业务流程模型相似度研究综述

简开宇, 史涯晴, 黄松, 许山山, 杨忠举

引用本文

简开宇, 史涯晴, 黄松, 许山山, 杨忠举业务流程模型相似度研究综述[J]. 计算机科学, 2023, 50(6): 338-350.

JIAN Kaiyu, SHI Yaqing, HUANG Song, XU Shanshan, YANG Zhongju. Review on Similarity of Business Process Models [J]. Computer Science, 2023, 50(6): 338-350.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220700061

业务流程模型相似度研究综述

简开宇　史涯晴　黄　松　许山山　杨忠举
陆军工程大学指挥控制工程学院　南京 210007
(757268993@qq.com)

摘　要　随着业务流程模型管理库规模的增大,传统的模型管理方式在效率和准确度方面已经无法达到预期,研究能够提升业务流程模型管理效率的技术成为人们的迫切需求。其中,业务流程模型相似度技术在模型搜索、模型一致性检测等模型管理的相关应用场景中能够有效提升工作的效率和精度,因此,对业务流程模型相似度技术的研究已经逐渐成为模型分析领域的一个研究热点,并取得了许多有价值的研究成果。业务流程模型相似度技术涉及的领域较多,可以向不同的分支方向发展,虽然不同分支的模型相似度技术会有方法之间的类比,但是缺乏系统性的整理和分析。文中从相似度计算方法和应用场景这两个层面对业务流程模型相似度技术进行了分类讨论,将相似度计算方法分为文本相似度、语义相似度、结构相似度、行为相似度和基于人类评估的相似度,并分析了每种计算方法的特点。较为常见的业务流程模型相似度应用场景包括一致性检测、标准化、流程模型搜索和模型重用,文中对基于以上场景的相关研究进行了梳理。最后分析了业务流程模型相似度研究面临的挑战。

关键词:相似度计算方法;业务流程模型相似度应用;结构相似度;流程模型搜索;模型库管理
中图法分类号　TP301.1;TP391.1

Review on Similarity of Business Process Models

JIAN Kaiyu,SHI Yaqing,HUANG Song,XU Shanshan and YANG Zhongju
College of Command and Control Engineering,Army Engineering University of PLA,Nanjing 210007,China

Abstract　With the increase of the scale of business process model management database,traditional model management methods are unable to meet the expectations in terms of efficiency and accuracy,and the technology that can improve the efficiency of business process model management has become an urgent demand. Technology of business process similarity can effectively improve efficiency and accuracy of model analysis in scenarios like model search and consistency judge. Therefore,the research on technology of business process similarity has become a research hotspot in the model analysis field. In recent years,researchers have got many valuable achievements,the technologies of business process similarity have developments in many branches involved in different areas. Although there are comparison of technologies in specific branch,there is a lack of systematic research on technologies of business process model similarity. We analyze the calculations of business process model similarity from these dimensions include text similarity,semantic similarity,structure similarity,behavior similarity and human estimation-based similarity,and summarizes the features of these measurements. We find that the technology of business process model similarity is commonly put into these applications include conformance,standardization,search and reuse,then we analyze the research on these scenarios. At last,the challenges of business process model similarity research are analyzed.

Keywords　Similarity calculation method,Application of business process model similarity,Structure similarity,Process model search,Model library management

1　引言

随着企业业务规则变化越来越频繁,业务流程模型需要根据业务规则的变化进行版本的调整,因此需要频繁地对流程版本进行管理。同时,业务资源配置多样化导致业务流程模型的复杂性也进一步提升,更复杂的业务流程模型促使人们在对流程版本进行管理的同时不得不大幅度增加人力和成本,因此需要一种高效的方法对业务流程模型进行定量分析。在此背景下,业务流程模型的相关研究得以展开。随着研究的发展,业务流程模型相似度在流程版本管理、流程检索、一致性和差异性检测等方面已经有了广泛的应用。

随着业务流程模型相似度技术的不断发展,近年来相关人员做了大量研究并取得了许多有价值的研究成果,但缺少对这些研究成果的归纳总结以及对研究现状的整体了解。为此,本文对业务流程模型相似度算法方面的研究和业务流程模型相似度的应用场景进行归纳总结,主要包括相似度计算和应用场景两个方面。

(1)相似度计算阶段的研究主要是对不同的相似度计算方法进行梳理和总结,针对不同的相似度计算方法整理了近年来与该方法相关的研究进展。

(2)业务流程模型相似度应用场景的研究归纳总结了相似度的常用应用场景。

本文第2节介绍了目前模型相似度领域遇到的问题与挑战;第3节对经典的业务流程模型计算方法进行了梳理和分析,比较了不同方法之间的特点,补充了近年来围绕该方法展开的相关研究;第4节对业务流程模型相似度技术的具体

到稿日期:2022-07-07　返修日期:2022-11-22
通信作者:史涯晴(shiyaqing@aeu.edu.cn)

应用场景进行了整理和分析；最后总结全文并展望未来。

2 问题与挑战

2.1 主要问题

不同的业务流程模型在路由结构、执行语义、资源配置等方面具有复杂性，在对模型进行相似度计算的过程中主要考虑以下两方面的问题：

(1)业务流程模型相似度计算的准确性。

(2)业务流程模型相似度技术为工程应用带来的效率提升。

以上述两方面为出发点分析以下问题。

(1)如何选取业务流程模型相似度计算方法

在不同的场景下，采用同一相似度计算方法计算的准确率不同[1-3]，例如在使用图编辑距离计算相似度过程中，在不同情况下编辑操作的代价不同，从而导致计算结果的准确性难以得到保证。另一方面，不同相似度计算方法的复杂程度不同，根据 Thaler 等[3]的调查发现，相似度度量复杂度从线性到 NP 完全，在内存和时间等方面的资源消耗不同，根据计算目的进行相似度方法的选取能够有效提高计算精度，降低计算的复杂度。

(2)在哪些场景下引入业务流程模型相似度技术能够带来效率的提升

业务流程模型相似度技术在许多工程应用场景中能够带来效率的提升，例如在模型的一致性检测[4]方面，如果模型信息的一致性维护单靠开发者手工处理，一方面会经常造成差错和遗漏，另一方面会导致工作效率低下，尤其是随着业务流程模型不断复杂化和快速迭代更新，会给上述场景带来更大的困难，这时，使用业务流程模型相似度技术科学定量地计算模型间的差异度，能够有效地提升工作效率[5]。通过研究业务流程模型相似度技术带来效率提升的应用场景，使人们了解在何种场景下应用业务流程模型相似度技术来提高工程效率具有指导意义。同时，本文还关注业务流程模型相似度计算在不同场景中的特点以及相应的辅助技术。

2.2 面临挑战

业务流程模型建模方法种类多样，经常会遇到相似的业务流程使用不同的建模方法构建的情况[6-9]，导致相似度无法计算，造成此结果的原因可能是不同建模语言的符号类型不同，也可能是模型之间的语言结构和特征不能相互支持。例如，如果一个相似度量需要 Petri 网的正式执行语义，那么这个度量不能在 EPC 模型中使用。如何解决异构模型相似度计算问题，是业务流程模型相似度研究领域面临的一大挑战。

此外，在基于匹配的相似度计算中，业务流程模型间的元素映射是业务流程模型计算的关键环节[8,10-11]，以往的流程元素映射模型往往忽略了流程间除活动映射以外其他元素的映射关系，进而无法保证结果的准确性[12]。近年来的研究大多进一步考虑了除活动映射以外的元素间的映射关系，例如通过考虑 Petri 网库所的上下文环境对库所进行映射，也有研究增加了对角色关系的映射。这些方法在特定环境下能够提升结果的准确率，但都有一定的局限性。因此，如何根据业务场景考虑重要的元素映射关系，仍然是确保结果准确率面临的一大挑战。

3 业务流程模型相似度计算

文本相似度、结构相似度、行为相似度是业务流程模型相似度计算中常见的 3 类方法，而基于人类评估的相似度计算在工程应用中更加广泛，作用通常是对相似度结果进行修正。4 种方法的具体对比如表 1 所列。

表 1 业务流程模型相似度计算方法分析

Table 1 Analysis of business process model similarity calculation methods

类型	计算类型	计算方法	基本思想	方法特性	参考文献
文本相似度	字符串匹配	编辑距离	根据字符串转化的编辑次数来衡量相似度	匹配方法简单，但是仅根据字符串编辑距离进行相似度计算，会因字符串前缀后缀的影响而难以匹配，计算出来的相似度低；另一方面，重要信息和次要信息的匹配没有区分，鉴于大多数相似度计算方法都没有区分主次信息的操作，因此此处不用考虑这一弊端，将其放在文本标签的预处理阶段进行讨论	[15-18]
		Dice 系数	根据字符串的相同部分来衡量相似度，无字符顺序的区分	匹配方法简单，但是没有区分字符串顺序，并且没有考虑语义的相似度，对于含近义词的长句匹配出来的结果准确率可能较低，但是在短句且句式规范的情况下，匹配出的较高相似度具有一定参考意义	[18-19]
		Jaccard 系数	根据字符串的相同部分和相异部分来衡量相似度，无字符顺序的区分	与 Dice 系数的特性较为相似，适用于句式规范的短句	[20-21]
		最长公共子串	考虑字符串的最长公共子序列来衡量相似度	如果句子存在多个长度相近的公共子序列，那么只考虑最长公共子序列计算出的相似度结果是不准确的	[22-23]
	字符串匹配（考虑匹配顺序）	Jaro-Winkler 相似度	根据匹配字符数和换回数目来计算文本相似度	考虑了字符的顺序，字符串的匹配仍是简单匹配，但是在匹配顺序上引入了匹配限制，匹配限制使得增加的冗余的短词对相似度计算的影响程度变小	[24]
	基于拓扑结构	基于拓扑结构的相似度计算	基于语义信息网络等知识拓扑结构，通过计算拓扑结构中的概念、词语所在的边或者节点之间的关联性来确定它们的相关度	基于拓扑结构相似度的计算中，计算概念之间的相关度通常用概率出现在实例中的概率来计算，计算词语的相关度通常使用词语所在边或节点之间的距离来计算	[25-29]
	基于统计的相似度计算	Normalized Google Distance，Pointwise Mutual Information	搜索关键词，根据搜索结果中共现的词数来衡量相似度	基于统计的相似度操作比较复杂，需要根据搜索结果的共现词数进行相似度计算，但是由于搜索结果的数据量较大，且维度众多，因此能够更有效地对相似度进行计算	[30-32]
	词袋相似性	余弦相似度	根据词频为句子建立单词向量，使用余弦相似度进行计算	计算词袋相似度需要对字符串进行分词和分句，然后构建语料库，对语料库中的单词和标点建立数字映射	[28,33-39]

