,*D*,1),(*g*5,*g*5,0.58),(*E*2,*E*3,1),(*H*,*H*,0.92),(End,End,1).

非叶子节点之间对应关系的构建与上一步类似,在 PST_1 中遍历到非叶子节点时,在另一棵 PST_2 中遍历寻找符合判别依据非叶子节点,并将该对应关系及其匹配值保存.我们以 PST_1 中的 $P11$ 为例,如图 4 所示.

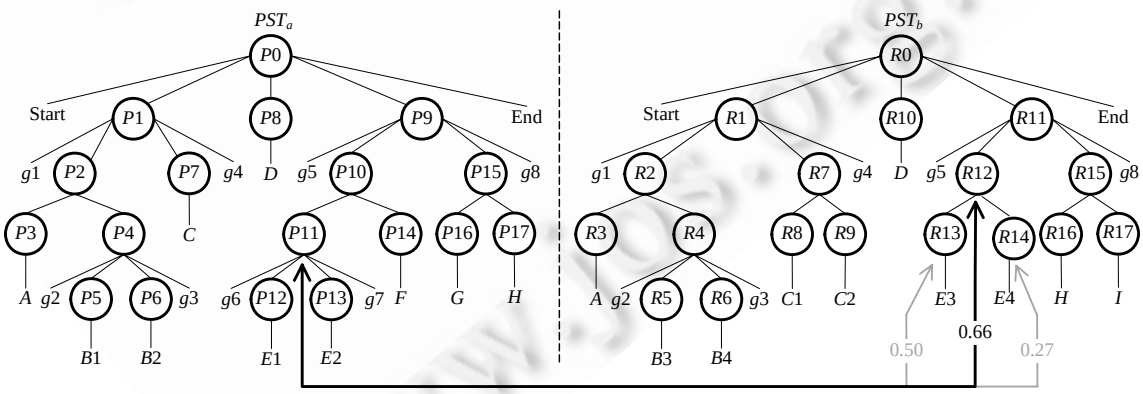


Fig.4 Constructing correspondences between nodes of PST_s

图 4 构建过程结构树之间的节点对应关系 PST_s

假设算法寻找到了 PST_2 中的 $R12$,在两过程块包含的节点中,活动 $E2$ 和 $E3$ 之间存在一条对应关系,因此,对应关系的覆盖率为 0.5,$P11$ 的串联属性串 $Str(P11)$="*GTW Inform Department ACT Log Request ACT GTW*",$R12$ 的串联属性串 $Str(R12)$="*Log Request ACT Request Inform ACT*",从而 $SimT(Str(P11),Str(R12))\approx0.75$.此外,$P11$ 和 $R12$ 的上下文节点相似度约为 0.29,计算得到匹配值 $Match(P11,R12)\approx0.66>cutoff_p$,从而认为 $P11$ 和 $R12$ 之间存在候选映射关系.

遍历所有节点后并去除重叠关系后,建立过程块之间映射关系及其匹配值如下:($P1$,$R1$,0.76),($P2$,$R2$,0.77),($P3$,$R3$,1),($P4$,$R4$,0.71),($P7$,$R9$,0.88),($P8$,$R10$,1),($P9$,$R11$,0.67),($P11$,$R12$,0.66),($P13$,$R13$,1),($P17$,$R16$,0.92).

2.3 过程模型节点相似性度量

为了 PST 之间叶子节点的匹配值,需要定义过程模型节点之间的相似性度量,即:给定两个过程模型图中的两个节点,返回表征它们相似程度的值,取值范围为[0,1].首先,我们认为只有相同类型的模型节点之间可能存在相似关系,例如活动节点与控制节点之间的相似度为 0,因此定义类型相似度如下:

定义 5(类型相似度). 给出$(N_1,E_1,T_1,\Omega_1,\alpha_1)$和$(N_2,E_2,T_2,\Omega_2,\alpha_2)$两个 PMG,对于节点 $n_1\in N_1$ 和 $n_2\in N_2$,它们之间的类型相似度记作:

$$SimY(n_1,n_2)=\begin{cases}1, & \alpha_1(n_1).TYPE=\alpha_2(n_2).TYPE\\ 0, & \text{otherwise}\end{cases} \tag{3}$$

对于事件和活动等具有明确命名标签的模型节点,我们可以通过它们具有的属性值来度量节点间的相似程度.属性相似度是两个节点公共属性值的文本相似度的加权平均:

定义 6(属性相似度). 给出$(N_1,E_1,T_1,\Omega_1,\alpha_1)$和$(N_2,E_2,T_2,\Omega_2,\alpha_2)$两个 PMG,对于节点 $n_1\in N_1$ 和 $n_2\in N_2$,它们之间的属性相似度记作:

$$SimA(n_1,n_2)=\sum_{\substack{(t_{i1},l_{i1})\in\alpha_1(n_1)\\(t_{i2},l_{i2})\in\alpha_2(n_2),1\leqslant i\leqslant x,t_{i1}=t_{i2}}}\omega_i\cdot SimT(l_{i1},l_{i2}) \tag{4}$$

其中,x 表示两个节点具有的公共属性的数量,ω_i 表示赋予各个属性的权重.例如,在本文实验评估的模型集合中,节点属性仅包括在任何模型中都存在的命名标签 LABEL 和类型 TYPE 两种,并未包含资源和数据等其他属性,因此应用的属性相似度计算式可以特例化为 $SimA(n_1,n_2)=SimT(\alpha_1(n_1).LABEL,\alpha_2(n_2).LABEL)$.

