



计算机科学

COMPUTER SCIENCE

支持多租户模式的业务流动态定制模型

张纪林, 邵玉曹, 任永坚, 袁俊峰, 万健, 周丽

引用本文

张纪林, 邵玉曹, 任永坚, 袁俊峰, 万健, 周丽. [支持多租户模式的业务流动态定制模型](#)[J]. 计算机科学, 2022, 49(6A): 705-713.

ZHANG Ji-lin, SHAO Yu-cao, REN Yong-jian, YUAN Jun-feng, WAN Jian, ZHOU Li. [Dynamic Customization Model of Business Processes Supporting Multi-tenant](#)[J]. Computer Science, 2022, 49(6A): 705-713.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[一种面向多租户的 Linux 容器集群组网方法](#)

Linux Container Cluster Networking Approach for Multiple Tenants

计算机科学, 2018, 45(9): 46-51. <https://doi.org/10.11896/j.issn.1002-137X.2018.09.006>

[带数据流的面向服务的业务流程模型变化传播 Petri 网方法](#)

Change Propagation Method of Service-oriented Business Process Model with Data Flows Based on Petri Net

计算机科学, 2018, 45(6A): 545-548.

[基于 Petri 网行为紧密度的业务流程配置优化分析](#)

Optimized Analysis of Business Process Configuration Based on Petri Net Behavior Closeness

计算机科学, 2017, 44(Z6): 539-542. <https://doi.org/10.11896/j.issn.1002-137X.2017.6A.120>

[业务流程模型抽象中最优子流程数的确定](#)

Determining Optimal Number of Subprocesses in Business Process Model Abstraction

计算机科学, 2017, 44(10): 245-248. <https://doi.org/10.11896/j.issn.1002-137X.2017.10.044>

[业务流程模型抽象中基于约束的行为聚类方法研究](#)

Constraint-based Activity Clustering in Business Process Model Abstraction

计算机科学, 2017, 44(1): 259-263. <https://doi.org/10.11896/j.issn.1002-137X.2017.01.048>

支持多租户模式的业务流程动态定制模型

张纪林^{1,2,3} 邵玉曹^{2,3} 任永坚^{2,3} 袁俊峰^{2,3} 万健^{2,3,4} 周丽^{2,3}

1 杭州电子科技大学网络空间安全学院 杭州 310018

2 杭州电子科技大学计算机学院 杭州 310018

3 杭州电子科技大学复杂系统建模与仿真教育部重点实验室 杭州 310018

4 浙江科技学院信息与电子工程学院 杭州 310023

(jilin.zhang@hdu.edu.cn)

摘要 流程定制是实现业务流程个性化服务的一种重要手段,在使用单一软件系统的情况下,其通过调整业务流程模型的内部结构来提供差异性业务服务。然而,随着业务流程的规模和复杂程度日益增加,现有的流程定制技术在应对复杂多变的业务流程时,需要对流程模型进行重构,影响了流程定制的开发效率。因此,提供一种高效的流程定制方法一直是业务流程领域的研究热点。文中从多租户应用的角度出发,提出了一种支持多租户模式的业务流程动态定制模型。首先采用可变性任务节点组装的方式构建业务子流程,通过租户感知器实现租户身份判别和流程实例派生;其次针对租户的可变性业务需求,提供一种流程动态定制方法;最后结合实例分析,验证了该模型的有效性。

关键词: 业务流程;个性化服务;多租户;可变性;动态定制

中图法分类号 TP311

Dynamic Customization Model of Business Processes Supporting Multi-tenant

ZHANG Ji-lin^{1,2,3}, SHAO Yu-cao^{2,3}, REN Yong-jian^{2,3}, YUAN Jun-feng^{2,3}, WAN Jian^{2,3,4} and ZHOU Li^{2,3}

1 School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China

2 School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

3 Key Laboratory of Complex System Modeling and Simulation, Hangzhou Dianzi University, Hangzhou 310018, China

4 School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

Abstract Process customization is an essential means to realize personal services of business processes. It provides differential business services by adjusting the internal structure of business process model while using a single software system. However, with the increasing scale and complexity of business processes, the existing process customization technology needs to reconstruct the process model when dealing with those complex and changeable business processes, which affects the development efficiency of process customization. Therefore, providing an efficient process customization method has always been a research hotspot in the field of business processes. From the perspective of multi-tenant application, this paper proposes a dynamic customization model of business processes supporting multi-tenant. Firstly, the business sub-process is constructed by means of assembling variable task nodes and then tenant identify identification and process instance derivation are realized by tenant sensor. Secondly, a dynamic process customization method is provided for the varying requirements of tenants. Finally, combined with case analysis, the validity of the model is verified.

Keywords Business process, Personal service, Multi-tenant, Variability, Dynamic customization

1 引言

业务流程是为实现特定价值目标而由不同的人共同完成的一系列活动^[1]。近年来互联网产业迅猛发展,业务流程的应用越来越广泛,这些应用包括在线教育、医疗健康、财务预算以及电子商务等。业务流程通过 BPaaS 云服务模式为用户提供服务^[2]。然而在多租户应用场景下,用户的业务需求往往存在一定差异,如何高效地设计出符合不同租户要求的

业务流程已成为业务流程领域的重要问题^[3]。

现有的流程定制^[4]方法通过调整流程模型的内部结构来提供差异性业务服务。通常有两种技术手段能达到流程定制的目的:流程模板技术^[5]和流程配置技术^[6]。流程模板技术用模板表示通用的流程片段,基于模板进行定制,产生符合需求的业务流程^[7],但该定制方法受流程模板约束,缺乏一定的灵活性。流程配置技术是对业务流程中的可变点^[8-9]、特征模型^[10]或者功能模块^[11]等配置项进行选择与设计,从而实现

基金项目:国家重点研发计划(2019YFB2102100);国家自然科学基金(62072146);浙江省重点研发计划(2019C03135, 2019C03134)

This work was supported by the National Key Research and Development Program of China(2019YFB2102100), National Natural Science Foundation of China(62072146) and Key Technology Research and Development Program of Zhejiang Province(2019C03135, 2019C03134).