(续表)

类型	计算类型	计算方法	基本思想	方法特性	参考文献
图结构相似度	基于公共节点和边的相似度	Minor 等提出的基于图结构的相似性度量	通过计算两个过程模型的公共节点和边的个数来计算模型相似度	Minor 等提出的方法只考虑了公共节点和公共边的数目,并没有考虑模型中的相似节点或等价节点	[5]
	基于树编辑距离或图编辑距离	树编辑距离	类似于字符串编辑距离,是基于将一个过程模型转化为另一个过程模型所需操作步骤数的度量方法	树编辑/图编辑距离计算过程中,过程模型转化方法的选取、代价函数的定义、编辑距离的求解等会影响图相似性比较的最终结果,基于图编辑距离算法的结果差异较大	[40-43]
	基于上下文的结构信息	基于特征的相似度估计	通过参数对节点进行描述,然后计算节点相似度	节点特征的选取和描述对相似度结果影响很大,可以考虑多个维度的数据进行节点相似度计算,方法较为灵活,但需要根据特定的应用场景设计参数,技术难度较高	[1,44]
行为相似度	痕量计算	基于轨迹的最长公共子序列	在模拟运行或实际流程执行期间生成执行轨迹,利用轨迹的最长公共子序列反映相似性的方法,通过跟踪两个过程模型来量化它们的相似性	这一算法在计算结果上没有给出一个统一的相似值,而是给出了两个组成部分,这两个组成部分表示一个模型的轨迹在多大程度上反映了另一个模型的轨迹	[45-46]
	足迹向量	基于因果足迹的相似度计算方法	根据过程模型的因果足迹生成足迹向量,通过余弦相似度来求解两个模型之间的相似度	在基于因果足迹向量的相似度计算基础上,提出了额外的语义相似度量来量化活动的相似性	[6,47-48]
	行为特征相似度	Weidlich 等提出的基于因果行为关系和预定义的活动映射的相似度量	将流程模型的行为概念化为一组执行跟踪的活动之间的依赖关系,将依赖关系用 4 种不同的关系集表示,在此基础上计算模型的行为相似度	此方法利用流程模型活动产生的执行日志来计算相似度,处理复杂节点匹配的能力更强	[41,47-49]
基于人类评估的相似度	众测相似度评估	用户反馈,相似度评定	人们根据自己的个人知识,主观地量化过程模型之间的相似性	基于人类评估的相似度的研究内容有:使用用户反馈的输入来改进流程模型元素的匹配、基于人群的匹配确定、通过比较与流程模型相关的标签确定相似值等。在数据量较大的场景下,使用人工的成本非常高,人类评估的模型相似度常常作为辅助决策,使得相似度结果在原基础上有一个修正	[50-52]

文本相似度计算在确定标签节点映射的过程中十分常用,其计算复杂度较低,常用的方法有编辑距离、公共子串的计算等。由于文本具有语义,因此通常情况下还需要考虑语义相似度,在文本相似度的基础上考虑同义词、相近概念等语义,这种方法通常被作为一种提高相似度准确率的改进手段使用[13],常见的语义相似度计算方法有基于拓扑结果的方法、基于统计的方法和基于词袋相似性的方法。常见的结构相似度计算有基于边和节点映射的方法、基于编辑距离的方法和基于上下文信息的方法[14]。行为相似度通常利用从活动日志中提取的行为轨迹来计算,常见的方法有基于痕量计算的方法、基于足迹向量的方法和基于行为特征的方法。

3.1 文本相似度

文本相似度主要处理业务流程模型中和自然语言相关的文本标签,自然语言在业务流程模型相似度方面的研究是非常重要的。自然语言形成标签标记模型中包含的元素,这些标签通过句法和语义方面的分析,形成的结果是流程模型相似度度量的重要来源之一[53]。

3.1.1 基于字符串

基于字符串的方法从字符串的匹配出发,把字符串共现和重复程度作为相似度的衡量标准。基于字符串的方法根据是否考虑字符串顺序可以分为两类:一类是不考虑字符顺序的方法,这一类文本相似度计算方法较为常见的有编辑距离、Dice 系数、欧氏距离等;另一类认为字符顺序相同也是字符串相似的重要因素,这一类比较常见的算法有最长公共子序列算法、Jaro-Winkler 算法等。接下来对这些常见方法进行详细介绍。

(1)编辑距离

Levenshtein Distance 由俄罗斯科学家 Levenshtein 于 1965 年提出,在信息论、语言学和计算机科学领域,其被用来度量两个序列相似程度的指标。编辑距离指给定两个标签 l_1,l_2,字符串编辑距离定义为 l_1 转换为 l_2 的字符编辑操作的最小数目。它支持插入、删除和替换。在 Levenshtein 的定义中,所有这些操作的权重都为 1,这些操作也可以由其他非负值加权。编辑距离通常可以转换为相似值。在这种情况下,相似值的计算式为:

$$sim(l_1,l_2)=\frac{dist(l_1,l_2)}{\max(len(l_1),len(l_2))}$$

在计算模型相似度的场景中,Cayoglu 等[57]认为字符串编辑距离计算相似度可以通过删除停止词来增强。此外,字符串编辑距离的经典应用场景还有 DNA 分析、拼字检查、语音辨识和抄袭侦测等。

(2)Dice 系数

给出字符串 l_1 和 l_2,Dice 系数计算两个字符串相似度的公式如下:

$$Dice(l_1,l_2)=\frac{2\times common(l_1,l_2)}{len(l_1)+len(l_2)}$$

其中,$common(l_1,l_2)$是 l_1 和 l_2 中相同字符的个数,$len(l_1)$和 $len(l_2)$是字符串 l_1 和 l_2 的长度。

Dice 系数只关心字符串的相同部分,当不同的字符串存在语义相似性时,Dice 系数计算结果的准确率会降低,使用词向量的形式来计算可以一定程度地规避这个问题。Ji 等[54]基于 Dice 系数进行了向量匹配以构建优化集,然后结合回溯思想和弱选择思想剔除相似性较小的原子,

进一步提高了优化集的生成质量。

(3)Jaccard 系数

Jaccard[21]于 1912 年提出杰卡德相似性度量方法并将其用于分析高山区的区系分布。Jaccard 系数定义为 l_1 与 l_2 交集的大小与 l_1 与 l_2 并集的大小的比值，其不考虑个体间具体差异值的大小，仅关注个体间是否存在相同的特征，在使用 Jaccard 系数进行计算时，给出字符串集合 l_1 和 l_2，计算式如下：

$$J(l_1,l_2)=\frac{|l_1\bigcap l_2|}{|l_1\bigcup l_2|}=\frac{|l_1\bigcap l_2|}{|l_1|+|l_2|-|l_1\bigcap l_2|}$$

基于 Jaccard 系数可以成对评估数据的相似性和多样性，Huang 等[17]在对数据集相似度方法评估时使用了 5 种最广泛的相似性度量，结果显示 Jaccard 相似度在数据评估方面效果最好；Niwattanakul 等[21]将 Jaccard 相似度用于比较关键词之间的相似性，并在关键词聚类上取得了较好的结果。

(4)最长公共子串

最长公共子串计算相似度考虑了两个字符串的字符顺序，在字符串中找到最长的公共子串，基于公共子串计算相似度计算并没有一个确定的公式，可以根据计算目标进行设计。本文给出的计算公式是基于公共子串的相似度计算方法之一，给定两个字符串集 l_1 和 l_2，计算式如下：

$$Similarity(l_1,l_2)=\frac{LCS(l_1,l_2)}{\max(len(l_1),len(l_2))}$$

其中，LCS 是 l_1 和 l_2 的最长公共子串，len 是字符串长度，max 是求最大字符串长度的函数。

最长公共子串考虑了字符顺序，计算出的相似度偏低，在文本相似度计算结果偏高时，可以考虑使用最长公共子串平衡计算结果。

(5)Jaro-Winkler 相似度

Jaro 相似度是由 Matthew A. Jaro 于 1989 年提出的算法，Jaro-Winkler distance[21]是由 William E. Winkler 在 Jaro distance 的基础上进一步改进的算法。Jaro-Winkler 相似度能够很好地匹配个人和实体名称[19]，因此被广泛应用于信息提取、记录链接、实体链接等领域。对于两个字符串 l_1 和 l_2，它们的 Jaro-Winkler 相似度计算式如下：

$$Sim_j=\begin{cases}0, & \text{if } m=0\\ \frac{1}{3}\left(\frac{m}{|l_1|}+\frac{m}{|l_2|}+\frac{m-t}{m}\right), & \text{otherwise}\end{cases}$$

其中，$|l_1|$ 和 $|l_2|$ 表示字符串 l_1 和 l_2 的长度，m 表示两字符串的匹配字符数，t 表示换位数目的一半。两个字符串越相似，Jaro-Winkler 的计算结果越大。