上述定义中的 *SimT* 表示两个字符串的文本相似度.考虑到在过程模型的节点名称中经常存在的近义词和单词顺序的影响,例如 *Check Invoice* 和 *Invoice Verified*,我们引入同义词的概念,将字符串之间的文本相似度定义如下[8]:

定义 7(文本相似度). 假设 f_1 和 f_2 分别是字符串 l_1 和 l_2 所含单词构成的多重集合(允许集合中含有相同元素),则字符串 l_1 和 l_2 的文本相似度可以用两个字符串拥有的相同或同义的单词的数量来衡量,记作:

$$SimT(l_1,l_2)=\frac{\omega_i\cdot|f_1\cap f_2|+\omega_j\cdot\sum_{(s,l)\in f_1\setminus f_2\times f_2\setminus f_1}sym(s,l)}{\max(|f_1|,|f_2|)} \tag{5}$$

其中,$sym(s,l)$表示单词 s 和 l 是否为同义词(基于字典查询判断):如果是,则返回值为 1;否则,返回 0.参数 ω_i 和 ω_j 分别是赋予相同词和同义词的权重,我们根据文献[8]中的推荐值设定为 1 和 0.75.在考虑文本相似度时,我们忽略特殊字符、字母大小写和一些常用词,并使用词干提取算法统一动词时态[21].

对于控制节点(网关 gateway)而言,他们除了类型之外并不具有其他的属性,因此属性相似度并不适用.我们采用上下文相似度对它们进行评估,上下文相似度表示节点的前驱或后继节点组成的集合之间的相似程度.我们注意到:由于控制节点的前驱(后继)节点仍有可能为控制节点,例如图 1 模型(a)中的节点 $g6$.为了保证此类节点的上下文相似度能够有效计算,我们首先引入传递前驱节点集合和传递后继节点集合的概念:

定义 8(传递前驱节点集合,传递后继节点集合). 在一个 PMG:(N,E,T,Ω,α)中,两个节点 $n,m\in N$ 之间存在路径当且仅当存在一系列节点 $n_1,\ldots,n_k\in N$,其中,$n=n_1$,$m=n_k$,并且对于所有 $i\in 1,\ldots,k-1$ 而言,$(n_i,n_{i+1})\in E$ 成立.控制节点路径是一条路径,满足:其中任意节点 $n'\in\{n_2,\ldots,n_{k-1}\}$均为控制节点,记作 $n\Rightarrow m$.定义 n 的传递前驱节点集合为 $\{n'\in N|n'\Rightarrow n\}$,记作 n^{in};n 的传递后继节点集合为$\{n'\in N|n\Rightarrow n'\}$,记作 n^{out}.

上述定义可以直观地理解为:当查找节点的前驱(后继)节点时,如果找到的节点仍为控制节点,则继续寻找该节点的前驱(后继)节点,直到不是控制节点为止,例如图 1 模型(a)中 $g6^{in}=\{D\}$.基于上述概念,定义节点间上下文相似度如下:

定义 9(上下文相似度). 给出$(N_1,E_1,T_1,\Omega_1,\alpha_1)$和$(N_2,E_2,T_2,\Omega_2,\alpha_2)$两个 PMG,对于节点 $n_1\in N_1$ 和 $n_2\in N_2$,它们之间的上下文相似度是传递前驱和传递后继节点集合之间匹配程度,记作:

$$SimC(n_1,n_2)=\frac{\sum_{(n_i,n_j)\in M_{SimA}^{opt}(n_1^{in},n_2^{in})}SimA(n_i,n_j)+\sum_{(n_p,n_q)\in M_{SimA}^{opt}(n_1^{out},n_2^{out})}SimA(n_p,n_q)}{\max(|n_1^{in}|,|n_2^{in}|)+\max(|n_1^{out}|,|n_2^{out}|)} \tag{6}$$

其中,$M_{SimA}^{opt}(n_1^{in},n_2^{in})$表示 n_1 和 n_2 的传递前驱节点集合之间的最佳节点对应关系集合.即在 n_1^{in} 和 n_2^{in} 之间选择不重叠的节点配对,使得它们属性相似度之和为最高,其详细定义如下:

定义 10(最佳节点对应关系集合). 给出$(N_1,E_1,T_1,\Omega_1,\alpha_1)$和$(N_2,E_2,T_2,\Omega_2,\alpha_2)$两个 PMG,其中,$S_1\subseteq N_1,S_2\subseteq N_2$ 是两个 PMG 节点的子集,节点 $n_1\in S_1,n_2\in S_2$.S_1 和 S_2 之间的节点对应关系集合是一个 S_1 到 S_2 的局部内射 $S_1\mapsto S_2$,记作 $M(S_1,S_2)$. $M_{SimA}^{opt}(S_1,S_2)$ 是一个最佳节点对应关系集合当且仅当对于任意的节点对应关系集合 $M(S_1,S_2)$满足条件:

$$\sum_{(n_1,n_2)\in M_{SimA}^{opt}(S_1,S_2)}SimA(n_1,n_2)\geqslant\sum_{(n_1,n_2)\in M(S_1,S_2)}SimA(n_1,n_2) \tag{7}$$