通信作者:任永坚(yongjian.ren@hdu.edu.cn)

流程个性化定制。然而,该方法通常需要将配置项转换为可执行的业务流程,增加了流程配置过程中的操作,降低了定制开发效率。同时,当租户的业务需求发生变化时,现有的方案需要对模型进行重构^[12],增加了定制操作步骤,因此会延长系统响应时间,影响系统的及时性。

为了解决上述问题,本文提出一种支持多租户模式的业务流程动态定制模型(Dynamic Customization Model for Supporting Multi-tenant,DCMSM),能够基于差异性业务需求派生出不同的流程实例,提高了流程定制效率,同时任务节点的复用能有效减少需求变更时模型的调整时间。本文的主要贡献如下:

(1)采用可变性任务节点组装的方式构建业务子流程,通过租户感知器获取租户配置文件信息,从而选择租户所需的任务节点,派生具体流程实例;

(2)设计业务流程动态定制方法,通过模式匹配器计算任务节点的匹配度,进而与流程分支进行匹配,动态响应租户需求变化;

(3)选取高校中的实际业务流程对模型进行实验评估,验证了模型的有效性。

本文第2节介绍业务流程定制的相关工作;第3节详细描述支持多租户模式的业务流程动态定制模型的设计细节;第4节通过实例对模型进行评估;最后总结全文。

2 相关工作

根据使用的方法,现有的业务流程定制方案可以分为两类。

(1)流程模板技术。流程模板是多个租户共有的,能够被定制、调整为可执行实例的流程片段^[13]。将基于模板的流程定制方式与自动流程定制方式进行对比发现,当预定义的流程片段被频繁使用时,使用模板方式进行定制更占优势^[14]。采用流程模板进行定制的一种做法是使用形式化语言融合技术^[15]将两个现有的业务流程组合在一起,以达到流程定制的目的。该方法通过扩展的临时逻辑规则语言^[16-17]对业务流程进行形式化编码,然后将两个流程的编码集合进行融合以

重构新的流程,重构后的流程同时保留了原始的流程特征,从而消除模板流程变化后对定制流程产生的影响。这种方法能适应业务需求的变化,但增加了形式化编码和融合操作两个中间步骤。基于流程模板进行定制的另一种做法是将定制操作限定在一定的约束范围内^[18],包括节点可添加范围、节点依赖规则以及节点是否可删除等,然后通过对流程模板进行定制,产生个性化流程。该方法虽然能定制出符合不同业务需求的流程,但设置约束范围时需要考虑较多的情况,从而影响开发效率。对于以上方法存在的缺陷,进一步的研究提出使用扩展的BPMN^[19]语言——流程模板定制语言(PTCL)来进行流程模板模型设计^[20]。在PTCL中添加了增加、删除和修改3种标签来标记BPMN元素的可定制操作,通过读取流程模板模型和流程定制模型来自动产生流程实例模型。该方法一定程度上提高了流程的开发和维护效率,但在模型结构发生扩展时需要模板进行重新设计。

(2)流程配置技术。较早的解决方案^[21]通过静态或动态的业务流程选择,以及对流程实例的调整来生成定制流程。该方案以选择和调整方式相结合的方法提高了业务流程的使用灵活性,然而却需要开发额外的功能和工具以支持流程的动态调整。基于可变点的选择方法^[22-23]通过对活动、网关或者事件进行选择,生成个性化定制流程,但可变点的设计方式会直接影响流程的维护和理解。

对于特征模型的选择,使用树型表示法描述特征之间的不同关系,并用BPFM模型^[24]表示所有可能的BP变体,通过配置得到最适合租户的业务流程。针对多租户在线学习应用场景,一种研究方法通过使用SPL^[25]管理多个租户不断变化的需求,定义了可重用和可配置的在线学习流程^[26]。然而这种方法需要将特征模型映射到可执行的流程片段,因此会增加流程定制的中间处理过程,一定程度上影响了开发效率。对于功能模块的选择,将业务流程从逻辑上分为业务层、服务层和数据层^[27],并分别设计能高效且独立执行的配置策略。该方法能满足用户灵活的业务需求,同时降低各层之间的耦合性,但无法处理单个任务节点需求发生变化的情况。

表1列出了现有研究方法的对比。

表1 现有研究方法对比
Table 1 Comparison of existing research approaches

Approach	Customization type	Customization method	Multi-tenant	Advantage	Insufficient
[15]	Template	Process merging, temporal logic	Not support	Eliminate modification influence	Additional operation steps, lack of flexibility
[18]	Template	Add constraint	Support	Version control, process evolution	Constraint setting, freedomless
[20]	Template	Extension of BPMN, process transformation	Not support	Automatic process transformation	Template redesign for extension
[21]	Configuration	Selection and adaption	Not support	Flexible	Develop additional functionality
[22-23]	Configuration	Variant management	Not support	Process fragment reuse	Model design influence
[26]	Configuration	Feature model selection	Support	Feature model reuse	Additional operation steps
[27]	Configuration	Layer based configuration	Support	Minimize the degree of layer coupling	Unable to handle the changed requirements of a single task node
DCMSM	Configuration	Variant selection, pattern matching	Support	Process fragment reuse, dynamics	Concurrency timeliness

3 业务流程动态定制模型设计方法

3.1 系统架构

业务流程动态定制模型的支持系统由5个模块构成:

业务流程模块、表单管理模块、租户感知模块、流程执行器和共享数据库,如图1所示。业务流程模块用于在线流程建模,并管理流程实例;表单管理模块用于创建和管理流程表单;租户感知模块用于判别租户身份,并在流程定制模型中派生

具体流程实例;流程执行器负责启动流程实例,并监视流程的运转情况;共享数据库用于存放需要持久保存的数据,如流程模型、流程表单和租户数据等。

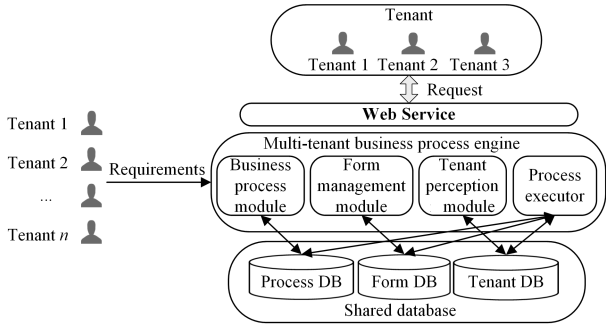


图1 业务流动态定制模型的支持系统架构图

Fig. 1 Supported system architecture diagram of dynamic business process customization model

3.2 多租户业务流程定制模型

BPaaS模式将业务流程以服务的形式为用户提供服务^[28-29]。在多租户应用场景下,租户需求存在一定差异,定制过程中需要分析租户共性需求和差异性需求,并基于差异性需求进行定制,产生定制流程^[30],如图2所示。

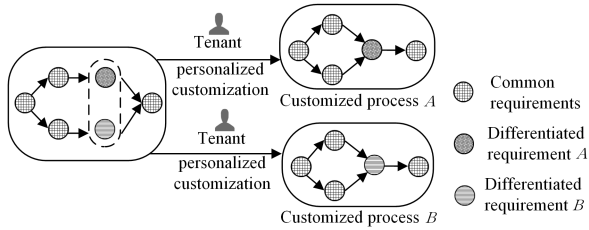


图2 基于BPaaS云服务模式的个性化流程定制

Fig. 2 Personalized process customization based on BPaaS cloud service mode

3.2.1 基本原理

传统软件模式^[31]需要分别为每个用户定制一套软件系统,开发成本较高。如图3(a)所示,软件系统部署在用户本地,3个用户所需的实例A,B,C均由相同流程定义对象产生,因此 $A=B=C$ 。然而,用户的业务需求存在差异,需要依据实际情况进行定制化开发,影响了系统的开发效率。