3.1.2 基于单词

基于词层面特征的相似度计算的常见方法有基于词的编辑距离、词的统计特征和词汇的语义特征等。基于词的编辑距离的方法类似于字符串编辑距离，重点是利用度量节点的相似性来完成对应的关系映射。基于词的统计特征的方法通过考虑词频、词性等信息来度量句子间的相似度。基于词汇语义特征的相似度计算方法利用了自然语言处理方面的技术进行相似度计算。例如使用同义词库来计算具有相似语义的词之间的相似度；还有通过识别句子结构，从而获取动作、操作对象、标签中的可选片段等，使这部分内容参与匹配的方法，能够更精确地计算相似度。

(1)基于单词编辑距离

单词编辑距离类似于字符串编辑距离，首先，对选取的流程模型的所有节点的标签进行标记，提取标签中的每个单词。例如："技术上检查和文档发票"被分为"检查""和""文档""发票"和"技术上"，然后根据单词编辑操作计算距离，与字符串编辑距离的情况一样，重点是利用度量节点的相似性来完成对应关系的映射。这些对应关系是随后量化过程模型相似度的基础。

单词编辑距离与字符串编辑距离的区别在于对长句的分割，因此不同的度量节点意味着不同的映射关系，不同的映射关系对最终计算结果的影响是不同的。

在处理长文本标签时，字符串编辑距离计算的相似度往往很低，但长文本标签中有部分片段相似度很高，这需要纳入相似度度量的考虑范围，此时使用基于字符串分割的单词编辑距离可以找到相似映射。另一方面，在计算单词编辑距离时，将单个单词缩减到其词干是常见的做法，被称为特定单词形式的抽象。

(2)基于词袋模型

词袋模型是一个在自然语言处理和信息检索下被简化的表达模型，它能够把一个句子转化为向量表示，然后通过向量计算两个句子间的相似性。这是一种比较简单直接的方法，它不考虑句子中单词的顺序，只考虑此表中单词在这个句子中出现的次数。该方法只考虑了句子的表层信息，并未考虑句子结构、语义等深入特征，只有当语料有一定的规模时这种统计的效果才能体现出来，有一定的局限性。

基于词袋模型的相似度计算方法通常通过附加的 nlp 技术得到增强。Schoknecht 等[55]描述了一个例子，它依赖于基于虚拟文档的流程模型的向量空间表示。与上述技术相比，这种方法不需要特定的节点匹配。

基于词袋模型的相似度计算的应用场景之一是信息检索，信息检索的关键是根据所使用的术语确定文档与输入文档之间的相似度。在基于假设使用相同术语的过程模型描述相似性的过程的前提下，在业务流程模型的上下文中，列出所有标签中包含的所有单词，根据它们出现的次数进行加权，在加权情况下，一个词包含在流程模型中的频率越高，其相关性程度越大。据此量化两个流程模型的相关性。

3.1.3 语义相似度

语义相似度指自然语言标签的相似度量化，考虑标签的意义进行相似度计算[56]，这样的相似度计算通常需要考虑标签中同义词和同音异词的使用[57]。例如，当考虑标签"发送发票"和"转移发票"时，两者都向客户发送发票，虽然字符串编辑和单词编辑距离很大，但是语义相似度度量可以检测同义词，因此可以为标签分配高相似度值[58]。下面介绍语义相似度计算的分类及典型方法。

(1)基于拓扑相似

基于拓扑相似通常借助于构建本体或者语义信息网络等知识拓扑结构，然后通过计算拓扑结构中的概念、词语所在的边或者节点之间的关联性来确定它们的相关度[59-60]。这种

相似度计算方法是基于知识的。最常用的知识结构是 WordNet、HowNet、同义词词林等词汇语义词典。下面介绍基于拓扑相似的相似度计算方法。

1)基于最短节点计数距离[25]

$$Sim_{lch}=-\log\frac{len(C_1,C_2)}{2*\max(depth(C_1),depth(C_2))}$$

其中,len 指两个概念之间最短节点计数距离,$depth$ 指概念在知识结构中的最大深度,max 是最大值函数。

2)基于知识结构深度[26]

$$Sim_{wup}=\frac{2*depth(LCS)}{depth(C_1)+depth(C_2)}$$

其中,LCS 指 Least Common SuperConcept,$depth$ 指概念在知识结构中的最大深度。

3)基于概念的实例概率[27]

$$IC(c)=-\log P(c)$$

$$Sim_{res}=IC(LCS)$$

其中,IC 指 Information Content,$P(c)$指在知识拓扑中出现概念 c 的实例的概率,LCS 指 Least Common SuperConcept。

4)Lin[28]提出的基于 Resnik 方法的语义相似度计算方法

$$Sim_{lin}=\frac{2*IC(LCS)}{IC(c_1)+IC(c_2)}$$

5)后续研究中提出的基于概念的实例概率算法[29]

$$Sim_{jc}=\frac{1}{IC(c_1)+IC(c_2)-2*IC(LCS)}$$

(2)基于统计相似

基于统计的相似计算需要一个语料库,以下是几种典型算法。

1)Normalized Google Distance(NGD)[30]

NGD 基于 Google 搜索关键词返回的 hits 数,两个关键词在搜索结果共现的词数越多,语义上越有可能相似。词语 x 和 y 的 NGD 距离计算式如下:

$$Sim_{NGD}(x,y)=\frac{\max\{\log f(x),\log f(y)\}-\log f(x,y)}{\log N-\min\{\log f(x),\log f(y)\}}$$

其中,N 为 Google 索引的页面数乘以平均每页的可搜索的条数,f 为搜索某一关键词返回的 hits 数。

2)Pointwise Mutual Information(PMI)[31]

与 NGD 类似,该算法的计算结果也与 w_1 和 w_2 在一个大的语料库中的共现词数有关。计算式如下:

$$Sim_{PMI}(w_1,w_2)=\log_2\frac{p(w_1\&w_2)}{p(w_1)*p(w_2)}$$

其中,p 为出现某个词的概率。

3)Latent semantic analysis[32](LSA)

LSA 算法核心为通过奇异矩阵分解(SVD)将文本分解为词汇、文档的特征矩阵,从而可以进一步将其应用到语义相似、搜索等方面。

(3)基于词汇语义

语义相似度方面的研究发展也较快,Li 等[33]提出了句子的相似度度量,Gacitua-Decar 等[34]在研究中使用了该度量。由于语义相似度的目的是考虑标签的意义,因此可以考虑标签的语法,这需要运用自然语言处理领域的技术,如部分词性标注技术和自然语言解析技术。部分词性标注技术将词性作为动词、名词或标点符号分配给标签的每个标记。自然语言解析技术旨在推导标签的语法结构,以识别标签的主语、谓语和宾语[61]。Leopold 等[35]描述的匹配方法中,使用该技术来确定动作、业务对象和标签中的可选片段以计算匹配[62]。由于语法结构和单词形式各不相同,这些技术针对不同语言的具体要求也不同,这就是不同的语言存在不同的词干算法的原因。可应用的自然语言处理领域的相关技术还有词干分析、基于 WordNet 的同义词查找[25,36]、Lin[28]描述的通过相似性度量来确定术语之间的亲密度等。

Li[37]基于 HowNet 和同义词词林提出了语句相关度的定量计算模型,能够挖掘语义信息相同的词语,但该方法比较依赖语义词典的完整性。

语义相似度是衡量模型相似度的一项重要指标,得到了广泛的应用,Antunes 等[38]和 Cayoglu 等[39]描述的业务流程模型相似度度量方法应用了自然语言处理(NLP)领域的技术,考虑到了节点标签的同义词、同音异词、反义语等语义信息。Antunes 等使用的匹配方法配置了额外的语言数据库,如 wiktionary4[63]和 Lin[28]描述的单词的相似度度量。

3.2 图结构相似度

图结构相似度的相关基础知识源于图论,可以将相关的相似度量大致分为基于图结构的相似度量和基于业务流程感知的控制流相似度度量[64]。一般情况下,基于图结构的模型之间的相似性可以量化,例如求两个模型的最大公共子图大小[65](根据节点和边的数量)。一般的基于图的算法不考虑任何连接器。当实际情况中有连接器时,往往会忽略它或通过扩展现有度量来处理。两个过程模型的相似度可以根据模型中控制流连接器的位置和种类进行计算。例如,根据活动顺序和连接器顺序计算两个模型的相似度。

在结构相似度的度量方面,需要考虑的有相似节点和边之间的关系以及图之间相互转化的能力[66-67],Li 等[40]认为,在过程感知信息系统中,图同构和子图同构的方法不能检查模型中的语法问题,比如无法区分 AND-Splite 节点和 XOR-Splite 节点,这样的缺陷在利用编辑距离计算相似度时同样存在,究其原因,在于对模型的简化处理导致的语义信息丢失,Li 等的研究通过增量算法来衡量两种模型间的差异,将包含语义信息的属性附给边和节点,来保证模型相似度计算的准确性。

结构相似度基于抽象化流程模型的拓扑结构,其中可能考虑到文本相似度。图同构[5,42]和图编辑距离[41,15]是衡量结构相似性的常用方法,但是这些度量通常只检查边缘和节点,捕捉语义信息。Li 等通过高级变更操作来度量流程模型[40]之间的距离。

La Rosa 等[68]在他们的流程合并方法中使用了上下文相似性的概念,在该方法中,使用连接器的前后活动来确定连接器间的相似性。

利用图结构计算相似度的相关技术有:计算图编辑距离、计算过程模型的图结构之间的图同构、使用特殊类图来确定模型间的相似性等。

3.2.1 基于公共节点和边的相似度

Minor 等[4]提出的一种基于图结构的相似性度量技术,通过计算两个过程模型的公共节点和边的数目,将它们与

整体节点和边的数目关联起来。如果边是通过开始节点和结束节点定义的，由于流程模型的边指定了它的控制流，因此它们涵盖了图结构方面，那么可以只考虑边。

模型 M_1 的节点集用 N_1 表示，边集用 E_1 表示。模型 M_2 的节点集用 N_2 表示，边集用 E_2 表示。边基于公共节点和边的相似度计算式如下：

$$Sim(M_1,M_2)=\frac{2*(|N_1\cap N_2|+|E_1\cap E_2|)}{|N_1|+|N_2|+|E_1|+|E_2|}$$

Minor 等提出的基于图结构的相似性度量技术只考虑了公共节点和公共边的数目，但实际上，往往存在一些节点并不是公共节点，却能在另一个模型中找到一个节点与其高度相似的情况，在这种情况下，该方法计算出的相似度就会偏低。因此可以考虑将节点间的相似度纳入计算范围，对该方法进行优化，使得计算结果更为准确。