基于上述定义的类型相似度、属性相似度和上下文相似度,我们可以综合地给出过程模型之间任意两个节点的相似度计算方法:

定义 11(节点相似度). 给出$(N_1,E_1,T_1,\Omega_1,\alpha_1)$和$(N_2,E_2,T_2,\Omega_2,\alpha_2)$两个 PMG,对于节点 $n_1\in N_1$ 和 $n_2\in N_2$,它们之间的节点相似度记作:

$$SimN(n_1,n_2)=\begin{cases}SimY(n_1,n_2)\cdot SimA(n_1,n_2), & \alpha_1(n_1).TYPE\neq Gateway\\ SimY(n_1,n_2)\cdot SimC(n_1,n_2), & \text{otherwise}\end{cases} \tag{8}$$

3 基于树编辑距离的过程模型相似性度量

本节将基于两个过程模型对应的过程结构树及其之间的节点映射关系来度量过程模型之间的整体匹配程度.树是计算机科学中最常见和广泛研究的组合结构之一,树之间的相互比较问题已经应用于结构化文本数据库、程序分析、编译器优化等多个研究领域中.树编辑距离[13]是一种常见的有序树(ordered tree)之间相似性度量方法,由字符串编辑距离的概念扩展得到.我们将本文使用的树编辑距离的概念定义如下:

定义 12(树编辑距离). 两棵有序树 T_1 和 T_2 之间的编辑距离是将 T_1 转换到 T_2 的所有基本操作的最小开销总和,记作$\delta(T_1,T_2)$.基本操作包括删除和转换现有节点以及插入新节点等 3 种.我们把插入和删除节点 v 的操作开销记作 $c_{del}(v)$,把节点 v_1 转换为节点 v_2 的操作开销记作 $c_{match}(v_1,v_2)$.本文中定义插入或删除节点操作开销 $c_{del}(v)=1$,节点转换操作开销反比于节点之间的匹配程度,具体地:$c_{match}(v_1,v_2)=2\cdot(1-Match(v_1,v_2))$.

由于树 T_1 中的一个删除操作等价于树 T_2 中的一个插入操作,在计算编辑距离时,可以仅考虑在两棵树上进行删除和转换操作来将 T_1 和 T_2 转换成相同的树.目前的树编辑距离的算法都是在有序森林(ordered forest)的基础上进行,而有序树是有序森林的特例.本文使用文献[22]中提出的经典递归算法来计算树编辑距离.我们把空的森林或树记作$\varnothing$,森林 F 的节点集合简单记作 F.对于节点 $v\in F$,以节点 v 为根的子树记作 F_v,删除节点 v 后得到的森林记作 $F-v$.特别地,若 F 为一棵树,将删除 F 的根节点后得到的森林记作 F^-.森林 F 中最左边和最右边的树分别记作 L_F 和 R_F,它们的根节点记作 l_F 和 r_F,则 F 和 G 之间的编辑距离$\delta(F,G)$递归地计算如下:

$$\delta(\varnothing,\varnothing)=0 \tag{9}$$

$$\delta(F,\varnothing)=\delta(F-r_F,\varnothing)+c_{del}(r_F) \tag{10}$$

$$\delta(\varnothing,G)=\delta(\varnothing,G-r_G)+c_{del}(r_G) \tag{11}$$

$$\delta(F,G)=\min\begin{cases}\delta(F-r_F,G)+c_{del}(r_F)\\ \delta(F,G-r_G)+c_{del}(r_G)\\ \delta(R_F^-,R_G^-)+\delta(F-R_F,G-R_G)+c_{match}(r_F,r_G)\end{cases} \tag{12}$$

文献[22]中证明,该算法的复杂度为 $O(n_{height}\cdot m_{height}\cdot n\cdot m)$,其中,$n$ 和 m 分别表示树 F 和 G 的规模,即节点的数量;n_{height} 和 m_{height} 分别表示树 F 和 G 的高度.当我们度量两棵树之间的相似度时,需要对编辑距离进行归一化处理,在两棵树之间不存在任何节点对应关系的极端情况下,他们之间的编辑距离将达到最大值.因此,可以定义树编辑距离相似度如下:

定义 13(树编辑距离相似度). 给定两棵过程结构树 PST_1 和 PST_2,他们之间的树编辑距离相似度为

$$TSim(PST_1, PST_2) = 1 - \frac{\delta(PST_1, PST_2)}{|PST_1| + |PST_2|} \tag{13}$$

4 评 估

本节将使用真实的过程模型对过程变体匹配技术进行评估,评估分为两个部分:第 1 部分的目标是评估模型变体之间元素匹配关系构建的有效性,并寻找匹配关系判别依据中各参数的合适取值;第 2 部分我们在过程模型集合中查找与给定的输入模型相似的模型,通过分析查询结果来评估过程模型整体匹配度量的有效性.