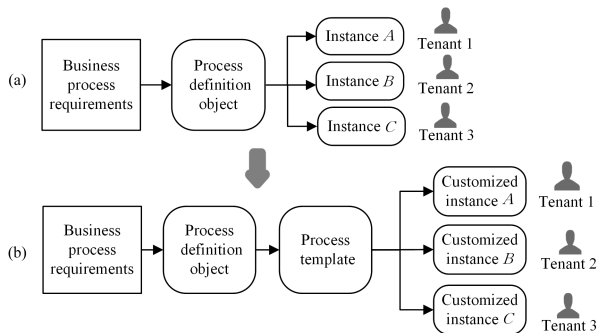


图3 传统软件模式与基于模板的SaaS软件模式

Fig. 3 Traditional software mode and template-based SaaS software mode

取而代之的是基于模板的SaaS软件模式^[32],如图3(b)所示,通过分析业务需求,将通用的流程片段作为模板提供给用户使用,用户基于模板进行定制,产生定制实例,此时 $A \neq$

$B \neq C$ 。然而,该模式并非开放式定制,定制范围受定制约束限制,同时需要考虑模板发生变化时对流程实例产生的影响。因此,该模式的开发效率并不是很高。

为了提高系统的灵活性,本文提出支持多租户模式的业务流动态定制模型(DCMSM),该模型通过复用租户共性流程节点并提供差异性节点的方式实现个性化流程服务。本文通过分析租户的共性需求和差异性需求,将任务节点划分为共性任务节点(记为 CT ,其中 $T=CT_1+CT_2+\dots+CT_n$)和可变性任务节点(记为 $VT_{11}, VT_{12}, \dots, VT_{ij}$, i 用于标识任务节点所属的子流程序号, j 用于标识任务节点序号),因此待开发的任务节点集记为 $T_{\text{总}}=CT+\sum_{i=1}^n VT_{ij}$ ($j=1, 2, \dots, m$)。然后描述不同租户的业务需求(即租户配置文件),用 T_i 表示租户需求对应的任务节点集,则 $T_i=CT+\sum_{i=1}^n VT_{ij}$ (j 为租户在第 i 个子流程中所需的任务节点序号)。租户基于模型进行定制,派生出不同的流程实例。如图4所示,该模型派生的流程实例 $A \neq B \neq C$ 。这种基于多租户差异性需求的BPaaS软件模式支持多租户多实例共存^[33]。与传统软件模式相比,租户间共用一套软件系统,在降低开发成本的同时,提高了流程定制开发效率和系统灵活性。

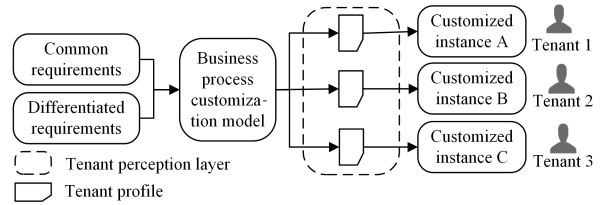


图4 基于租户差异性需求的BPaaS软件模式

Fig. 4 BPaaS software mode based on differentiated requirements of tenants

3.2.2 定制模型的开发框架

多租户业务流程定制模型开发框架如图5所示。向多租户业务流程定制模型中加入租户感知器(图中虚线圆),用于租户身份的判别,同时引入可变性设计(图中圆角矩形),支持租户对可变性任务节点进行选择。在流程运行时根据不同的租户身份获取租户配置信息,选择所需的任务节点,从而派生出不同的流程实例。

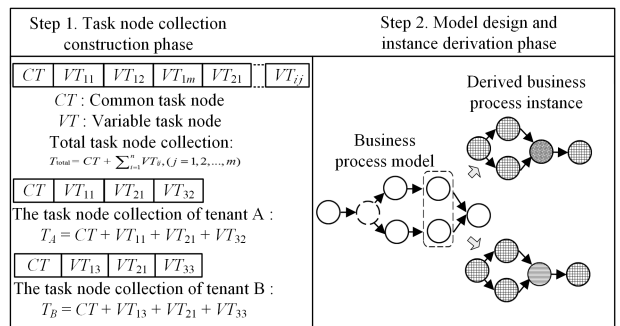


图5 多租户业务流程定制模型开发框架

Fig. 5 Development framework of multi-tenant business process customization model

(1)任务节点集构建阶段的主要任务是分析租户需求涉及的用户任务,按业务需求是否存在差异将任务节点划分为共性任务节点和可变性任务节点,建立待开发任务节点集;然后根据租户个性化业务需求,描述不同租户对任务节点的选择。

(2)模型设计和实例派生阶段的主要任务是按任务节点层级顺序设计业务流程模型,以可变性任务节点组装的方式构建业务子流程,向模型中添加租户感知器,进行租户身份判别,然后通过读取租户配置信息,派生出具体流程实例。

租户配置文件的描述信息如表 2 所列,其中 $TASK_ID$ 为租户在流程模型($PROCESS_DEF_ID$)的子流程($SUB_PROCESS_DEF_ID$)中所需的可变性任务节点。 $SUB_PROCESS_WEIGHT$ 和 $TASK_VALUE$ 分别为子流程权重和任务节点权值,用于根据租户选择的任务节点计算匹配度,动态调整后续业务流程。关于业务流程的动态定制方法将在 3.3 节进行说明。

表 2 租户配置文件信息描述表

Table 2 Tenant profile information description

Filed	Type
$TENANT_ID$	varchar(64)
$PROCESS_DEF_ID$	varchar(64)
$SUB_PROCESS_DEF_ID$	varchar(64)
$SUB_PROCESS_WEIGHT$	int(11)
$TASK_ID$	varchar(64)
$TASK_VALUE$	int(11)

租户感知器通过 Groovy 动态脚本^[34] UserGT 调用后端服务,通过获取环境上下文的用户信息感知租户身份。租户感知器的实现算法如算法 1 所示。

算法 1 租户感知算法

输入:无

输出:当前用户对应的租户 tenant

Begin:

```

1. /* 从线程和缓存中获取租户对象 */
2. tenant ← getCachedTenant(currentTenant)
3. if tenant = null then // 租户对象为空
4.   user ← getCurrentUser() // 获取当前用户
5.   /* 调用用户服务接口获取用户的 json 字符串 */
6.   data ← getJsonById(user)
7.   /* 获取参与者实体对象 */
8.   employeePo ← fromJsonToEntity(data)
9.   if is Not Empty(employeePo) // 实体对象非空
10.    /* 获取用户组的 json 字符串 */
11.    group ← entityToJson(employeePo)
12.    tenant ← getTenant(group) // 获取租户对象
13. else return null

