

面向服务集成的 workflow 模型及其实现方法

张型龙*, 李松犁, 肖俊超

(中国科学院软件研究所 互联网软件技术实验室, 北京 100080)

(* 通信作者电子邮箱 xinglong@nfs.iscas.ac.cn)

摘 要: 针对现有 workflow 系统集成已有软件服务时存在的两个问题: 1) 对于业务流程中所集成服务的描述信息不足, 无法满足服务集成需要; 2) 流程定制时可选择节点类型少, 节点不具有业务含义, 导致其定制过程复杂; 提出一种新的 workflow 模型。首先, 通过分析服务集成环境下的实际业务流程确定了模型的三部分信息: 结构信息、服务信息和人员信息; 其次, 将模型中各部分信息与 JPD L 作了对应说明, 表明了模型的完备性; 最后, 详细介绍了三部分信息的关键要素。实验结果表明, 基于 35 种服务可以快速搭建出满足实际业务需求的 48 种业务流程。模型中丰富的服务信息为流程运行提供了保障, 流程运行正确率 100%; 流程定制更加简便快捷, 流程从定制到运行平均时间小于 2 min。通过该模型可以快速地基于已有服务构建新的业务流程, 降低了为新需求开发软件的成本。

关键词: 服务集成; workflow 模型; 模型实现; workflow 语言; 流程翻译

中图分类号: TP319; TP311.52 **文献标志码:** A

Service integration-oriented workflow model and implementation method

ZHANG Xinglong*, LI Songli, XIAO Junchao

(Laboratory for Internet Software Technologies, Institute of Software Chinese Academy of Science, Beijing 100080, China)

Abstract: Since there are two problems when using current workflow to integrate existing software services: 1) Information of the integrated service is insufficient to satisfy the service integrating needs; 2) Only few node types without any business meaning can be chosen during process customization, which complicates the process customization. Thus, a new workflow model was proposed. Firstly, three information parts of the workflow model: structure information, service information and people information were determined by analyzing actual business processes under service integration environment; secondly, the corresponding description between three information parts and JPD L (JBoss jBPM Process Definition Language) was given to prove the completeness of the new workflow model; finally, the key elements of each information part were described. The experimental results show that 48 business processes can be quickly built and met the actual business needs based on 35 services. The rich service information provides guarantee for the process operation and the execution correct rate is 100%; the process customization is more convenient with less than 2 min from process customization to execution. The results show that the proposed model can help to build new business process quickly based on existing software services and save the software development costs.

Key words: service integration; workflow model; model implementation; workflow language; process translation

0 引言

企业发展过程中, 随着业务的扩展, 应用软件不断增加。这些软件大部分是来自企业外部用于解决特定业务需求的第三方工具。第三方工具可以通过 API (Application Program Interface) 对外提供服务。通过集成这些服务来满足新的业务需求, 节约开发软件的成本。

workflow 系统的作用是“使在多个参与者之间按照某种预定义的规则传递文档、信息或任务的过程自动进行, 从而实现某个预期的业务目标, 或者促使此目标的实现”^[1]。在作服务集成时, workflow 系统可将业务流程的任务关联到第三方工具的具体服务, 通过业务流程的执行满足业务需求。

由于第三方工具提供的服务种类多, 调用方式不统一, 在使用现有 workflow 系统进行业务流程定制时存在两个问题。

首先, 对业务流程中所集成服务的描述信息不足, 缺乏服务地址、调用方式及所需参数等信息, 无法满足服务集成需要。在第三方工具服务集成环境下定制业务流程时, 每个节点关联的服务来自系统外部, 需要丰富的服务描述信息来支持 workflow 的定制与执行。现有 workflow 系统所支持的业务流程大多基于系统内部功能点之间的串接。比如请假流程, 其中每一个节点对应的功能都位于系统内部。此时 workflow 系统的重点是流程结构信息存储以及节点状态转变, 而对于节点所关联的服务信息描述较少。

其次, 业务流程定制时可选择节点类型少, 节点本身不能

收稿日期: 2015-01-21; 修回日期: 2015-03-12。

基金项目: 国家自然科学基金资助项目(91318301, 91218302, 61432001); 国家 863 计划项目(2012AA011206)。

作者简介: 张型龙(1990-), 男, 山东东平人, 硕士研究生, 主要研究方向: workflow 模型; 李松犁(1990-), 男, 重庆万州人, 硕士研究生, 主要研究方向: 服务集成; 肖俊超(1978-), 男, 吉林珲春人, 副研究员, 博士, CCF 会员, 主要研究方向: 软件过程管理、workflow 管理、资源优化和调度。