4.1 过程变体元素匹配的评估

使用 BPM AI 过程模型库[14]作为实验评估模型的来源.BPM AI 是由工业界和学术界合作创建的用于科学研究的过程建模平台,合作组织的研究人员和学生使用平台提供的在线建模工具创建了大量描述真实过程的模型.通过设定过滤条件,我们从模型库中共获得 2 183 个使用 BPMN Process 描述的过程模型,其中,修订版本数量多于 5 个的模型 516 个,我们选取每个模型的初始版本及最新版本的组合作为候选过程变体集合,以保证过程变体之间具有一定程度的差异性.为了从这些过程变体集合中更广泛地选取具有代表性的模型,并且避免活动语义不明确的模型,通过设置模型名称关键字来定义实验集的选取范围.最终,从符合条件的 255 组过程变体中随机选择了 25 组模型,去除 2 组具有明显结构错误的模型,最后得到 23 组过程变体组成的实验模型集合**.集合中包含的过程变体分布情况见表 1.

Table 1 Experimental collection of process model variants

表 1 过程模型变体实验集

关键字	模型总量	选取数量	关键字	模型总量	选取数量	关键字	模型总量	选取数量
Account	16	2	Order	56	3	Recruitment	12	2
Bank	6	2	Patient	5	1	Request	32	2
Invoice	62	4	Purchase	13	2	Travel	10	1
Loan	15	1	Registration	7	1	Vendor	21	2

通过使用本文方法自动构建的匹配关系和手工建立的参考匹配关系进行对比分析,来评估方法的有效性.首先将 23 组测试模型以问卷形式分发给课题组内其他 2 名具有过程建模项目经验的研究生,由他们建立每组过程模型之间的元素匹配关系,并为找到的每条复杂匹配关系赋予信任值(取值范围为{1,2,3,4,5},值越高则表示匹配程度越高).本文作者同时也对所有 23 组模型进行手工分析,将分析结果与回收的问卷结果进行对比,通过与问卷者讨论解决不一致意见后,最终建立参考匹配关系 318 条,其中包括原子对应关系 248 条、单对多对应关系 18 条以及多对多对应关系 52 条.

为了寻找复杂对应关系的匹配值计算的 3 个权重参数的合理取值,我们以 0.1 为间隔对 3 个权重值的所有组合进行测试,即,(0.1,0.1,0.8),(0.1,0.2,0.7),…,(0.8,0.1,0.1)等共计 36 种情况,将人工建立的 70 条复杂对应关系的信任值与他们在本文方法中计算得到的匹配值进行线性回归分析,结果表现最好的 4 组权重值组合见表 2,可以看出过程块所包含节点的属性组合而成的文本相似度 ω_2 对过程块的匹配程度起到最重要的影响.

PST 节点映射关系判别依据中的阈值是决定匹配关系是否成立的重要因素,我们通过准确率 P、召回率 R 以及 F 值等指标来评估构建匹配关系的有效性:准确率表示构建的匹配关系中正确的比例;召回率表示构建的匹配关系在参考匹配关系集合中的覆盖率;F 值则是综合考虑准确率和召回率的结果,即,$F=2\cdot(P\cdot R)/(P+R)$.为了得到阈值 $cutoff_n$ 和 $cutoff_p$ 的合适取值,我们以 0.05 为间隔在[0,1]的范围内进行了测试,得到的各项评价指标的结果如图 5 所示.可以看出:设置较低的阈值事实上对准确率的影响并不大,这是因为去除了候选映射关系集合中重叠的匹配关系,即每个节点或过程块都最多只能存在一条匹配关系;但是过高的阈值会导致召回率的迅速下降.如果我们以 F 值为综合评价指标,阈值 $cutoff_n$ 和 $cutoff_p$ 可以分别设置为 0.55 和 0.6.

**为了便于过程模型的读取和解析,使用 ILOG JViews BPMN Modeler 1.0 工具重新建模了 23 组过程模型变体,实验模型集合及其更多的统计信息可以从以下链接获取:https://drive.google.com/file/d/0B2rnkzeFwaxCaVpXRUhnM3VXU1U/edit?usp=sharing.

Table 2 Analysis of weights in criteria of complex matching relations

表 2 复杂对应关系判别条件的权重值分析

ω_1	ω_2	ω_3	R^2
0.2	0.7	0.1	0.774
0.1	0.7	0.2	0.762
0.1	0.8	0.1	0.717
0.3	0.6	0.1	0.695