```

14. return tenant

End

上述租户感知算法在进行租户身份判别时不会立即调用用户服务接口获取用户的 json 字符串,而是先查询保存于线程变量和本地缓存中的历史数据。若历史数据存有租户信息,则直接返回租户对象,否则获取当前用户信息并进行租户身份判别。参与者实体对象 employeePo 保存了参与者基本属性(用户信息、租户信息以及所在组织),在获取参与者实体对象后,通过调用 entityToJson 方法得到用户组 json 字符串,最后通过用户组获取租户对象。

3.2.3 多租户业务流程定制模型

一个完整的业务流程能反应租户对该流程从开始节点到结束节点的全部业务需求,包括涉及的人物角色、用户任务以及业务逻辑^[35]。本文提出的支持多租户模式的业务流程

定制模型(见图 6)将租户的差异性需求设计为多选一的可变性任务节点,通过获取租户配置信息来选择相应任务节点,派生出具体的流程实例。

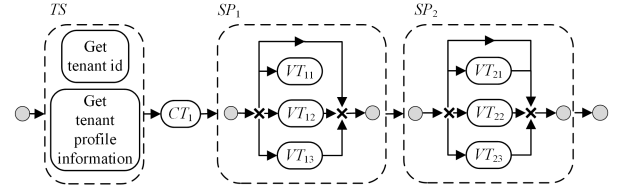


图 6 多租户业务流程定制模型

Fig. 6 Multi-tenant business process customization model

定义 1 多租户业务流程定制模型由一系列流程要素按照一定业务逻辑组成,用八元组 $MTBP = \{Pid, B, TS, D, CT, SP, R, P\}$ 表示,其中:

- (1) Pid 表示流程标识,用于区分不同的业务流程;
- (2) B 表示端节点, $B = \{start, end\}$, $start$ 节点控制流程实例创建, end 节点控制流程实例终止;
- (3) TS 表示租户感知器,是设置于 $start$ 节点之后的 Java 服务节点,用于租户身份的判别,并根据租户配置信息派生出具体流程实例;
- (4) D 表示网关节点, $D = \{d_1, d_2, \dots, d_n\}$,用于控制流程分支选择;
- (5) CT 表示共性任务节点, $CT = \{CT_1, CT_2, \dots, CT_n\}$,它是一组由相同的租户业务需求产生的任务节点;
- (6) SP 表示子流程,其主体是一组同级可变性任务节点,并包含一条不带有任务节点的顺序流;
- (7) R 表示资源,包括业务信息和资源数据,在流程实例化之前表示流程绑定的表单;
- (8) P 表示业务流程交互过程中的参与者。

定义 2 子流程 SP 由流程标识、端节点、网关节点、可变性任务节点、资源和参与者组成,表示为 $SP = \{Pid, B, D, VT, R, P\}$ 。其中, VT 表示可变性任务节点, $VT = \{VT_{i1}, VT_{i2}, \dots, VT_{ij}\}$ ($i=1, 2, \dots, n$, 表示子流程序号, $j=1, 2, \dots, m$, 表示每个子流程中的可变性任务节点序号)。

每个业务流程都有唯一的 Pid 标识,为模型中的子流程 SP 设定需要调用的子流程 Pid ,完成模型与子流程的绑定。

不同应用场景下,业务流程在流过程中表现出不同的执行路径,这些执行路径由基本流程结构构成。

定义 3 顺序结构指处理完一个节点后,顺序执行下一个节点,表示为 $SEQ = \{start, T_1, T_2, end\}$,即 T_1 执行完毕后,顺序执行 T_2 。

定义 4 回退结构指通过网关节点回退到之前的某个任务节点,表示为 $BACK = \{start, T_1, T_2, \dots, T_n, D, end\}$ 。

回退结构适用于对任务节点的处理结果未达到预期要求,需要返回到上级节点重新处理的情况。循环结构是回退结构的一种形式,通过控制回退次数能够避免流程出现死循环。

定义 5 分支结构是指通过网关节点从多个并行任务节点中选择其中一个执行,表示为 $FORK = \{start, D, \{T_1, T_2, \dots, T_n\}, end\}$ 。

通过流程定制模型派生流程实例的过程如下:首先,用户基于流程定制模型启动流程实例,同时租户感知器调用用户

服务 API 从环境上下文中获取参与者实体对象,并判断当前用户所属租户,将租户信息保存于线程局部变量 *Tenant* 中。然后,当流程流转到网关节点时,通过读取该变量值,获取租户配置信息中的任务节点。最后,从子流程 *SP* 中选择该节点,派生出符合租户需求的流程实例。

3.3 多租户业务流程动态定制方法

用户的业务需求不是一成不变的,业务流程需要根据实际情况进行调整^[36]。在上述业务流程定制模型的基础上,加入了动态定制方法,以适应租户需求变化的情况。引入模式匹配器(Pattern Matcher)和网关路由节点(Router,记为 *R*),用于动态调整后续业务流程。动态定制模型如图 7 所示。

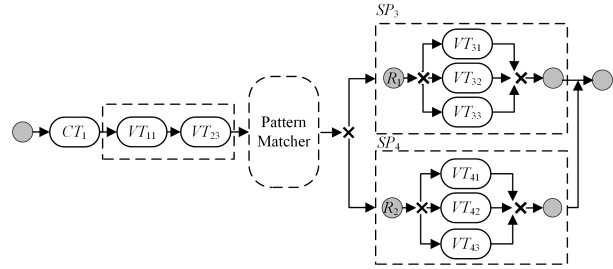


图 7 基于可变性任务节点的业务流程动态定制模型

Fig. 7 Dynamic customization model of business process based on variable task nodes

在多租户应用场景下,对于共性任务节点,每个租户都是相同的,不会对后续流程产生影响,因此只需考虑可变性任务节点。对于可变性任务节点,当选择了某个子流程中的一个任务节点后,就表示选择了该子流程。根据不同业务需求,将可变性任务节点所属的子流程与后续子流程进行映射并归类,为每一类别设计匹配条件,通过模式匹配器对已选的可变性任务节点与后续流程分支进行匹配。

模式匹配器根据已选的可变性任务节点计算匹配度,并将计算得到的匹配度与网关路由节点预设的模式进行匹配,从而选择相应的子流程。

匹配度的计算公式为:

$$P = \sum_{i=1}^n w_i * v_{ij}, j=1, 2, \dots, m \quad (1)$$

其中, w_i 为所选的第 i 个子流程的权重, $w_i = 10^{i-1}$; v_{ij} 为所选的第 i 个子流程中的第 j 个任务节点 VT_{ij} 对应的自然数权值, $v_{ij} = j$ (子流程中第一条顺序流不含任务节点,其对应的权值为 0)。