由于几乎所有的业务流程模型都可以抽象成由节点和边组成的网络，因此基于公共节点和边的相似度计算方法具有较强的适用性。

3.2.2 树编辑距离和图编辑距离

树编辑距离相似度计算方法是将业务流程模型用树状结构表示，树的根节点下有两个孩子节点表格头(Thead)和表格体(Tbody)，表格头和表格体的孩子节点是表格行，树的叶子节点是单元格，每个叶子节点包含 3 种属性：行跨度(Rowspan)、列跨度(Colspan)、单元格内容(Content)。采用树的编辑距离来度量两棵树之间的相似度，类似于上述字符串和单词编辑距离，这是基于将一个过程模型转化为另一个过程模型所需操作步骤数的度量方法，相似度值可以通过将距离除以最大距离来计算，计算式如下：

$$Sim(Ta,Tb)=1-\frac{EditDist(Ta,Tb)}{\max(|Ta|,|Tb|)}$$

其中，Ta 和 Tb 表示两个树模型，$EditDist$ 表示两个模型间转化所需的最少操作步骤，max 表示生成树所需的最少操作步骤。

Li 等[40]对于图之间的转换进行了进一步的研究，其将边缘和节点的变更视作原始变更，通过对原始变更过程的分析，发现存在以下 3 个问题：

(1)发现原始变更操作可靠性不强，例如删除一条边便不能保证模型的完整性。

(2)原始变更需要的操作步骤繁琐。

(3)原始变更操作无法准确地确定模型之间的差异。

因此，Li 等提出了高层次的变更操作，高级变更提供了比原始变更更加丰富的语义含义，高级变更操作建立在一组原始变更的基础上，这些原始变更共同表示对流程模型的复杂修改。

图编辑距离也是基于编辑距离的相似度计算方法，计算方法与树编辑距离类似。

Kunze 等[41]将流程模型之间的距离度量问题转化为图匹配问题。在考虑结构相似度的同时，需要进一步结合行为相似度对模型相似度进行综合评定，否则在一定程度上可能会导致相似度值的准确度较低，在 Kunze 等的研究中对此有详细说明。两个进程在结构和使用的标签方面可能非常相似，但他们的行为可能非常不同[42]。

Xu 等[43]在对图编辑距离的研究中探讨了图编辑距离的两个研究重点，也是图编辑距离算法进一步发展的可能方向，表述如下：

(1)代价函数的定义

代价函数定义的优劣直接决定了图相似性比较的最终结果。目前有关定义代价函数的方法具有一定的局限性，或者有一定的约束要求。随着深度学习与机器学习的不断发展，如何构建更加高效的代价模型值得关注。

(2)编辑距离的求解

模式匹配存在精确和非精确匹配，现有大量匹配算法致力于降低精确匹配的复杂度，提高非精确匹配的精确性。但如何更好地表示原图和目标图之间的相关性，尤其是图中边的相关性，值得进一步探究。

图编辑距离常常应用在图模式匹配技术中，基于图编辑距离的匹配方法能够处理多类型的图数据，图编辑距离作为度量图相似性的一种有效且灵活的方法，在模式识别和图像检索领域有着广泛的应用。

3.2.3 基于功能的相似度

量化流程模型的相似性的另一种技术是在节点上下文中使用与结构有关的信息。Yan 等[1]的基于特征的相似度估计就是该技术的一个应用，它使用 5 个参数来描述 1 个节点：$R=\{satrt,stop,split,join,regular\}$，$|\cdot n|$ 为 n 的前面节点的个数，$|n\cdot|$ 为 n 的后面节点的个数，对于每个 $node_n\in N$，由函数：$n\to P(R)$ 分配角色，如下所示：

$$start\in roles(n)\Leftrightarrow|\cdot n|=0$$

$$stop\in roles(n)\Leftrightarrow|n\cdot|=0$$

$$split\in roles(n)\Leftrightarrow|n\cdot|\geqslant 2$$

$$join\in roles(n)\Leftrightarrow|\cdot n|\geqslant 2$$

$$regular\in roles(n)\Leftrightarrow|\cdot n|=1\wedge|n\cdot|=1$$

对于每一组节点 (n_1,n_2)，其中 $n_1\in N_1$，$n_2\in N_2$，N_1 和 N_2 分别为模型 M_1 和 M_2 的节点集，n_1 和 n_2 分别为节点集 N_1 和 N_2 中的一个节点，约定集合 $croles=roles(n_1)\cap roles(n_2)$，角色特征相似度的定义如下：

$$rsim(n_1,n_2)=\begin{cases}1, & \text{if } start\in croles\wedge stop\in croles\\ 1-\dfrac{\|n_1\cdot|-\|n_2\cdot\|}{2(\|n_1\cdot|+|n_2\cdot\|)}, & \text{if } start\in croles\wedge stop\notin croles\\ 1-\dfrac{\|\cdot n_1|-\|\cdot n_2\|}{2(\|\cdot n_1|+|\cdot n_2\|)}, & \text{if } start\notin croles\wedge stop\in croles\\ 1-\dfrac{\|n_1\cdot|-\|n_2\cdot\|}{2(\|n_1\cdot|+|n_2\cdot\|)}-\dfrac{\|\cdot n_1|-\|\cdot n_2\|}{2(\|\cdot n_1|+|\cdot n_2\|)}, & \text{if otherwise}\end{cases}$$

将两个节点 (n_1,n_2) 视为字符串相似度(Levenshtein Distance)和角色特征相似度超过某一阈值的等价节点。最后，定义两个模型 M_1 和 M_2 之间的相似度的是与两个模型节点数相关的对应节点数。相似度计算式如下：

$$Sim(M_1,M_2)=\frac{2|N_1\cap N_2|}{|N_1|+|N_2|}$$

除了这种具体的取样技术外，根据 Melcher 等[44]的描述，根据业务流程模型度量值进行基于结构的相似性量化也是可能的。将度量标准的值作为描述流程模型的向量特征，然后对向量特征进行比较。可以使用特定的通用距离度量分别计算距离和相似值，如 Hamming 距离、Euclidean 距离或 Jaccard 距离等。

3.3 行为相似度

行为相似度计算侧重于进程的执行轨迹。执行轨迹可以通过模拟运行或在流程实际执行期间生成，执行轨迹通常存储在日志中供进一步分析[69]。在相似性度量中，可以通过日志中相同执行序列的数量来确定两个模型之间的相似性[70]，因此考虑了可能执行序列的特征，如最长公共子序列的长度、因果足迹[6]。

关于文本相似度和结构相似度的研究已经较为深入，但在一些情况下，仅仅考虑文本相似度和结构相似度无法得出准确的计算结果[71]。此时，因为两个流程在考虑活动标签和流程结构时，它们可能看起来非常相似，但是行为却非常不同[42]。针对此问题，研究者们提出了许多行为相似度度量方法。Xing 等[47]针对如何从流程模型领域准确地衡量 BPEL (Business Process Execution Language)流程间的相似性提出了解决方案，即利用行为度量来量化 BPEL 流程间的相似性。常规的相似性度量在确定 BPEL 的相似性方面存在一些缺陷，造成这些缺陷有以下两方面的原因：一方面，BPRL 流程有很多详细信息，如数据信息、相关连接等，这些信息在流程模型的抽象过程中缺失了；另一方面，这些测度中大部分都不能满足三角不等式原理[33-34]，这意味着每次模型搜索必须进行穷举搜索，导致对大型 BPEL 存储库的管理、维护和检索的效率非常低。行为度量的引入提高了相似度搜索效率。

3.3.1 轨迹的最长公共子序列

Gerke 等[45]提出了通过轨迹的最长公共子序列反映相似性的方法。有 M_1 和 M_2 两个模型，其中 σ_1 是 M_1 的轨迹序列；σ_2 是 M_2 的轨迹序列；$cd_{\mathrm{Trace}}(\sigma_1,\sigma_2)$是轨迹的遵从度，表示一个过程对活动顺序的遵从程度；$md_{\mathrm{Trace}}(\sigma_1,\sigma_2)$是轨迹的成熟度，表示活动在其他模型中的召回程度。具体的定义如下：

$$cd_{\mathrm{Trace}}=\frac{len(\max(LCS(\sigma_1,\sigma_2)))}{len(\sigma_2)}$$

$$md_{\mathrm{Trace}}=\frac{len(\max(LCS(\sigma_1,\sigma_2)))}{len(\sigma_1)}$$

基于轨迹的遵从度和成熟度，将两模型间的遵从度和成熟度定义为最大轨迹遵从度和最大轨迹成熟度的总和：

$$cd(M_1,M_2)=\frac{\sum_{\sigma_2\in\sum M_2}\max_{\sigma_1\in\sum^{M_1}}(cd_{\mathrm{trace}}(\sigma_1,\sigma_2))}{|\sum M_2|}$$

$$md(M_1,M_2)=\frac{\sum_{\sigma_1\in\sum M_1}\max_{\sigma_2\in\sum M_2}(md_{\mathrm{trace}}(\sigma_1,\sigma_2))}{|\sum M_1|}$$

该算法在计算结果上没有给出一个统一的相似值，而是给出了两个分量，它们反应了一个模型的活动轨迹与另一个模型轨迹之间的距离，这样的表示不利于对模型相似度值进行统计与比较，通常用来辅助提高相似度计算的精度，或者通过参数的描述确定模型的匹配范围，提高模型搜索的效率[46]。