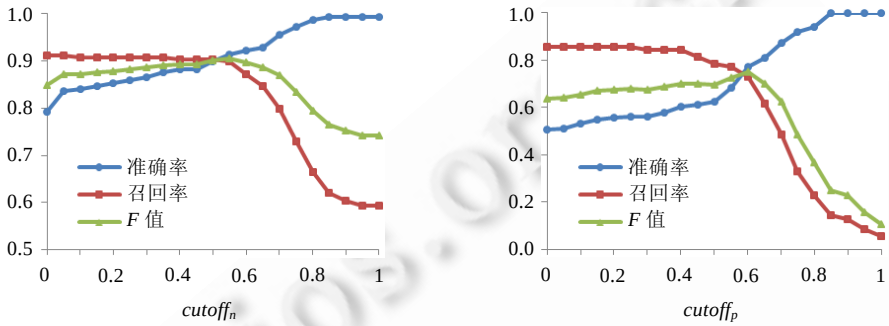


Fig.5 Selecting threshold of elementary and complex matching relations

图 5 原子和复杂匹配关系的阈值选取

通过设定上述实验得到的权重和阈值参数,最终得到如表 3 第 1 行所示的匹配关系构建结果.表 3 中各列指标的数值中均在括号内依次给出了 3 类对应关系的分类统计结果.我们还将其他文献中相关的 3 种方法在本文的实验集的基础上进行了分析,这些方法包括活动标签匹配法[15]、贪心图编辑距离匹配法[6]以及文献[23]中提出的业务-IT 模型映射方法,它们的实验结果同样在表 3 列出.可以看出:方法 1 和方法 2 仅支持原子对应关系的构建,方法 3 能够支持复杂对应关系的建立,但本文方法在原子和复杂匹配关系的构建上都能够表现出较好的效果.通过分析认为原因是:1) 尽管方法 1 采用词法和语义相似度结合的活动名称相似度计算技术,方法 2 利用图编辑距离选取最佳对应关系集合,它们都能够一定程度的提高原子匹配关系构建的准确率,但并没有涉及控制节点之间匹配关系构建的策略,使得原子对应关系查找的召回率较低;2) 尽管方法 3 能够支持复杂对应关系的构建,但它未考虑活动命名的差异,即要求原子对应关系的两个节点具有相同的命名,并且在判别复杂对应关系时过于依赖于原子对应关系的识别结果,使得构建出的复杂对应关系遗漏数量较多.

Table 3 Construction results of matching relations between model elements

表 3 模型元素匹配关系的构建结果

	正确匹配数量	错误匹配数量	遗漏匹配数量	正确率(%)	召回率(%)	*F* 值
0. 本文研究	274(223;10;41)	36(21;6;9)	44(25;8;11)	88(91;63;82)	86(90;56;79)	0.87(0.90;0.59;0.80)
1. 活动标签匹配	181(181;–;–)	9(9;–;–)	137(67;–;–)	95(95;–;–)	57(73;–;–)	0.71(0.83;–;–)
2. 贪心图匹配	167(167;–;–)	6(6;–;–)	151(81;–;–)	97(97;–;–)	53(67;–;–)	0.69(0.79;–;–)
3. 业务-IT 映射	155(125;9;21)	7(0;2;5)	163(123;9;31)	96(100;82;81)	49(50;51;40)	0.65(0.67;0.62;0.54)

通过实验评估发现:尽管本文提出的元素匹配技术相比现有方法在构造复杂对应关系方面具有一定的优势,但仍然存在局限性.由于本方法以过程结构树为载体,因此仅能够支持过程块之间的复杂对应关系的自动构建,导致了一些匹配关系查找的遗漏.此外,为了有效地生成过程结构树,本文假设进行匹配的过程模型是块结构化的,在实施本方法前需要对部分非结构化的模型进行结构化转换处理[17],因此对包含复杂控制流结构的过程模型的支持有待提升.

4.2 过程模型相似性度量的评估

本节对基于过程结构树编辑距离的过程模型全局相似性度量方法进行评估分析.我们使用过程模型搜索

的场景来进行实验,即给定一个输入模型 M 和被查询模型集合 S,在集合 S 中查询与输入 M 相似的过程模型,并将查询结果按照相似程度排序.采用上节实验中使用的 46 个模型为基础,另外还在其相应的 23 组过程变体中随机选择了 14 个中间版本的模型,得到 60 个过程模型组成的被查询模型集合.我们从这 60 个模型的集合中随机选取了 10 个模型作为输入模型.为了使输入模型和被查询模型保持一定的差异,我们对输入模型进行了任意的少量修改操作,包括改变节点名称、删除或添加节点以及改变节点之间的顺序.将每个输入模型与 60 个被查询模型依次进行相似性分析,最终,每个输入模型都得到了按照树编辑距离相似度值由高到低排序的被查询模型的序列.

通过手工在被查询模型集合中为每个输入模型选择与之相似的模型.从直观上看,对于每个输入模型,集合中应该有 2 个或 3 个与之匹配的模型,但实际上存在更多相似的模型,这是因为不同组的过程变体之间也可能存在相似关系.通过人工判断,确认共有 31 对模型之间存在相似关联关系.将本文方法与节点匹配相似度[4]、图编辑距离相似度[24]以及我们前期工作[25]中扩展的图编辑距离相似度等方法进行对比分析.为了统一评估标准,这里的 4 种方法在进行模型节点文本相似性度量时均使用本文定义 7 给出的计算方法.图 6 的结果表示了准确率随召回率的变化曲线,即:按照相似度值由高到低依次判断该组模型是否被人工判定为相似,直到找全所有人工判别的匹配对(即正确解).为了便于展示实验结果,在图中按照每发现 3 个正确解标出一次准确率数据的曲线呈现.可以看出:本文提出的基于树编辑距离的方法在找到超过 80%正确解的情况下仍然保持着较高的准确率,能够较好地适用于过程模型查询的应用场景中.