网关路由节点(如图 7 中的 R_1 和 R_2)用于标记不同的子流程。选用正则表达式作为选择网关路由节点的匹配模式。将计算得到的匹配度进行“归一化”处理,然后按预先分好的流程类别为每个网关路由节点设置不同的正则表达式。归一化过程为将匹配度中的非 0 值用 1 替代,0 值不做处理。归一化后得到的是用 0 和 1 表示的字符串,设置的正则表达式为形如“. *11”的字符串。因为设置的子流程权重是以 10 的倍率递增的,所以归一化处理后,从低位到高位分析字符串的每一位,第 i 位的值为 1,表示选择了第 i 个子流程中的某个可变性任务节点,从而与相应流程类别的正则表达式进行匹配。模式匹配算法如算法 2 所示。

算法 2 模式匹配算法

输入:租户 ID,流程模型 ID

输出:原始匹配度和归一化匹配度

Begin:

1. $tenant \leftarrow createTenant()$ //创建租户实例
2. $matcher \leftarrow 0$ //初始化匹配度
3. $process_weight \leftarrow query(SUB_PROCESS_WEIGHT)$ //获取子流程权重
4. $task_value \leftarrow query(TASK_VALUE)$ //获取任务节点权值
5. $matcher \leftarrow matcher + process_weight * task_value$
6. $setOrUpdate(tenant, matcher)$ //设置匹配度
7. 重复步骤 3—步骤 6,计算模式匹配度
8. for i to $length(matcher)$ by 1 do //循环处理匹配度中的每一位
9. $updateMatcher(matcher)$ //更新匹配度值,将非 0 值用 1 代替,得到归一化匹配度

End

通过计算得到的匹配度进行网关路由节点下的分支任务节点选择,选择过程为:由低位到高位分析匹配度数值,第 i 位的值为 j ,表示租户选择了第 i 个子流程的第 j 个任务节点(例如有一流程模型包含可选子流程 SP_1, SP_2, SP_3 ,计算得到匹配度值为 203,则表示选择了 SP_1 中的第 3 个任务节点和 SP_3 中的第 2 个任务节点,而 SP_2 中的任务节点都没有被选择,即跳过执行),然后匹配分支正则表达式,从而选择对应的分支任务节点。

租户配置信息中的子流程和任务节点在共享数据库中用二元组 $\{SP_i, VT_{ij}\}$ 表示(其中 VT_{ij} 表示租户在第 i 个子流程中选择第 j 个任务节点)。若租户未选择该子流程中的任务节点,则表示为 $\{SP_i, None\}$ 。流程定制过程中,定制操作发生于可变性任务节点,本文提出的流程动态定制方法采取 3 种定制手段:节点新增、节点替换以及节点删除。

定义 6 节点新增方法是在原模型的子流程中添加新的可变性节点,包括新增单个节点与新增多个节点。

节点新增过程将二元组 $\{SP_i, None\}$ 更改为 $\{SP_i, VT_{ij}\}$,其中,元素 VT_{ij} 表示待添加节点,当需要添加多个节点时,该元素表示含多个任务节点的子流程节点 SP_j 。

定义 7 节点替换方法将原模型中的可变性节点用新节点代替,包括一对一替换、一对多替换以及多对多替换。

一对一替换用新二元组 $\{SP_i, VT_{ij}\}$ 取代原二元组 $\{SP_i, VT_{ik}\}$,其中 VT_{ij} 为替换节点, VT_{ik} 为待替换节点;进行一对多替换时, VT_{ij} 表示含多个任务节点的子流程节点 SP_j ;进行多对多替换时, VT_{ij} 与 VT_{ik} 均表示含多个任务节点的子流程节点。

定义 8 节点删除方法是在原模型的子流程中删除可变性节点,包括删除单个节点与删除多个节点。

节点删除过程将二元组 $\{SP_i, VT_{ij}\}$ 替换为 $\{SP_i, None\}$,元素 VT_{ij} 表示待删除节点,当需要删除多个节点时,该元素表示含多个任务节点的子流程节点 SP_j 。

流程定制算法如算法 3 所示。

算法 3 流程定制算法

输入:当前租户 $tenant$

输出:流程定制模型 BPModel

Begin:

1. $BPDef \leftarrow obtainBP()$ //获取流程定义
2. $BpProfile \leftarrow queryProfile(tenant, BPDef)$ //查询配置信息
3. $BpProfile \leftarrow addProfileNode(SP_i, None, VT_{ij})$ //新增单个节点
4. $SP_j \leftarrow setSPNodes()$ //设置含多节点的子流程节点

- 5. /* 添加含多节点的子流程节点 */
 - 6. BpProfile←addProfileSPNode(SP₁, None, SP₃)
 - 7. /* 更新流程定制模型 */
 - 8. BpModel←customModel(tenant, BpProfile)
 - 9. BpProfile←repProfileNode(SP₁, VT_{ik}, VT_{ij}) // VT_{ij}取代 VT_{ik}
 - 10. /* 用多节点 SP₃取代 VT_{ik},实现一对多替换 */
 - 11. BpProfile←repProfileSPNode(SP₁, VT_{ik}, SP₃) // SP₃取代 VT_{ik}
 - 12. BpProfile←delProfileNode(SP₁, VT_{ik}, None) //删除节点 VT_{ik}
 - 13. BpModel←customModel(tenant, BpProfile) //更新
- End