3.3.2 因果足迹相似性

一个模型 M 和它的活动集的因果足迹是一个(Lk_{lb}，Lk_{la})形式的元组[52]，其中 Lk_{lb} 是后向链接，$Lk_{lb}\subseteq(P(F)\times F)$；$Lk_{la}$ 是前向链接，$Lk_{la}\subseteq(F\times P(F))$。一个元组形为($\Theta$，$f$)满足 $\Theta\in P(F)$且 $f\in F$，如果 F 的每个追踪出现 f 之前出现 θ，即 $\sigma_i=\langle\cdots,f,\cdots,\theta,\cdots\rangle$，其中 θ 属于 Θ，那么元组(Θ，f)属于 Lk_{lb}。类似地，一个元组形为(Θ，f)满足 $\Theta\in\Theta\in P(F)$且 $f\in F$，如果 F 的每个追踪出现 f 之后出现 θ，即 $\sigma_i=\langle\cdots,f,\cdots,\theta,\cdots\rangle$，其中 θ 属于 Θ，那么元组(Θ，f)属于 Lk_{la}。为了计算两个模型的相似度，把因果足迹作为索引词向量。两个模型的 $M_1=(N_1,A_1)$和 $M_2=(N_2,A_2)$的索引词集合被定义为 $\Omega=N_1\cup N_2\cup L_{la}^{M_1}\cup L_{lb}^{M_2}\cup Lk_{lb}^{M_1}\cup Lk_{lb}^{M_2}$，$\Omega$ 包含所有节点、前向链接和后向链接。设 λ 为每个索引项赋一个运行索引的索引函数，业务流程模型 M_i 表示为足迹向量$\vec{g_1}=(g_{1,1},\cdots,g_{1,j},\cdots,g_{1,|\Omega|})$和$\vec{g_2}=(g_{2,1},\cdots,g_{2,j},\cdots,g_{2,|\Omega|})$，在此场景下 i 的取值可能有 1 和 2，进一步地有：

$$g_i,\lambda(\omega)=\begin{cases}0, & \text{if } \omega\notin(N_i\cup Lk_{la}^{M_i}\cup Lk_{lb}^{M_i})\\ \dfrac{1}{2^{|\Omega|-1}}, & \text{if } \omega\in(Lk_{la}^{M_i}\cup Lk_{lb}^{M_i}\\ 1, & \text{if } \omega\in N_i\end{cases}$$

两个流程模型之间的相似性被定义为它们的足迹向量之间夹角的余弦值，计算式如下：

$$Sim(M_1,M_2)=\frac{\vec{g_1}\cdot\vec{g_2}}{|g_1|\cdot|g_2|}=\frac{\sum_{j=1}^{|\Omega|}g_{1,j}\cdot g_{2,j}}{\sqrt{\sum_{j=1}^{|\Omega|}g_{1,j}^2}\cdot\sqrt{\sum_{j=1}^{|\Omega|}g_{2,j}^2}}$$

向量组足迹的计算要求分别对 M_1 和 M_2 的特定节点之间的对应关系进行量化，提出了一个额外的语义相似度量来量化流程模型活动之间的相似性。该度量得分基于节点标签，这些标签被分割成单词。相同的单词评分为 1 分，同义词评分 0.75 分。设 $f_1\in F_1$ 和 $f_2\in F_2$ 是两个不同过程模型的两个活动，其中 f_1 和 f_2 是本例中对应活动标签的词集。如果标签中给定的单词是同义词，则 $synonym(w_1,w_2)$返回 1，如果不是，则返回 0。节点相似度计算式如下：

$$sem(f_1,f_2)=\frac{1.0\cdot|f_1\cap f_2|+0.75\cdot\sum_{(w_1,w_2)\in F_1F_2\times F_2F_1}synonym(w_1,w_2)}{\max(|f_1|,|f_2|)}$$

节点的相似值取值在[0,1]范围内，随后用于对足迹向量的元素进行加权。

3.3.3 行为特征相似性

在 Weidlich 等[49]的研究中，流程模型的行为被概念化为行为概要文件，通过行为概要文件来判断过程模型的一致性，其中执行轨迹来自模型的执行日志。F 是活动集，对于所有的 $f_i,f_j\in F$，依赖被表达为 4 种不同的关系[41]。

(1)严格顺序关系：当所有的踪迹[72]σ 满足在 f_i 和 f_j 组成的集合中，f_i 总是在 f_j 之前，则认为 f_i 和 f_j 有严格的顺序关系。

(2)排他关系：当不存在踪迹 σ 同时包含 f_i 和 f_j，则认为 f_i 和 f_j 是排他关系。

(3)交错序关系：存在踪迹 σ，在 σ 的子序列中，至少存在 σ_i 中有 f_i 在 f_j 之前，至少存在 σ_j 有 f_i 在 f_j 之后，则认为 f_i

和 f_j 是的交错序关系。

(4)共现关系：对于所有的踪迹 σ，既有 f_i 在 f_j 之前，又有 f_i 在 f_j 之后，则认为 f_i 和 f_j 是共现关系。

基于因果行为关系和预定义的活动映射，Weidlich 等定义了一个相似度量如下：将 Λ_{M_1,M_2} 作为 M_1 和 M_2 的匹配集。$F_1^{\sim}=\{f^{M_1}\in F_1 \mid \exists f^{M_2}\in F_2:(f^{M_1},f^{M_2})\in\Lambda_{M_1,M_2}\}$，$F_2^{\sim}$ 的定义类似。进一步地，对于活动对 $(f_1^{M_1},f_2^{M_1})\in P_1$，其中 $P_1\subseteq F_1^{\sim}\cdot F_1^{\sim}$，$F_1^{\sim}$ 包含所有的活动对，相对地，有活动对 $(f_1^{M_2},f_2^{M_2})\in P_2$，其中 $P_2\subseteq F_2^{\sim}\cdot F_2^{\sim}$。$F_2^{\sim}$ 和 $(f_1^{M_2},f_2^{M_2})$，$(f_2^{M_1},f_2^{M_2})\in\Lambda_{M_1,M_2}$ 处于同种关系。基于以上论述，M_1 和 M_2 的相似度被定义为：

$$sim(M_1,M_2)=\frac{|P_1|+|P_2|}{|F_1^{\sim}\cdot F_1^{\sim}|+|F_2^{\sim}\cdot F_2^{\sim}|}$$

可以看出，行为概要文件不仅可以根据流程模型计算，还可以根据执行日志计算。该方法关注模型间踪迹的关系，因此在处理复杂节点匹配时具有优势，但是用行为概要文件计算相似度时也会出现计算不准确的情况，原因在于行为概要文件不能发现循环结构中的顺序约束[41]。

Xing 等[47]通过基于行为特征和 Jaccard 系数的 5 个基本相似性度量来综合评估模型相似性，并且通过实验评估发现该指标接近人类的相似性评估。

3.4 基于人类评估的相似度

3.4.1 众测相似度评估

相似性度量的另一个重要方面是人对过程模型相似度的判断。人们可以根据自己的个人知识，主观地量化过程模型之间的相似性，通过人工辅助模型相似度决策，可使相似度结果在原基础上有一个修正。根据所涉及人员的知识，可以分为 3 种类型：1)过程专家；2)过程参与者；3)大众。过程专家对公司或其部门的过程环境有一定的了解，而过程参与者则是特定过程或过程部件的专家。因此，可以假设过程专家从更一般的角度量化相似性，而过程参与者从更详细的角度；与此相反，大众只从过程描述(如业务流程模型)中获得知识。大众根据自己对过程描述的个人解释来量化相似性。鉴于此，可以在相似度度量中加入学习算法，该算法通过人工来判断自动相似度计算的正确性，另一方面，可以通过游戏化方法和基于群体的相似度估计来整合人工输入。目前有 3 种常见的使用人工辅助模型相似度决策的方法。

(1)Klinkmuller 等[50]的研究中，使用用户反馈的输入来改进流程模型元素的匹配。

(2)Rodríguez 等[51]使用基于人群的匹配确定。

(3)Laue 等[52]通过比较与流程模型相关的标签确定相似值。

Laue 等考虑了通过社会标记来计算过程模型的相似性。其基本思想是，模型之间的相似程度越高，它们共享的标签就越多。因此利用 Dice 系数计算相似度，过程模型 M_1 带有多重标签集 $Tags_1$，过程模型 M_2 带有多重标签集 $Tags_2$，计算式如下：

$$Sim(M_1,M_2)=\frac{2|Tags_1\cap Tags_2|}{|Tags_1|+|Tags_2|}$$

其中 $|Tags_i|$ 是 $Tags_i$ 中标签列表的元素数量。

3.4.2 用户反馈

Klinkmüller 等希望改进业务流程模型与用户反馈输入的匹配，用户将看到自动匹配方法的匹配，用户输入被传递给自动匹配方法，进而删除不正确的匹配并添加正确的匹配。该方法使用输入来适应底层的标签相似度计算。

3.5 小结

文本相似度主要处理业务流程模型中和自然语言相关的文本标签，这些标签通过句法和语义方面的分析，形成的结果是流程模型相似度度量的重要来源之一。文本相似度计算可大致分为基于字符串、单词编辑距离、语义相似度、词袋相似度 4 个方面。基于字符串计算的经典算法有编辑距离、Dice 系数、Jaccard 系数、最长公共子串、Jaro-winkler 相似度和单词编辑距离。单词编辑距离类似于字符串编辑距离，不同之处在于对长句的分割，不同的映射关系对最终计算的影响不同。由于文本具有语义，通常需要进一步考虑语义相似度。语义相似度又可以根据语义来源分为基于拓扑相似和基于统计相似。基于拓扑相似的计算方法通常借助于构建本体或者语义信息网络等知识拓扑结构来计算概念、词语等的相关度；基于统计相似的计算方法则通过搜索关键词返回的共现词数来反映相似度。