表达业务含义,当通过有限种类节点来映射大量服务并定制业务流程时,定制过程复杂。以 JBPM(Java Business Process Management)为例,其流程节点类型主要是状态、任务和分支结构,节点本身不能表达业务含义。通过有限种类节点映射大量服务,流程定制过程会繁琐复杂。如果节点种类较多且能够表达一定的业务含义,工作流的定制将更加简便,业务流程更易理解。但是基于现有 workflows 系统修改或者拓展节点困难。

在工作流研究过程中,研究者们提出了一些经典工作流模型^[2]。Sadiq 等^[3]使用节点和变迁这两类符号来描述业务过程;李红臣等^[4]提出了基于组织、数据和过程的三维工作流模型;Bajaj 等^[5]提出了基于状态、实体和活动的 workflow 模型。其他工作流模型还包括:基于随机拓展 Petri 网^[6]、跨企业工作流模型^[7]、基于 Petri 网和实例迁移^[8]、基于多维工作流网络^[9]和基于着色 Petri 网^[10]。以上工作流模型有些侧重于对工作流语义的分析,有些偏重于工作流模型的实现。但无法解决服务信息描述少和流程定制复杂问题,不适合服务集成环境下的业务流程定制。

针对服务集成环境,也有一些工作流方面的研究工作。程俊^[11]提出了基于 Petri 网的服务集成研究方案;程强^[12]提出了服务集成环境下的工作流事务模型;Rizescu^[13]提出了工作流与 Web 服务集成的思路。但上述方法和模型并未解决服务集成环境下业务流程定制的需要。

本文通过分析服务集成环境下的实际业务流程,提出一种由结构信息、服务信息和人员信息组成的工作流模型。该模型可以有效地克服上文提到的服务描述信息不足,节点类型单一及流程定制复杂问题。在介绍模型的过程中,引入了 JPDL(JBoss jBPM Process Definition Language)用于描述各部分在业务流程中的作用。本文给出了模型的实现方案并结合实际项目验证了模型的有效性。

1 面向服务集成的工作流模型

在服务集成环境下定制业务流程包括 3 个步骤:首先,要确定流程结构,明确流程中的任务数量及执行顺序;其次,流程中的节点要关联到具体的服务;最后,分配人员参与流程中具体任务。由上可知,业务流程主要包含三部分信息:结构信息、服务信息和人员信息。结构信息用于表示业务过程结构中的各种实体及其约束关系;服务信息用于表示业务过程所涉及服务实体的内容;人员信息是对一个组织内各种实体及其相互关系的描述。

基于三部分信息本文提出了一种工作流模型,并通过工作流语言 JPDL 用以描述模型中各部分的作用。工作流模型中各部分内容与 JPDL 映射关系如图 1。

三部分相辅相成,服务信息和人员信息依赖于结构信息进行呈现;人员信息通过结构信息关联到具体服务信息。

1.1 结构信息

每一个业务都有许多任务组成,任务的执行存在顺序关系,如先后关系和并列关系等。图 1 所示业务流程中圆圈表示状态,方框表示任务,有向线段表示顺序关系。其中状态(圆圈)、任务(方框)以及顺序(有向线段)构成流程的结构信息。

业务流程由具体的任务以及任务执行顺序组成。给定一

个工作流业务流程,从流程结构上讲,其主要包括:节点(圆圈、方框)、边(有向线段)、节点配置和节点约束信息。节点和边是构成业务流程的主要元素;节点配置和节点约束关联到具体节点,用于描述或限制节点使用情况。结构信息用类图表示如图 2。