4 实验评估

将本文提出的支持多租户模式的业务流程动态定制模型

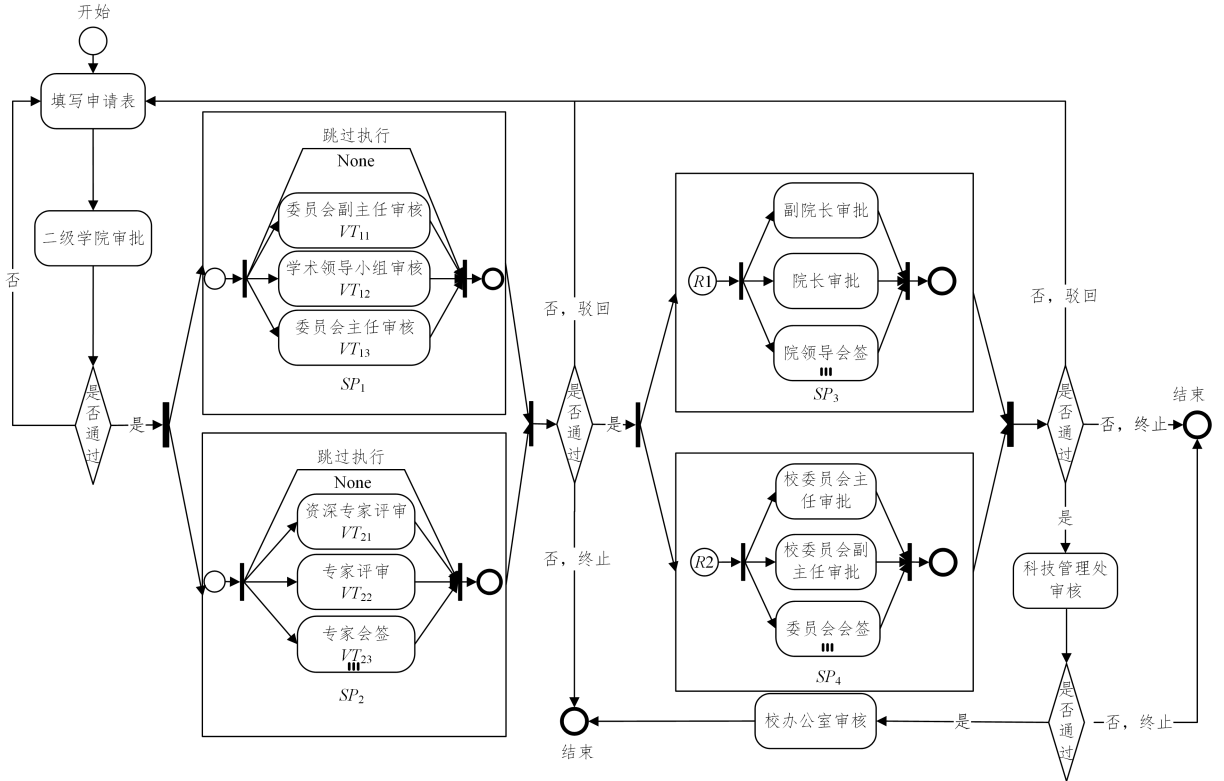


图 8 多租户科研项目申报审批流程

Fig. 8 Scientific research project approval process supporting multi-tenant

通过对需求进行分析,该流程包含 8 个活动,其中,“二级学院学术委员会审核”和“校外专家评审”为可选的子流程,供不同租户根据实际业务需求进行选择,而“院领导审批”和“校学术委员会审批”为两种流程结果。将上述 4 个环节分别用子流程 SP₁—SP₄表示。

在上述流程中,若用户选择了“校外专家评审”子流程中的某个任务节点,则后续流程必须流转到“校学术委员会审批”子流程,否则流转到“院领导审批”子流程,根据这两种情况为网关路由由节点 R₁和 R₂设置的正则表达式分别为“. * 1”和“1.”,并且根据子流程 SP₁和 SP₂中的任务节点与后续子流程 SP₃和 SP₄中的任务节点的匹配情况,设置 R₁和 R₂下的分支正则表达式从上到下分别为“. * 2”“. * 1”“. * 3”和“1.”“2.”“3.”。

现有 A, B, C 3 个租户,针对图 8 所示的科研项目申报审批流程,有不同的业务需求。租户 A 在二级学院审批完成后需要提交至二级学院学术委员会主任审核;租户 B 的流程

应用于高校流程审批系统中,并选取有代表性的案例进行实验。通过与传统定制方式进行对比,验证所提模型的有效性。

4.1 实施案例及可变更性节点构建

校级科研项目申报审批流程为:申请人填写项目申请表并报送二级学院,由二级学院审核员对项目内容进行形式审查和申请人资格审核,审核通过则报送二级学院学术委员会审核或校外专家评审;然后报送院领导或者校学术委员会审批,再提交至科技管理处和校长办公室审核;最终审批通过后,签订任务合同书,进行项目立项。在上述过程中,二级学院学术委员会审核和校外专家评审为租户可选情况,当选择不同时,后续的业务流程也应该有所改变,根据该业务场景构建如图 8 所示的业务流程图。

相对比较复杂,需同时由学术领导小组审核和专家会签后,流程才能流向下一个审批节点;租户 C 则需要提交专家评审。因此,租户 A 的配置信息包含元组 {SP₁, VT₁₃}, {SP₂, None}, 租户 B 的配置信息包含元组 {SP₁, VT₁₂}, {SP₂, VT₂₃}, 租户 C 的配置信息包含元组 {SP₁, None}, {SP₂, VT₂₂}。同时,针对租户的不同业务需求,后续的业务流程也存在相应变化,流程将流向不同的审批节点。

租户启动流程后,模式匹配器计算租户 A 的匹配度为 3,归一化匹配度为 1;租户 B 的匹配度为 32,归一化匹配度为 11;租户 C 的匹配度为 20,归一化匹配度为 10。

将匹配度与设置的正则表达式进行匹配,则租户 A 将匹配到 R₁下的院领导会签,租户 B 匹配到 R₂下的委员会会签,租户 C 匹配到 R₂下的校委员会副主任审批。因此租户 A 派生的流程实例为:填写申请表→二级学院审批→委员会主任审核→院领导会签→科技管理处审核→校办公室审核;租户 B 派生的流程实例为:填写申请表→二级学院审批→学术

领导小组审核、专家会签→委员会会签→科技管理处审核→校办公室审核;租户 C 派生的流程实例为:填写申请表→二级学院审批→专家评审→校委员会副主任审批→科技管理处审核→校办公室审核。

当租户的业务需求发生变化时,例如租户 A 在二级学院审批完成后需要增加校外专家评审环节,则修改租户 A 配置信息的元组为 $\{SP_1, VT_{13}\}, \{SP_2, VT_{22}\}$ 。当租户需要再次启动流程时,模式匹配器将实时计算出租户 A 的匹配度为 23,归一化后的匹配度为 11,此时将匹配到 R_2 下的校委员会主任审批节点。因此,租户 A 需求变更后派生的流程实例为:填写申请表→二级学院审批→委员会主任审核、专家评审→校委员会主任审批→科技管理处审核→校办公室审核。

4.2 模型的性能评估

DCMSM 模型的核心组件为租户感知器和模式匹配器,通过调用核心算法实现流程动态定制。对于某流程 P ,假设其子流程数为 n ,用 T_1, T_2, T_3 分别表示租户感知算法、流程定制算法以及模式匹配算法的时间开销。租户感知算法通过查询线程变量、缓存变量或者调用用户服务接口实现租户身份判别,其时间复杂度 $T_1 = O(1)$;流程定制算法通过替换节点元素实现流程定制,其时间复杂度 $T_2 = O(1)$;模式匹配算法通过循环获取元组信息计算匹配度,其时间复杂度 $T_3 = O(n)$ 。对比 T_1, T_2 与 T_3 的时间开销,得出 DCMSM 模型的总体时间复杂度为 $O(n)$ 。

用 P_1, P_2, P_3 分别表示租户感知算法、流程定制算法、模式匹配算法的空间开销。租户感知算法定义了用于存储租户对象的临时变量,其空间复杂度 $P_1 = O(1)$;流程定制算法对节点元素进行替换,仅定义了临时节点变量,其空间复杂度 $P_2 = O(1)$;模式匹配算法定义了 3 个与匹配度相关的变量,其空间复杂度 $P_3 = O(1)$ 。对比 P_1, P_2 与 P_3 的空间开销,得出 DCMSM 模型的总体空间复杂度为 $O(1)$ 。