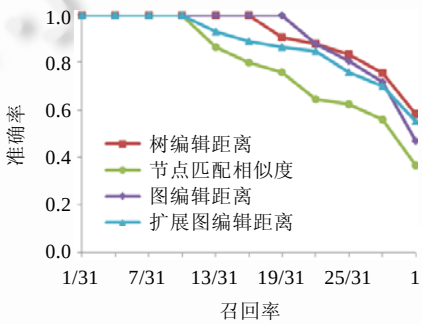


Fig.6 Curve between precision and recall

图 6 准确率-召回率曲线

分析结果看出:基于树编辑距离的过程模型相似性度量方法虽然具有较好的准确率,但相比其他方法优势并不明显.未来可以从以下两个方面来提高方法的有效性和效率:1) 针对不同层次和类型的树编辑操作,研究设定合适的权重值以提高度量的准确性;2) 由于复杂元素匹配关系的构建以及树编辑距离度量过程占用较多的时间开销,可以研究设置高效而相对简单的过滤条件(如文本特征),把一些明显不存在关联的模型组快速排除,避免在这些模型上耗费匹配度量时间,从而提高相似性搜索的效率.

5 相关工作

如何有效地追踪管理模型库中的过程变体,并保持它们之间的一致性演化,是广泛研究的问题.目前,对过程变体的管理技术可以分为两类:1) 通过过程模型合并[26,27]的方式,将相关联的过程变体通过一个可配置过程模型[28,29]进行统一集中地管理;2) 分散管理过程变体,仍然保持它们之间的独立性,但是通过一致性分析[30]、变更传播[31]、版本控制[32]等技术手段,保证过程变体之间的一致性演化.这两类方法有一个相同之处,就是它们都需要识别过程变体之间的共同点和差异性.构建过程变体之间的匹配关系,是它们共同的理论基础.

文献[6]提出并对比了 3 种过程模型元素匹配的方法,包括活动名称词法匹配、贪心图匹配以及 A*图匹配.实验评估结果表明:基于图编辑距离的贪心算法(贪心图匹配)能够表现出最佳的匹配结果,准确率和召回率分

别达到 0.89 和 0.60.但是这 3 种方法都只能支持原子对应关系的识别.

ICoP 过程模型匹配框架[5]提供了一种组件式的体系架构,它将过程模型的元素匹配过程规范化为输入/输出明确的 4 个步骤.对于每个步骤都提供不同的可选组件,通过选择组件来创建出需要的模型匹配器,具有较强的可扩展性.但是这些组件的具体实现算法并没有具体描述.该框架中表现出最佳实验结果的匹配器实例的准确率和召回率分别达到 0.82 和 0.51,能够支持原子对应关系和单对多复杂对应关系的构建.

文献[15]提出了一种优化的活动标签词法匹配的方法,采用单词包(bag-of-words)和活动标签剪枝(label pruning)的思想,在原子对应关系的识别上能够比字符串编辑距离和语义相似度等技术的准确率更高.文献[16]提出了基于语义注释的过程模型匹配方法,并在活动匹配中考虑了活动输入输出数据的影响,但是该方法同样不能支持多对多对应关系,并且没有给出具体的实验评估数据.

文献[23]提出的业务层模型到 IT 层模型映射方法能够支持多对多对应关系的识别,但是没有考虑节点名称之间的相似性,即要求匹配节点具有相同的命名;此外,在判别复杂对应关系是否成立时,该方法采用两个阈值"一刀切"的方法,即所包含节点的原子对应关系覆盖率和综合文本相似度必须同时超过给定阈值,这使得复杂对应关系的建立过于依赖于原子对应关系的识别结果.而本文采用加权平均的判别方法,并考虑了上下文节点相似度的因素.上述方法各自支持的特性在表 4 中进行了总结.

Table 4 Comparison of related work

表 4 相关研究对比:+:支持;–:不支持;+/–:部分支持

	贪心图匹配[6]	ICoP 框架[5]	标签匹配[15]	语义注释[16]	业务-IT 映射[23]	本文方法
考虑节点名称	+	+	+	+	–	+
考虑模型结构	+	+	–	+	+	+
原子对应关系	+	+	+	+	+/–	+
单对多对应关系	–	+	–	+	+	+
多对多对应关系	–	–	–	–	+	+
方法实现和验证	+	+/–	+	–	+	+

本文相关的另一个研究问题是过程模型之间的相似性度量.Dijkman 等人[4]提出并评估了 3 类过程模型相似性度量方法:节点匹配相似度、结构相似度和行为相似度.实验结果表明,基于图编辑距离的结构相似度的表现略优于其他两种度量方法.文献[24]进一步分析和评估了 4 种基于图编辑距离相似度指标的算法,最终,贪心算法能够得到更加准确的度量结果并且效率最佳.文献[7]提出了一种快速过程模型相似性搜索方法,它能够在传统的结构相似性度量之前,通过提取简单的模型特征对模型的相似度进行快速评估,避免了过多不必要的结构分析操作,从而较大幅度地提高了时间效率.本文的前期研究[25]扩展了传统的图编辑距离相似度的计算方法,考虑了模型元素的复杂对应关系对相似度的影响.而本文采用了更直观的树编辑距离的相似性度量方法,从模型结构的另一种角度进行了探索.此外,还有一些研究通过抽取模型的行为特征,基于活动之间的因果关系来度量模型之间的行为相似性[8–10].