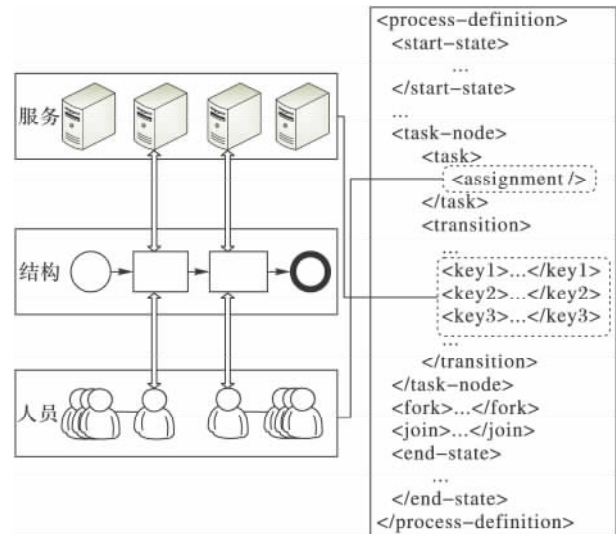


图 1 工作流模型与 JPDL 映射关系

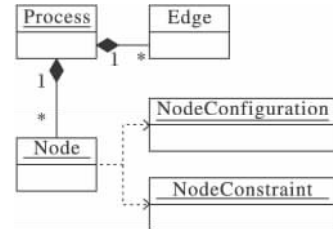


图 2 结构信息元素类图

各元素详细描述如下。

1) 节点(Node)。节点代表业务流程中的任务和状态。在服务集成环境下,节点通过关联到第三方工具的服务来确定流程中的任务,通过关联到人员信息来确定任务执行者。节点是服务信息与人员信息相关联的桥梁。

2) 边(Edge)。边代表业务流程中任务间的关联关系。在服务集成环境下,通过有向边串联所有任务节点,边的方向表达了节点间的执行顺序。

3) 节点配置(NodeConfiguration)。节点配置用于表达业务流程中节点的配置内容。在服务集成环境下,节点配置包括节点关联服务信息、人员信息以及流程分支信息。分支信息应用于复杂业务流程定制中,通过配置节点分支信息确定流程下一步分支结构。分支信息种类如表 1。

表 1 节点配置中的分支信息种类

分支类型	分支含义
简单分支	后续节点都执行
条件分支	后续节点中满足条件的节点执行
择一分支	后续节点中只需执行一个
汇合	前面节点执行后汇合到当前节点

4) 节点约束(NodeConstraint)。节点约束用于规范节点在流程定制中的使用。在服务集成环境下,存在很多业务规则,通过节点约束可以规范节点的连接属性。比如节点规定出度为 2,那么该节点后续可连接其他节点的数目最多为 2。

通过判断是否满足节点约束信息可以保证流程结构的正确性。

在 JPDL 中: 流程开始节点和结束节点对应到 < start-state > 和 < end-state >; 流程中的任务节点对应到 < task-node >; 而边的信息用于表示节点间的转移, 对应到 < transition >、< fork >、< join > 等标签。对于任意给定的流程, 都可以通过图 1 中的映射关系实现流程与 JPDL 元素的一一对应, 保证了流程结构的完整性和正确性。

1.2 服务信息

业务流程中每一个任务节点都与具体服务相关联。业务流程的执行是其所关联服务按序被调用的结果。为了保证业务流程的正确执行, 合理的服务描述信息也是 workflow 模型所必需的。

在服务集成环境下, 服务信息是关键, 也是与现有 workflow 定制主要区别所在。为了保证服务的正确调用, 必须明确了解服务信息。设计合理的服务信息表达方式是解决服务描述信息少和定制过程复杂的关键。服务信息用类图表示如图 3。

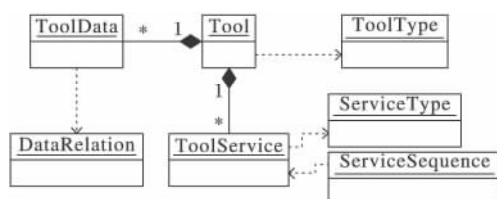


图3 服务信息元素类图

各元素详细描述如下。

1) 工具(Tool)。工具代表服务提供方, 在服务集成环境下指的是第三方工具。第三方工具通过 API 对外提供服务。业务流程节点处所关联的服务来自于第三方工具。

2) 工具服务(ToolService)。工具服务代表服务实体, 在服务集成环境下指的是第三方工具对外提供的服务。在业务流程中, 服务信息关联到具体节点, 依托于结构信息进行呈现。为满足服务集成的需要, 业务流程中所关联服务信息应该包含表 2。

表2 工具服务包含属性

属性名称	属性含义
serviceName	服务名称
serviceDescription	服务描述
serviceAddress	第三方工具数据交互服务地址
serviceWebAddress	第三方工具 Web 功能界面地址
inputParameters	服务调用所需参数
outputParameters	服务调用后返回数据参数
expectParameters	服务调用返回数据的期望格式

3) 服务类型(ServiceType)。服务类型即工具服务所属的类别。在服务集成环境下, 工具服务很多, 根据服务的功能属性及服务含义可抽象出服务类型信息, 如创建缺陷类服务、删除需求类服务等。这些服务类型名称具有明显的语义性, 能表达业务含义。