为进一步评估模型的定制效率,将其与传统模式下针对多租户独立定制的方式进行对比。从 3 所高校中总共收集了 207 个业务流程,并随机选取 8 个流程进行实验评估(实验环境配置见表 3)。从流程初始配置、执行和调整的时间开销几个方面评估 DCMSM 模型的性能。

表 3 实验环境配置

Table 3 Experimental environment configuration

Configuration item	Parameter value
CPU/GHz	2.80 * 2.81
Memory/GB	16
Disk/TB	1
Operating system/bit	Windows10-64

初始配置时间指从开始构建任务节点到流程能够运行的时间开销。在构建多租户业务流程模型的过程中,占配置耗时较高的两个操作为:配置任务节点和配置网关路由节点分支判断条件。忽略构建顺序流和子流程的时间开销,因为在构建下一个任务节点时会自动产生顺序流,而对于子流程的构建,只须设计节点框而无须配置额外信息。将配置单个任务节点耗费的时间记为 t_1 ,配置网关路由节点分支判断条件的的时间记为 t_2 。记录采用 DCMSM 模型配置 8 组业务流程模型中的共性任务节点数 CT 、可变性任务节点数 VT 以及网关路由节点数 R ,并计算 DCMSM 模型所需的配置时间,如表 4 所列。

表 4 DCMSM 模型的节点数和配置时间

Table 4 Node number and configuration time of DCMSM model

Business process	CT	VT	R	DCMSM
BP1	6	8	2	$14t_1 + 2t_2$
BP2	5	7	2	$12t_1 + 2t_2$
BP3	2	3	0	$5t_1$
BP4	3	6	0	$9t_1$
BP5	3	10	3	$13t_1 + 3t_2$
BP6	4	9	2	$13t_1 + 2t_2$
BP7	2	5	0	$7t_1$
BP8	6	9	2	$15t_1 + 2t_2$

同时,记录传统模式下为 3 所高校(A,B,C)独立配置流程时分别所需的任务节点数,并计算总时间,如表 5 所列。

表 5 传统模式的节点数和配置时间

Table 5 Node number and configuration time for traditional mode

Business Process	A	B	C	Total
BP1	9	8	9	$26t_1$
BP2	8	8	7	$23t_1$
BP3	3	3	3	$9t_1$
BP4	5	4	5	$14t_1$
BP5	7	7	6	$20t_1$
BP6	7	6	7	$20t_1$
BP7	4	3	4	$11t_1$
BP8	9	8	9	$26t_1$

随机选取 20 个任务节点和网关路由节点分别进行测试实验,计算单个任务节点的平均配置耗时 t_1 为 12 s,网关路由节点的平均配置耗时 t_2 为 18 s (t_2 为 1.5 倍 t_1),得到 DCMSM 模型与传统模式的配置时间对比,如图 9 所示。从图中可见,采用 DCMSM 模型的配置时间要少于传统模式为租户独立配置流程的总时间,任务节点的复用提高了多租户场景下的流程定制开发效率。

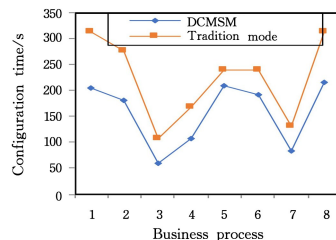


图 9 DCMSM 模型与传统模式的配置时间对比

Fig. 9 Comparison of configuration time between DCMSM and traditional models

图 10 给出了采用 DCMSM 模型和传统方式的流程执行时间对比。流程的执行时间指从开启流程实例到流程结束的运行时间。从图中可以看出两者的执行时间相当,因为流程执行时间的长短主要取决于流程执行过程中的任务节点被调用和执行的时间。

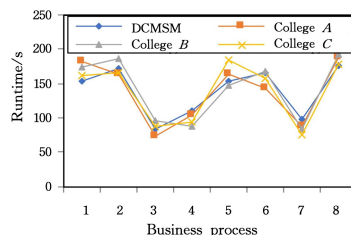


图 10 DCMSM 模型与传统模式的流程执行时间对比

Fig. 10 Comparison of process execution time between DCMSM and traditional models

调整时间指从开始调整流程模型到流程能够正常执行的时间开销。当租户需求发生变化时,DCMSM 模型通过调用后端服务,替换租户配置信息中的节点元素,实现节点定制。同时,计算任务节点匹配度与后续流程分支进行匹配,动态产生业务流程。该模型用租户配置信息空间开销换取更短的时间开销。对于 DCMSM 模型,其调整时间相当于一个常规操作时间 T ;而对于传统模式,需要对模型中的任务节点进行调整,其调整时间相当于一个复杂操作时间 M ,而 T 远小于 M 。图 11 给出了当租户需求发生变化时,调整 DCMSM 模型所需的时间与传统调整任务节点方式的调整时间对比,从图中可见,DCMSM 模型的调整时间远少于传统模式的调整时间,提高了业务流程系统的灵活性。

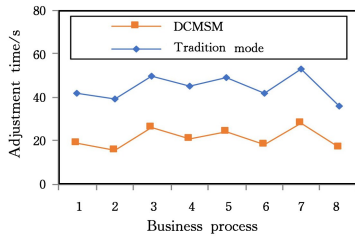


图 11 需求变更时 DCMSM 模型与传统模式调整时间对比
Fig. 11 Comparison of adjustment time between DCMSM and traditional models with changing demands

结束语 本文主要探讨如何在多租户应用场景下设计高效的业务流程定制方法。基于对流程个性化的思考,提出一种支持多租户模式的业务流程动态定制模型。首先,通过分析多个租户的业务需求,将任务节点划分为共性任务节点和可变性任务节点,采用节点组装的方式构建业务子流程;其次,设计了用于租户身份判别的租户感知器,能基于业务流程定制模型派生出不同的流程实例;然后,提供了一种流程动态定制方法,能够及时响应租户需求变更。然而,在本文的工作中并未针对高并发情况进行响应时间的优化。在未来的工作中,我们将进一步研究如何在并发用户量非常庞大时缩短系统的响应时间。