6 结束语

本文从模型的微观和宏观两个视角对过程变体匹配技术进行了研究:从微观的角度,提出了基于过程结构树的模型元素匹配关系查找技术,保证了过程变体匹配的效率和有效性;此外,还从宏观的角度进一步研究了如何度量过程模型之间整体的匹配程度,利用已构建好的模型元素匹配关系,提出了基于树编辑距离的过程模型相似性度量方法.利用从 BPM AI 过程模型库中选取的过程变体集合进行了实验评估,结果表明,本文提出的过程变体匹配方法能够有效地支持全部 3 种类型的元素对应关系,并在查全率和查准率指标上表现出了良好的效果,基于树编辑距离的过程相似性度量方法也能够较好地适用于过程模型查询的应用场景中.

在评估过程中发现:仍然存在少量的匹配关系并非由过程块构成,造成了一些匹配关系的遗漏.在未来工作中,我们考虑研究过程块形式以外的其他模型节点组合模式,以提高匹配关系发现的召回率.此外,将对方法进行扩展来支持包含复杂控制流模式(如取消和多实例等)的过程模型.最后,本文实验使用的过程变体源自于修

改模型带来的不同版本,未来将使用多种来源以及更大规模的过程变体集合对本方法作进一步地评估和改进.

References:

[1] Dijkman R, La RM, Reijers HA. Managing large collections of business process models—Current techniques and challenges. Computers in Industry, 2012,63(2):91−97. [doi: 10.1016/j.compind.2011.12.003]

[2] Lu R, Sadiq S, Governatori G. On managing business processes variants. Data & Knowledge Engineering, 2009,68(7):642−664. [doi: 10.1016/j.datak.2009.02.009]

[3] Weidlich M, Mendling J, Weske M. Propagating changes between aligned process models. Journal of Systems and Software, 2012, 85(8):1885−1898. [doi: 10.1016/j.jss.2012.02.044]

[4] Dijkman R, Dumas M, Van Dongen B, Kaarik R, Mendling J. Similarity of business process models: Metrics and evaluation. Information Systems, 2011,36(2):498−516. [doi: 10.1016/j.is.2010.09.006]

[5] Weidlich M, Dijkman R, Mendling J. The ICoP framework: Identification of correspondences between process models. In: Proc. of the 22th Int'l Conf. on Advanced Information Systems Engineering. 2010. 483−498. [doi: 10.1007/978-3-642-13094-6_37]

[6] Dijkman R, Dumas M, García-Banuelos R. Aligning business process models. In: Proc. of the 13th IEEE Int'l Enterprise Distributed Object Computing Conf. 2009. 45−53. [doi: 10.1109/EDOC.2009.11]

[7] Yan ZQ, Dijkman R, Grefen P. Fast business process similarity search. Distributed and Parallel Databases, 2012,30(2):105−144. [doi: 10.1007/s10619-012-7089-z]

[8] Van Dongen B, Dijkman R, Mendling J. Measuring similarity between business process models. In: Proc. of the 20th Int'l Conf. on Advanced Information Systems Engineering. 2008. 450−464. [doi: 10.1007/978-3-540-69534-9_34]

[9] Kunze M, Weidlich M, Weske M. Behavioral similarity—A proper metric. In: Proc. of the 9th Int'l Conf. on Business Process Management. 2011. 166−181. [doi: 10.1007/978-3-642-23059-2_15]

[10] Wang SH, Wen LJ, Wei DS, Wang JM, Yan ZQ. SSDT matrix-based behavioral similarity algorithm for process models. Computer Integrated Manufacturing Systems, 2013,19(8):1822−1831 (in Chinese with English abstract).

[11] Dijkman RM, Van Dongen B, Dumas M, Garcia BL, Kunze M, Leopold H, Mendling J, Uba R, Weidlich M, Weske M, Yan ZQ. A short survey on process model similarity. In: Proc. of the 25 Years of CAiSE: Seminal Contributions to Information Systems Engineering. 2013. 421−427. [doi: 10.1007/978-3-642-36926-1_34]

[12] Vanhatalo J, Volzer H, Leymann F. Faster and more focused control-flow analysis for business process models through SESE decomposition. In: Proc. of the 5th Int'l Conf. on Service-Oriented Computing. 2007. 43−55. [doi: 10.1007/978-3-540-74974-5_4]

[13] Bille P. A survey on tree edit distance and related problems. Theoretical Computer Science, 2005,337(1):217−239. [doi: 10.1016/j.tcs.2004.12.030]