图结构相关的相似度量大致可以分为基于图结构的相似度量和基于业务流程感知的控制流相似度度量。一般情况下，基于图结构的模型之间的相似性可以量化，例如求两个模型的最大公共子图大小。利用图结构计算相似度的相关技术有：计算图编辑距离、计算过程模型的图结构之间的图同构、使用特殊类图确定模型间的相似性，Minor 等[4]提出了一种基于公共节点和边相似度的度量技术。树编辑距离和图编辑距离的基本思想一致，想要得到准确的相似度计算结果的难点在于对代价函数和编辑距离的求解。La Rosa 等[68]在他们的流程合并方法中使用了一个上下文相似性的概念，Yan 等[1]的基于特征的相似度估计就是该技术的一个应用，在节点的上下文中使用了与结构有关的信息。

在相似性度量中，可以通过日志中相同执行序列的数量来确定两个模型之间的相似性，执行轨迹可以通过模拟运行或在流程实际执行期间生成，执行轨迹通常存储在日志中供进一步分析，行为相似度计算侧重于进程的执行轨迹。计算行为相似度通常使用的技术有：Gerke 等[45]提出的通过轨迹的最长公共子序列反映相似性的方法、通过执行精度和召回度量的最长公共子序列、因果足迹相似性、行为特征相似性。

相似性度量的另一个重要方面是人对过程模型相似度的判断。可以在相似度度量中加入学习算法，该算法使用人工来判断自动相似度计算的正确性。可以通过游戏化方法和基于群体的相似度估计来增加人工决策的数据量，目前已知有 3 种方法使用人工辅助模型相似度决策，分别是使用用户反馈的输入来改进流程模型元素的匹配、使用基于人群的匹配确定、通过比较与流程模型相关的标签确定相似值。Becker 等[2]考虑了通过社会标记来计算过程模型的相似性。其基本思想是，模型之间的相似程度越高，共享的标签就越多。

4 模型相似度应用

对相关文献的资料整理结果显示，使用业务流程模型相似

度技术进行效率提升的常见应用场景有一致性检测、标准化、流程模型搜索和模型重用。4 种应用的具体对比如表 2 所列。

表 2 相似度技术应用场景分析

Fig. 2 Application scenario analysis of similarity technology

应用场景	基本思想	相似度技术的特点	参考文献
一致性检测	通过业务流程模型相似度计算方法将业务流程模型的相似性转化为一类或多类定量指标,对相似度特征属性进行形式化描述	在一致性检测过程中,可能出现图在表现形式上差异较大、难以直接进行比较的情况,这时需要借助其他技术的支持,如模型变换技术、追踪技术等,如果希望尽可能简化相似度的计算过程,可以使用行为相似度或文本相似度等技术	[41,50,72-75]
标准化	从不同的过程模型变体或版本生成一个标准化的过程模型	标准化的进一步应用是流程变体的识别,业务流程模型的变化挖掘可以从过程挖掘方面入手,过程挖掘的方法之一是通过日志的过程挖掘技术来发现流程变化,文本相似度计算和基于日志提取的因果足迹相似度计算是较为常用的方法	[75-77]
流程模型搜索	面对模型数量较多的存储库,人工管理的效率非常低下,这时需要用到模型相似度对流程模型进行搜索	效率问题是模型检索中的首要问题,如果通过在线计算每个模型间相似度,计算效率将大幅度降低,可以通过计算核心语义等方法先进行模糊搜索,缩小计算范围,再进一步进行相似度计算,该方法在语义相似度研究方面有相关应用	[39,41,43,76]
重用	流程建模被认为是耗时且昂贵的,因此可以通过部分或者完全重用已有的流程模型来更有效地执行该活动	业务流程模型的重用不基于特定的相似度计算方法,但是模型能否重用需要根据模型的拟合程度进行判断,对于模型间差异点的识别也有相应的要求	[28,38,78]

4.1 一致性检测

一致性检测是检测一个模型与另一个模型的一致性,通过业务流程模型相似度计算方法将业务流程模型的相似性转化为一类或多类定量指标,对相似度特征属性进行形式化描述[79]。一致性检测根据两种不同的子目标可以进一步区分[69]:一是衡量模型与给定参考模型的拟合程度,二是量化不同任意模型之间的差异。第一个子目标可以用来确定模型与参考模型在调节意义上的一致性,可以将引用模型视为最佳实践参考模型,通过相似度量,可以通过模型和最佳实践参考模型之间的差异来寻找过程改进的机会。在该目标下,另一个应用是将引用模型视为实际流程应该遵守的某种规则。对于第二个子目标,差异的检测不局限于参考模型,可以使用任意的其他模型。在跨国公司根据不同的模型在不同的国家执行相同过程时[72],可以通过相似性分析发现的差异帮助过程分析人员统一这些过程或过程改进。

在建模过程中,模型信息一致性的维护是开发人员必须解决的一个重要问题。经验表明,如果这方面的工作单靠开发者手工处理,不仅烦琐而且容易出现差错和遗漏,极不现实,特别是随着业务流程模型不断复杂化和快速迭代更新,提供一种一致性检测的计算方法尤为重要[1]。

在流程驱动的应用程序工程中可以通过流程建模来弥合业务需求和系统规范之间的差距。然而,业务流程建模活动的目的不同导致模型在不同的抽象级别和透视图中的对齐出现问题,例如,流程建模师构建 BPMN 流程模型[73],而另一个流程建模师将 BPMN 模型转换为抽象 BPEL 流程,这将产生从不同透视图和抽象级别的两个不同的抽象流程,因此,检查相应模型的一致性是过程建模理论和实践的挑战之一[41];另一方面,在流程映射、流程集成和差异检测中,流程的一致性检测也至关重要[74]。因此,研究者们针对模型的一致性检测展开了多方面的研究。

模型一致性的检测实质上是模型间对应关系的捕获,其被认为是一种系统间转换的对齐[74]。如果一个转化通过对应关系与某个对应的活动相关联,那么就称这个转换是对齐的。Weidlich 等[75]受垂直和水平流程集成概念的影响,提出了模型间的垂直和水平对齐的概念。在一致性检测中,以业务为中心的流程模型和技术流程模型通常采用不同的抽象级别,建模目的不同,导致相同的业务案例也会从不同的角度和粒度进行分析。Zhang 等[74]认为业务流程模型建模一般在 3 个抽象级别上进行,即系统需求的概念级别、系统规格说明的逻辑级别和软件开发的物理级别。在不同抽象级别的流程模型间的对齐方式被称为垂直对齐。另一方面,在同一抽象级别上的变体流程节点之间的对齐叫做水平对齐,这些变体可能是由不同国家或不同产品的业务策略等因素不同导致的。然而,无论是使用水平对齐还是垂直对齐,对一致性的检测都不会造成影响,这一论述需要对齐操作本身能够识别在垂直或水平方向上变体活动节点的对应关系。

在一致性检测方面,模型的行为相似度计算方法可以用来进行一致性的检测,根据流程模型的执行获得跟踪。在 Van 等的研究中,他们将活动执行序列表示为跟踪,模型间跟踪的对齐是一致性检测的基准[61],衡量一致性的对齐要求一个模型中所有可能到达的点火序列在另一个模型中具有相应的序列。在某些情况下,模型在表现形式上差异较大,难以直接进行比较,需要借助其他技术的支持,如模型变换技术、追踪技术等。

Zhang 等[74]认为通过行为概要的方法检测一致性忽略了业务流程的数据流,因此提出了一种系统的方法定量地度量不同抽象级别的业务流程之间的一致性,使用关键事件约束从控制流和数据流的角度来量化一致性。在之后的研究中,Zhang 等在检测数据感知过程的流程变体的差异性时,发现基于传统的跟踪等价性和行为概要的方法无法很好地解决度量数据感知过程一致性的问题,因此提出了一种系统性的构造 ACG 算法,通过新的 ACG 概念来描述活动约束,更加准确地检测出了数据感知过程之间的差异。

Weidlich 等[75]在控制流方面展开了研究,研究了一致性概念的充分性。他们认为踪迹等价不太适合作为一致性的概念,因为它不太符合我们对一致性的感知。相比之下,忽略单一活动重复的基于行为特征的标注更符合对一致性的感知。

4.2 标准化

流程作为一种动态结构,需要随时根据业务需求作出

改变，因此会产生很多流程版本。在管理流程版本时，一种有效的方法是建立管理知识库，通过管理知识库来提高流程的进化效率[75]，使得企业在日后面对新改变时，能够促进流程演化以适应环境的变化。

标准化的最终目的是从不同的过程模型变体或版本生成一个标准化的过程模型。由于业务流程的更改和流程模型在不同环境下的自适应变化，存在大量来自于同一模型的流程变体。流程变体在结构功能上非常相似，分析这些模型的关键在于对异变位置的识别，然后通过归纳生成一个目标参考流程模型。这一类的场景涉及企业的合并与收购、组织中不同部门相似流程的重组，以及标准化业务流程而进行的组织间协作。

参照模型的归纳生成可以看作是过程模型相似度度量的一种应用，在这种情况下，可以使用度量来识别不同组织的相应流程模型或不同的参考模型，从而归纳生成一个参考模型，其中包含输入模型的最佳执行片段。

标准化的进一步应用是流程变体的识别。在数据库中将与不同用户组、合法规则等相关的变体作为单独的流程模型存储在存储库中，这样做的原因在于，不断变化的合法规则或业务需求可能需要调整或标准化这些变体，以满足新的需求。而相似度量可以通过识别相关流程模型来帮助确定受影响的流程。业务流程模型的变化挖掘可以从过程挖掘方面入手。过程挖掘的方法之一是通过日志的过程挖掘技术来发现流程变化，文本相似度计算和基于日志提取的因果足迹相似度计算是较为常用的方法。过程挖掘方面的研究较为丰富，Ayora等[76]研究了在已知流程变体的情况下，挖掘与流程变体适合度最高的过程模型；Günther等[80]提出了一种在自适应流程管理系统中挖掘变化日志的方法；Pourmasoumi等[77]提出了一种从事件日志集合中提取可变的过程片段的方法，从事件日志角度出发，通过定义变元片段，提出相应的日志挖掘方法来发现日志中的变元片段。