参考文献

- [1] XIE L, XU L, DE VRIEZE P. Lightweight business process modelling[C]// Proceedings of the International Conference on E-Business and E-Government. Washington: IEEE Computer Society, 2010: 183-186.
- [2] WOITSCH R, UTZ W. Business Processes as a Service (BPaaS): A model-based approach to align business with cloud offerings [C]// eChallenges e-2015 Conference Proceedings. New York: IEEE, 2015: 1-8.
- [3] KHAN K M, NHLABATSI A, KHAN N. A process model for customisation of software in multi-tenant SaaS model [C]// IEEE/ACM 8th International Conference on Utility and Cloud Computing. New York: IEEE, 2015: 418-419.
- [4] PAN H, HUANG Z, HE Z. Research on process customization technology for intelligent transportation cloud service platform [J]. Cluster Computing, 2019, 22(6): 15245-15254.
- [5] GUO Y B, CAI Z Y, LIN Z W, et al. Design and implementation of light-weighted dynamic workflow engine[J]. Computer Engineering and Design, 2014, 35(3): 1105-1109.
- [6] ZHOU L, CAO J, CHEN J J. Self-envolving for process model of software as a service[J]. Computer Integrated Manufacturing Systems, 2011, 17(8): 1603-1608.
- [7] WEN J, XU B, BU F, et al. A service composition model based on business process template[C]// IEEE International Conference on Progress in Informatics and Computing. New York: IEEE, 2010: 1029-1033.
- [8] PUHLMANN F, SCHNIEDERS A, WEILAND J, et al. Variability mechanisms for process models[J]. PESOA-Report TR, 2005, 17: 10-61.
- [9] DELGADO A, CALEGARI D. BPMN 2.0 based modeling and customization of variants in business process families[C]// the 43rd Latin American Computer Conference. New York: IEEE, 2017: 1-9.
- [10] MONTERO I, PENA J, RUIZ-CORTES A. From feature models to business processes[C]// IEEE International Conference on Services Computing. IEEE, 2008: 605-608.
- [11] LIU Q, CHEN X, CHEN X D, et al. Mechanism of customization for application service based on ASP mode[J]. Computer Integrated Manufacturing Systems, 2007, 13(5): 1035-1040.
- [12] YIN J, LUO Z, LI Y, et al. Service pattern: An integrated business process model for modern service industry [J]. IEEE Transactions on Services Computing, 2016, 10(6): 841-853.
- [13] ILAHI L, MARTINHO R, GHANNOUCHI S A, et al. Towards a Business Process Management Governance approach using process model templates and flexibility[C]// IEEE World Congress on Services. New York: IEEE, 2016: 27-34.
- [14] GONZÁLEZ F, SILVESTRE L, BASTARRICA M C, et al. Template-based vs. Automatic process tailoring[C]// the 33rd International Conference of the Chilean Computer Science Society. New York: IEEE, 2014: 124-127.
- [15] BULANOV P, LAZOVIK A, AIELLO M. Business process customization using process merging techniques[C]// IEEE International Conference on Service-Oriented Computing and Applications. New York: IEEE, 2011: 1-4.
- [16] SARNO R, NURLAILI A L. Determining linear temporal logic formula for decomposed process model[C]// International Conference on Information and Communications Technology. New York: IEEE, 2018: 466-470.
- [17] SHI Y L, LUAN S, LI Q Z, et al. TLA Based Customization and Verification Mechanism of Business Process for SaaS[J]. Chinese Journal of Computers, 2010, 33(11): 2055-2067.
- [18] CHEN D, LI Q, KONG L. Process customization framework in SaaS applications[C]// the 10th Web Information System and Application Conference. New York: IEEE, 2013: 471-474.
- [19] GEIGER M, HARRER S, LENHARD J, et al. BPMN 2.0: The state of support and implementation [J]. Future Generation Computer Systems, 2018, 80: 250-262.
- [20] CUI X. An approach implementing template-based process development on BPMN[C]// the 16th IEEE/ACIS International Conference on Computer and Information Science. New York: IEEE, 2017: 239-244.
- [21] HEINL P, HORN S, JABLONSKI S, et al. A comprehensive approach to flexibility in workflow management systems[J].

- ACM SIGSOFT Software Engineering Notes, 1999,24(2): 79-88.
- [22] SBAI H, FREDJ M, KJIRI L. To trace and guide evolution in configurable process models[C]//Proceedings of IEEE/ACS International Conference on Computer Systems and Applications. Washington:IEEE Computer Society, 2013:1-4.
- [23] AYORA C, TORRES V, PELECHANOV, et al. Applying CVL to business process variability management[C]//Proceedings of the Variability for You Workshop: Variability Modeling Made Useful for Everyone. New York:ACM, 2012:26-31.
- [24] COGNINI R, CORRADINI F, POLINI A, et al. Business process feature model: an approach to deal with variability of business processes[M]// Domain-Specific Conceptual Modeling. Berlin: Springer, 2016:171-194.
- [25] SINNEMA M, DEELSTRA S, NIJHUIS J, et al. Covamof: A framework for modeling variability in software product families [C]//Proceedings of Software Product Lines. Berlin: Springer, 2004:197-213.
- [26] AZOUZI S, BRAHMI Z, GHANNOUCHI S A. Customization of multi-tenant learning process as a service with business process feature model[J]. Procedia Computer Science, 2018, 126:606-615.
- [27] HAN C, WU C, GONG W. Research on SaaS oriented business process customization method[C]// the 9th International Symposium on Computational Intelligence and Design. New York: IEEE, 2016:412-416.
- [28] BARTON T, SEEL C. Business process as a service-status and architecture[J]. Enterprise Modelling and Information Systems Architectures, 2014, 234:145-158.
- [29] BOURNE S, SZABO C, SHENG Q Z. Transactional behavior verification in business process as a service configuration[J]. IEEE Transactions on Services Computing, 2017, 12(2): 290-303.
- [30] KAPURUGE M, COLMAN A, HAN J. Achieving multi-tenant business processes in SaaS applications[C]// International Conference on Web Information Systems Engineering. Berlin: Springer, 2011:143-157.
- [31] BIBI S, KATSAROS D, BOZANIS P. Business application acquisition: On-premise or SaaS-based solutions? [J]. IEEE Software, 2012, 29(3):86-93.
- [32] CHEN W, SHEN B, QI Z. Template-based business logic customization for SaaS applications[C]// IEEE International Conference on Progress in Informatics and Computing. New York: IEEE, 2010, 1:584-588.
- [33] YONGSIRIWIT K, ASSY N, GAALOUL W. A semantic framework for configurable business process as a service in the cloud [J]. Journal of Network and Computer Applications, 2016, 59: 168-184.
- [34] HUANG Z C, HE C. Groovy service: on-demand Web service by script language[C]// IEEE International Workshop on Service-Oriented System Engineering. New York:IEEE, 2005:105-110.
- [35] ZHU J, TANG Y. Study of the Role and Task - based Access Control Technology for CSCW System[J]. Computer Science, 2010, 37(7):130-133.
- [36] RUSINAITE T, VASILECASO. Dynamic business process model implementation solutions [C] // Open Conference of Electrical, Electronic and Information Sciences. New York: IEEE, 2015:1-4.



ZHANG Ji-lin, born in 1980, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include mass data storage and processing, complex system modeling and simulation.



REN Yong-jian, born in 1963, Ph. D, professor. His main research interests include cloud computing, cloud storage and disaster recovery backup technology.