[14] BPM academic initiative. http://bpt.hpi.uni-potsdam.de/BPMAcademicInitiative/

[15] Klinkmüller C, Weber I, Mendling J, Leopold H, Ludwig A. Increasing recall of process model matching by improved activity label matching. In: Proc. of the 11th Int'l Conf. on Business Process Management. 2013. 211−218. [doi: 10.1007/978-3-642-40176-3_17]

[16] Gater A, Grigori D, Bouzeghoub M. Complex mapping discovery for semantic process model alignment. In: Proc. of the 12th Int'l Conf. on Information Integration and Web-Based Applications & Services. 2010. 317−324. [doi: 10.1145/1967486.1967537]

[17] Kiepuszewski B, Hofstede AHM, Bussler CJ. On structured workflow modelling. In: Proc. of the 12th Int'l Conf. on Advanced Information Systems Engineering. 2000. 431−445. [doi: 10.1007/3-540-45140-4_29]

[18] Reijers HA, Mendling J. Modularity in process models: Review and effects. In: Proc. of the 6th Int'l Conf. on Business Process Management. 2008. 20−35. [doi: 10.1007/978-3-540-85758-7_5]

[19] Thom LH, Reichert M, Iochpe C. Activity patterns in process-aware information systems: Basic concepts and empirical evidence. Int'l Journal of Business Process Integration and Management, 2009,4(2):93−110. [doi: 10.1504/IJBPIM.2009.027778]

[20] Johnson R, Pearson D, Pingali K. The program structure tree: Computing control regions in linear time. In: Proc. of the ACM SIGPLAN '94 Conf. on Programming Language Design and Implementation. 1994. 171−185. [doi: 10.1145/178243.178258]

[21] Porter MF. An algorithm for suffix stripping. Program: Electronic Library and Information Systems, 1980,14(3):130−137. [doi: 10.1108/eb046814]

[22] Zhang K, Shasha D. Simple fast algorithms for the editing distance between trees and related problems. SIAM Journal on Computing, 1989,18(6):1245−1262. [doi: 10.1137/0218082]

[23] Branco MC, Troya J, Czarnecki K, Kuster J, Völzer H. Matching business process workflows across abstraction levels. In: Proc. of the 15th Int'l Conf. on Model Driven Engineering Languages and Systems. 2012. 626−641. [doi: 10.1007/978-3-642-33666-9_40]

[24] Dijkman R, Dumas M, Garcia BL. Graph matching algorithms for business process model similarity search. In: Proc. of the 7th Int'l Conf. on Business Process Management. 2009. 48−63. [doi: 10.1007/978-3-642-03848-8_5]

[25] Ling JM, Zhang L, Feng Q. An improved structure-based approach to measure similarity of business process models. In: Proc. of the 26th Int'l Conf. on Software Engineering and Knowledge Engineering. 2014. 377−380.

[26] La Rosa M, Dumas M, Uba R, Dijkman R. Business process model merging: An approach to business process consolidation. ACM Trans. on Software Engineering and Methodology, 2013,22(2):1−42. [doi: 10.1145/2430545.2430547]

[27] Li C, Reichert M, Wombacher A. Mining business process variants: Challenges, scenarios, algorithms. Data & Knowledge Engineering, 2011,70(5):409−434. [doi: 10.1016/j.datak.2011.01.005]

[28] La Rosa M, Dumas M, Hofstede AHM, Mendling J. Configurable multi-perspective business process models. Information Systems, 2011,36(2):313−340. [doi: 10.1016/j.is.2010.07.001]

[29] Rosemann M, Van der Aalst WMP. A configurable reference modelling language. Information Systems, 2007,32(1):1−23. [doi: 10.1016/ j.is.2005.05.003]

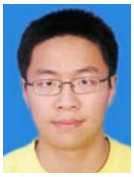
[30] Weidlich M, Mendling J, Weske M. Efficient consistency measurement based on behavioral profiles of process models. IEEE Trans. on Software Engineering, 2011,37(3):410−429. [doi: 10.1109/TSE.2010.96]

[31] Weidlich M, Weske M, Mendling J. Change propagation in process models using behavioral profiles. In: Proc. of the 6th Int'l Conf. on Services Computing. 2009. 33−40. [doi: 10.1109/SCC.2009.58]

[32] Ekanayake CC, La Rosa M, Hofstede AHM, Fauvet MC. Fragment-Based version management for repositories of business process models. In: Proc. of the 10th Int'l Conf. on Cooperative Information Systems. 2011. 20−37. [doi: 10.1007/978-3-642-25109-2_3]

附中文参考文献:

[10] 汪抒浩,闻立杰,魏代森,王建民,闫志强.基于任务最短跟随距离矩阵的流程模型行为相似性算法.计算机集成制造系统,2013,19(8):1822−1831.



凌济民(1989－),男,江西南昌人,博士生,主要研究领域为业务过程管理,业务过程建模与分析,过程模型可变性管理.

张莉(1968－),女,博士,教授,博士生导师,主要研究领域为软件工程,软件体系结构,软件产品线,业务过程建模和优化.