4.3 流程模型搜索

大型模型库的有效使用需要访问和管理模型的方法，特别是快速搜索方法[41]。作为流程执行和进一步的业务流程管理活动的知识库[76]，一个存储库中可能包含数百个甚至数千个模型。传统的搜索方式可以由员工进行，通过文件和字段查询的方式找到他们所涉及的流程模型。但是面对模型数量较多的存储库，人工搜索的效率和准确率非常低，这时需要用到模型相似度技术对流程模型进行搜索。在模型搜索时使用的方法有穷尽搜索和非穷尽搜索。

流程模型搜索还可以应用于基于相似性的服务搜索中。面向服务的体系结构的主要思想之一是服务的可替代性，如果服务过程作为流程模型或者流程模型的一部分，那么相似性搜索就很可能识别出此类交换的候选者，并且确定从这些标准化软件服务的集成中受益的流程。因此，只要服务之间的相似性度量或服务搜索是基于底层流程的，原则上，这些工作就可以应用于流程模型的相似性度量[39]。

业务流程模型搜索通过计算模型间的相似度，反馈给检索系统与给定模型相似度高的业务流程模型。在这个过程中，在选择合适相似度计算方法情况下，效率问题是模型检索中的首要问题，如果通过在线计算每个模型间的相似度，计算效率将大幅度降低，因此可以通过计算核心语义等方法先进行模糊搜索，缩小计算范围，再进一步进行相似度计算。该方法在语义相似度研究方面有相关应用[78]。

4.4 模型重用与替换

随着时代的发展，业务流程变得多样化且关系复杂，业务流程模型也变得越来越复杂。对于企业来说，想要构建业务流程模型或对业务流程模型进行重构是耗时且昂贵的[73]，通过研究相似领域的业务流程模型可以发现，业务流程模型在许多片段上存在相似性，这些具有相似性的片段可以通过重用来减少建模的时间代价和成本[69]。考虑通过业务流程模型相似度进行相似片段的匹配是模型重用的有效手段之一。

流程建模被认为是耗时且昂贵的，因此可以通过部分或者完全重用已有的流程模型来更有效地执行该活动[78]。在建模活动期间，模型编辑器可以推荐现有的模型，然后建模人员可以重用这些模型。这样的推荐函数通常使用相似性度量来从存储库中推荐合适的模型[73]。重用的另一个应用场景是流程模型的模块化。在此场景中，使用相似性度量来检测不同流程模型的相似子图。这些子图随后可能被提取到一个新的流程模型中，以提高流程模型的可理解性、一致性和可维护性。

业务流程模型的重用不基于特定的相似度计算方法，但是模型能否重用需要根据模型的拟合程度进行判断，对于模型间差异点的识别也有相应的要求。

4.5 小结

业务流程模型相似度在工程领域有着广泛的应用，通过业务流程模型相似度实现模型检索、模型重用等操作，模型的检索能够有效增加模型仓库管理的效率，模型重用能够减少重复的工作量，从而降低工程成本。另一方面，较之应用于工程场景下的业务流程模型相似度，在学者研究的过程中，业务流程模型相似度的应用场景和计算方法在原有的基础成果上得到了进一步的扩充，但是业务流程模型相似度的应用最终会回归工程背景，因此，利用业务流程模型技术提高工作效率才是最终目的。

结束语 目前，业务流程模型相似度的主要研究在于对模型相似度计算方法的研究，而对于模型相似度计算结果的评估分析较少，模型相似度计算结果的准确程度不得而知。因此，对相似性度量的度量标准进行更深入的分析非常必要。

另一方面，在业务流程模型相似度计算领域，现有的方法更关注于比较“扁平”的流程模型。实际上，对于一些复杂的流程，往往进行多维度的描述，流程模型的维度和要素更多，在对这一类模型进行相似度计算时，往往是将其分解为子流程模型，从而控制需要对比的要素和维度，但流程的分解在很大程度上取决于建模者、域和建模目标，因此最后相似度计算的准确性难以保证。如何通过更加科学有效地方式计算复杂流程模型是需要进一步关注的问题。

在未来对业务流程模型的研究中，可以进一步关注通过自动化方法计算出的相似值与人工判断的比较。虽然在搜索流程模型集合的相似性度量中，基于相似性的搜索功能检索模型已经与人类预期的结果进行了比较，但由于比较的不是

相似度值本身，而是一个更复杂的过程结果（如搜索的结果），因此这只是一种间接的确定相似度值合理性的方法。有研究者认为应该有更多的研究致力于理解人类如何主观地评价模型的相似性，进一步研究现有的度量在多大程度上覆盖了人类的主观度量，以及如何在自动化相似性度量中适当地实现这一点。由于相似性度量的最终目的是在人类主观量化模型相似度或利用这些量化结果的任务中支持人类决策，因此可以对过程模型相似度量过程中的人工输入进行更深入的研究，人工输入可能有助于提高自动匹配和相似性度量的质量。

另一方面，研究者们可以进一步关注异构模型的统一度量标准问题，在实际的工程场景下，常常出现模型异构导致相似度准确性不高或无法计算的问题，因此，建立统一的度量标准对加强模型仓库的管理和模型搜索有着重大意义。

参 考 文 献

[1] YAN Z, DIJKMAN R, GREFEN P. Fast business process similarity search with feature-based similarity estimation[C]// OTM Confederated International Conferences on the Move to Meaningful Internet Systems. Berlin: Springer, 2010: 60-77.

[2] BECKER M, LAUE R. A comparative survey of business process similarity measures[J]. Computers in Industry, 2012, 63(2): 148-167.

[3] THALER T, SCHOKNECHT A, FETTKE P, et al. A comparative analysis of business process model similarity measures [C]// International Conference on Business Process Management. Cham: Springer, 2016: 310-322.

[4] MINOR M, TARTAKOVSKI A, BERGMANN R. Representation and structure-based similarity assessment for agile workflows[C]// International Conference on Case-Based Reasoning. Berlin: Springer, 2007: 224-238.

[5] DONGEN B, DIJKMAN R, MENDLING J. Measuring similarity between business process models[M]// Seminal Contributions to Information Systems Engineering. Berlin: Springer, 2013: 405-419.

[6] CORRALES J C, GRIGORI D, BOUZEGHOUB M. BPEL processes matchmaking for service discovery[C]// OTM Confederated International Conferences "On the Move to Meaningful Internet Systems". Berlin: Springer, 2006: 237-254.

[7] YAN Z, SUN B, WANG T. Research on Business Process Model Analysis Method Based on UML [J]. Computer Engineering and Applications, 2004, 40(29): 226-228.

[8] CAO B, WANG J, FAN J. Interprocess Element Mapping Based on Petri Net [J]. Journal of Software, 26(3): 474-490.

[9] ZENG Y. Implementation of BPR process based on extended Petri net model [J]. Modular Machine Tool & Automatic Manufacturing Technique, 2005(9): 37-39, 48.

[10] SMIRNOV S, REIJERS H A, WESKE M, et al. Business process model abstraction: a definition, catalog, and survey[J]. Distributed & Parallel Databases, 2012, 30(1): 63-99.

[11] SETIAWAN Y, SUNGKONO K R, SARNO R. A new similarity method based on weighted graph models for matching parallel business process models[J]. International Journal of Intelligent Engineering and Systems, 2020, 13(5): 267-276.

[12] AKKIRAJU R, IVAN A. Discovering business process similarities: An empirical study with SAP best practice business processes[C]// International Conference on Service-Oriented Computing. Berlin: Springer, 2010: 515-526.

[13] AHN H, CHANG T W. Measuring similarity for manufacturing process models[C]// IFIP International Conference on Advances in Production Management Systems. Cham: Springer, 2018: 223-231.

[14] DIJKMAN R, DUMAS M, VAN DONGEN B, et al. Similarity of business process models: Metrics and evaluation[J]. Information Systems, 2011, 36(2): 498-516.

[15] BUNKE H. On a relation between graph edit distance and maximum common subgraph[J]. Pattern Recognition Letters, 1997, 18(8): 689-694.

[16] ZHANG H, WANG G, ZHONG Y. Text Similarity Calculation Based on Hamming Distance[J]. Computer Engineering and Applications, 2001, 37(19): 2-7.

[17] HUANG C H, YIN J, HOU F. A Text Similarity Measurement Method Combining Term Semantic Information and TF-IDF Method [J]. Chinese Journal of Computers, 2011, 34(5): 856-864.

[18] DICE L R. Measures of the amount of ecologic association between species[J]. Ecology, 1945, 26(3): 297-302.

[19] WANG Y, QIN J, WANG W. Efficient approximate entity matching using jaro-winkler distance[C]// International Conference on Web Information Systems Engineering. Cham: Springer, 2017: 231-239.

[20] NIWATTANAKUL S, SINGTHONGCHAI J, NAENUDORN E, et al. Using of Jaccard Coefficient for Keywords Similarity [C]// Iaeng International Conference on Internet Computing & Web Services. International Association of Engineers, 2013: 237-245.

[21] JACCARD P. The distribution of the flora in the alpine zone. 1 [J]. New Phytologist, 1912, 11(2): 37-50.

[22] BERGROTH L, HAKONEN H, RAITA T. A survey of longest common subsequence algorithms[C]// Proceedings Seventh International Symposium on String Processing and Information Retrieval. IEEE, 2000: 39-48.

[23] DUMAS M, GARCÍA-BAÑUELOS L, DIJKMAN R M. Similarity search of business process models[J]. IEEE Data Engineering Bulletin, 2009, 32(3): 23-28.

[24] JARO M A. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida[J]. Journal of the American Statistical Association, 1989, 84(406): 414-420.

[25] LEACOCK C, CHODOROW M. Combining local context and WordNet similarity for word sense identification[J]. WordNet: An Electronic Lexical Database, 1998, 49(2): 265-283.

[26] WU Z, PALMER M. Verb semantics and lexical selection[J]. arXiv: cmp-lg/9406033.

[27] RESNIK P. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language[J]. Journal of Artificial Intelligence Research, 1999, 11: 95-130.

[28] LIN D. An information-theoretic definition of similarity[C]// In-

ternational Conference on Machine Learning. 1998：296-304.

[29] JIANG J J，CONRATH D W. Semantic similarity based on corpus statistics and lexical taxonomy[J]. arXiv：cmp-lg/9709008.

[30] CILIBRASI R L，VITANYI P M B. The google similarity distance[J]. IEEE Transactions on Knowledge and Data Engineering，2007，19(3)：370-383.

[31] RECCHIA G，JONES M N. More data trumps smarter algorithms：Comparing pointwise mutual information with latent semantic analysis[J]. Behavior Research Methods，2009，41(3)：647-656.

[32] LANDAUER T K，FOLTZ P W，LAHAM D. An introduction to latent semantic analysis[J]. Discourse processes，1998，25(2/3)：259-284.

[33] LI Y，MCLEAN D，BANDAR Z A，et al. Sentence similarity based on semantic nets and corpus statistics[J]. IEEE Transactions on Knowledge and Data Engineering，2006，18(8)：1138-1150.

[34] GACITUA-DECAR V，PAHL C. Automatic Business Process Pattern Matching for Enterprise Services Design[C]// 2009 World Conference on Services-II. 2009：313-374.

[35] LEOPOLD H，NIEPERT M，WEIDLICH M，et al. Probabilistic optimization of semantic process model matching[C]// International Conference on Business Process Management. Berlin：Springer，2012：319-334.

[36] FELLBAUM C. A semantic network of English：the mother of all WordNets[C]// EuroWordNet：A multilingual Database with Lexical Semantic Networks. Dordrecht：Springer，1998：137-148.

[37] LI S J. Research on Sentence Relevance Based on Semantic Computing [J]. Computer Engineering and Applications，2002，38(7)：3-12.

[38] ANTUNES G，BAKHSHANDEH M，BORBINHA J，et al. The process model matching contest 2015[M]. Geellschaftfür Informatik，2015.

[39] CAYOGLU U，DIJKMAN R，DUMAS M，et al. Report：The process model matching contest 2013[C]// International Conference on Business Process Management. Cham：Springer，2013：442-463.

[40] LI C，REICHERT M，WOMBACHER A. On measuring process model similarity based on high-level change operations[C]// International Conference on Conceptual Modeling. Berlin：Springer，2008：248-264.

[41] KUNZE M，WEIDLICH M，WESKE M. Behavioral similarity—a proper metric[C]// International Conference on Business Process Management. Berlin：Springer，2011：166-181.

[42] VAN DER AALST W M P，MEDEIROS A K，WEIJTERS A. Process equivalence：Comparing two process models based on observed behavior[C]// International Conference on Business Process Management. Berlin：Springer，2006：129-144.

[43] XU Z，KUN Z，NING L，et al. Overview of graph editing distance [J]. Computer Science，2018，45(4)：11-18.

[44] MELCHER J，SEESE D. Visualization and clustering of business process collections based on process metric values[C]// 2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. IEEE，2008：572-575.

[45] GERKE K，CARDOSO J，CLAUS A. Measuring the compliance of processes with reference models[C]// OTM Confederated International Conferences on the Move to Meaningful Internet Systems. Berlin：Springer，2009：76-93.

[46] ZHA H，WANG J，WEN L，et al. A workflow net similarity measure based on transition adjacency relations[J]. Computers in Industry，2010，61(5)：463-471.

[47] XING J，ZHANG X，SONG W，et al. BPEL Similarity—a metric based on activity constraint graphs[C]// Asia-Pacific Conference on Business Process Management. Cham：Springer，2013：39-55.

[48] LI H，GUO C，QIU W. Computing Method of Similarity of Normal Cloud Model [J]. Acta Electronica Sinica，2011，39(11)：25-61.

[49] WEIDLICH M，DIJKMAN R，MENDLING J. The ICoP framework：Identification of correspondences between process models [C]// International Conference on Advanced Information Systems Engineering. Berlin：Springer，2010：483-498.

[50] KLINKMÜLLER C，LEOPOLD H，WEBER I，et al. Listen to me：Improving process model matching through user feedback [C]// International Conference on Business Process Management. Cham：Springer，2014：84-100.

[51] RODRÍGUEZ C，KLINKMÜLLER C，WEBER I，et al. Activity matching with human intelligence[C]// International Conference on Business Process Management. Cham：Springer，2016：124-140.

[52] LAUE R，BECKER M. Evaluating social tagging for business process models [C] // International Conference on Business Process Management. Springer，2012：280-291.

[53] SHAHMIRZADI O，LUGOWSKI A，YOUNGE K. Text similarity in vector space models：a comparative study[C]// 2019 18th IEEE International Conference on Machine Learning and Applications(ICMLA). IEEE，2019：659-666.

[54] JI C，WANG J，GENG R. Weak selection backtracking matching tracking algorithm based on Dice coefficient [J]. Journal of Northeastern University(Natural Science)，201，42(02)：189-195.

[55] SCHOKNECHT A，FISCHER N，OBERWEIS A. Process model search using latent semantic analysis [C] // International Conference on Business Process Management. Cham：Springer，2016：283-295.

[56] TANG H G. Research and Implementation of Data Migration Based on Business Process Model Similarity [D]. Shanghai：Shanghai Institute of Computing Technology，2020.

[57] MERKX D，FRANK S L，ERNESTUS M. Semantic sentence similarity：size does not always matter[J]. arXiv：2106.08648.

[58] EHRIG M，KOSCHMIDER A，OBERWEIS A. Measuring similarity between semantic business process models[C]// APCCM. 2007：71-80.

[59] HUANG G，ZHOU Z. Research on Semantic Similarity Calculation of Concept Based on Domain Ontology[J]. Computer Engineering and Design，2007，28(10)：2460-2463.

[60] ZHANG P. Computational Model of Sentence Similarity based

on Multi-Feature Fusion [J]. Computer Engineering and Applications, 2010, 46(26): 136-137.
[61] VAN GLABBEEK R J, WEIJLAND W P. Branching time and abstraction in bisimulation semantics[J]. Journal of the ACM (JACM), 1996, 43(3): 555-600.
[62] LOPEZ-GAZPIO I, MARITXALAR M, LAPATA M, et al. Word n-gram attention models for sentence similarity and inference[J]. Expert Systems with Applications, 2019, 132: 1-11.
[63] QU R, FANG Y, BAI W. Computing semantic similarity based on novel models of semantic representation using Wikipedia[J]. Information Processing & Management, 2018, 54(6): 1002-1021.
[64] KRISSINEL E B, HENRICK K. Common subgraph isomorphism detection by backtracking search[J]. Software: Practice and Experience, 2004, 34(6): 591-607.
[65] RAYMOND J W, GARDINER E J, WILLETT P. Rascal: Calculation of graph similarity using maximum common edge subgraphs[J]. The Computer Journal, 2002, 45(6): 631-644.
[66] DIJKMAN R, DUMAS M, GARCÍA-BAÑUELOS L. Graph matching algorithms for business process model similarity search[C] // International Conference on Business Process Mana-gement. Berlin: Springer, 2009: 48-63.
[67] YU H. Research on Business Process Similarity Measurement Based on Internal Structure [D]. Shenzhen: Shenzhen University, 2016.
[68] LA ROSA M, DUMAS M, UBA R, et al. Business process model merging: An approach to business process consolidation[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 2013, 22(2): 1-42.
[69] SCHOKNECHT A, THALER T, FETTKE P, et al. Similarity of business process models—a state-of-the-art analysis[J]. ACM Computing Surveys(CSUR), 2017, 50(4): 1-33.
[70] ZHOU C, LIU C, ZENG Q, et al. A comprehensive process similarity measure based on models and logs[J]. IEEE Access, 2019, 7: 69257-69273.
[71] MENDLING J, VAN DONGEN B F, VAN DER AALST W M P. On the Degree of Behavioral Similarity between Business Process Models[C]// EPK. 2007: 39-58.
[72] VAN DER AALST W, WEIJTERS T, MARUSTER L. Workflow mining: Discovering process models from event logs[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1128-1142.
[73] SONG W, JACOBSEN H A. Static and dynamic process change [J]. IEEE Transactions on Services Computing, 2016, 11(1): 215-231.
[74] ZHANG X, SONG W, WANG J, et al. Measuring business process consistency across different abstraction levels[J]. IEEE Transactions on Network and Service Management, 2018, 16(1): 294-307.
[75] WEIDLICH M, MENDLING J, WESKE M. Efficient consistency measurement based on behavioral profiles of process models [J]. IEEE Transactions on Software Engineering, 2010, 37(3): 410-429.
[76] AYORA C, TORRES V, DE LA VARA J L, et al. Variability management in process families through change patterns[J]. Information and Software Technology, 2016, 74: 86-104.
[77] POURMASOUMI A, BAGHERI E. Business process mining [J]. Encyclopedia with Semantic Computing and Robotic Intelligence, 2017, 1(1): 1-32.
[78] KOSCHMIDER A, FELLMANN M, SCHOKNECHT A, et al. Analysis of process model reuse: Where are we now, where should we go from here? [J]. Decision Support Systems, 2014, 66: 9-19.
[79] LIU H, XU D. A survey of Semantic Similarity and Relevance Computing Based on Ontology [J]. Computer Science, 2012, 39(2): 8-13.
[80] GÜNTHER C W, RINDERLE S, REICHERT M, et al. Change mining in adaptive process management systems[C]// 14th International Conference on Cooperative Information Systems, 2006: 309-326.

JIAN Kaiyu, born in 1998, postgraduate. His main research interests include intelligent software testing and similarity of business process model.

SHI Yaqing, born in 1981, professor, is a member of China Computer Federation. Her main research interests include intelligent software testing, temporal and spatial data processing.

(责任编辑:何杨)