4) 服务顺序(ServiceSequence)。服务顺序即服务之间的关联规范。在流程定制中, 服务顺序信息通过限制服务间的可连接性来约束流程定制。服务顺序根据实际业务场景确定, 比如 A 类型服务后面可以连接 B 类型服务, 可以通过点

对(A, B) 来表示。

5) 工具类型(ToolType)。工具类型即第三方工具所属的域。在服务集成环境下, 可根据第三方工具功能范围和属性进行划分。比如 Bugzilla 属于缺陷管理域, TestLink 属于测试管理域。

6) 工具数据(ToolData)。工具数据用于记录业务流程执行中第三方工具所产生的数据。工具数据应该包含所属工具、数据类型、数据内容等信息。如工具 A 中的 B 类型数据 C 在数据库中可以通过一行记录表示。

7) 数据关系(DataRelation)。数据关系用于描述不同工具间数据的关联关系。服务集成环境下的业务流程涉及到多种工具服务, 流程运行时所产生的数据间存在关联关系。

服务信息是服务集成环境下的关键信息所在。详细的工具服务描述信息解决了传统 workflow 对节点服务描述信息较少的问题, 为在业务流程执行时完成服务调用提供了丰富的信息支持, 满足了服务集成需要。

根据服务信息可以将服务按两个维度进行分类: 服务类型(ServiceType) 和工具类型(ToolType)。服务分类后, 可以将分类结果作为节点类型信息。如创建缺陷, 节点具有业务含义, 使得流程拖拽定制过程简单, 业务流程也更易理解。服务顺序(ServiceSequence) 和节点约束(NodeConstraint) 分别从服务连接角度和流程结构角度规范了业务流程定制。

服务信息依托于结构信息中的节点进行配置, 对应到 JPDL 的 < task-node > 中, 可以通过自定义标签进行标记。< type > 用于记录服务类型, < servicename > 用于记录服务名称, < address > 用于记录服务调用地址等。由于是自定义属性标签, 可以按需进行拓展。在服务集成的业务场景下, 对于任何连接到集成环境中的服务, 都可以通过这种方式进行映射, 实现服务的接入。

1.3 人员信息

人员信息描述流程参与者的信息。业务流程的实现是活动按某种特定顺序执行的结果, 而活动的执行需要人员的参与, workflow 模型中必然包含人员信息。一个公司或者组织内部, 人员信息非常地繁杂。对于人员, 有两种划分准则: 一种是按照职能划分为不同的角色, 如开发人员、项目经理; 另一种是按照组织结构划分为不同的组织单元, 如开发组、测试组。本文给出了人员信息表示模型, 用类图表示如图 4。

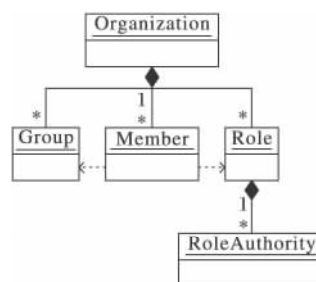


图4 人员信息元素类图

各元素详细描述如下。

1) 组织(Organization)。组织是定制业务流程的基本团体, 如公司或者社团等。在服务集成环境下, 每个组织都有自己的业务流程, 不同组织间的业务流程相互独立, 数据隔离。

2) 分组(Group)。分组是组织内人员的细分。实际业务

流程涉及到组织中的多人参与,可将人员按操作范围再次划分为更小的单位。

3) 角色(Role)。角色是组织内人员按职能权限进行的划分。在业务流程定制和执行过程中,角色保证了流程正确性和数据安全性。

4) 角色权限(RoleAuthority)。角色权限用于描述角色的功能范围。流程定制过程中依据角色权限选择任务参与人员。

5) 组织成员(Member)。组织成员属于某种角色,拥有对应权限;属于某一分组,拥有其操作范围。

按照人员的职能进行划分,限定了人员的操作权限;按照组织结构划分限定了用户的操作范围,这样便保证了业务流程在人员操作上的合法性和合理性。人员信息依托于结构信息进行展示。对应到 JPDL 的 < task-node > 中,通过 < assignment > 来进行标识。在服务集成环境下,对于组织内的任何成员,都可以连接到系统中。

2 工作流模型的实现方案及使用方式

本文提出的工作流模型在具体实现时可采用如图 5 所示的类图。

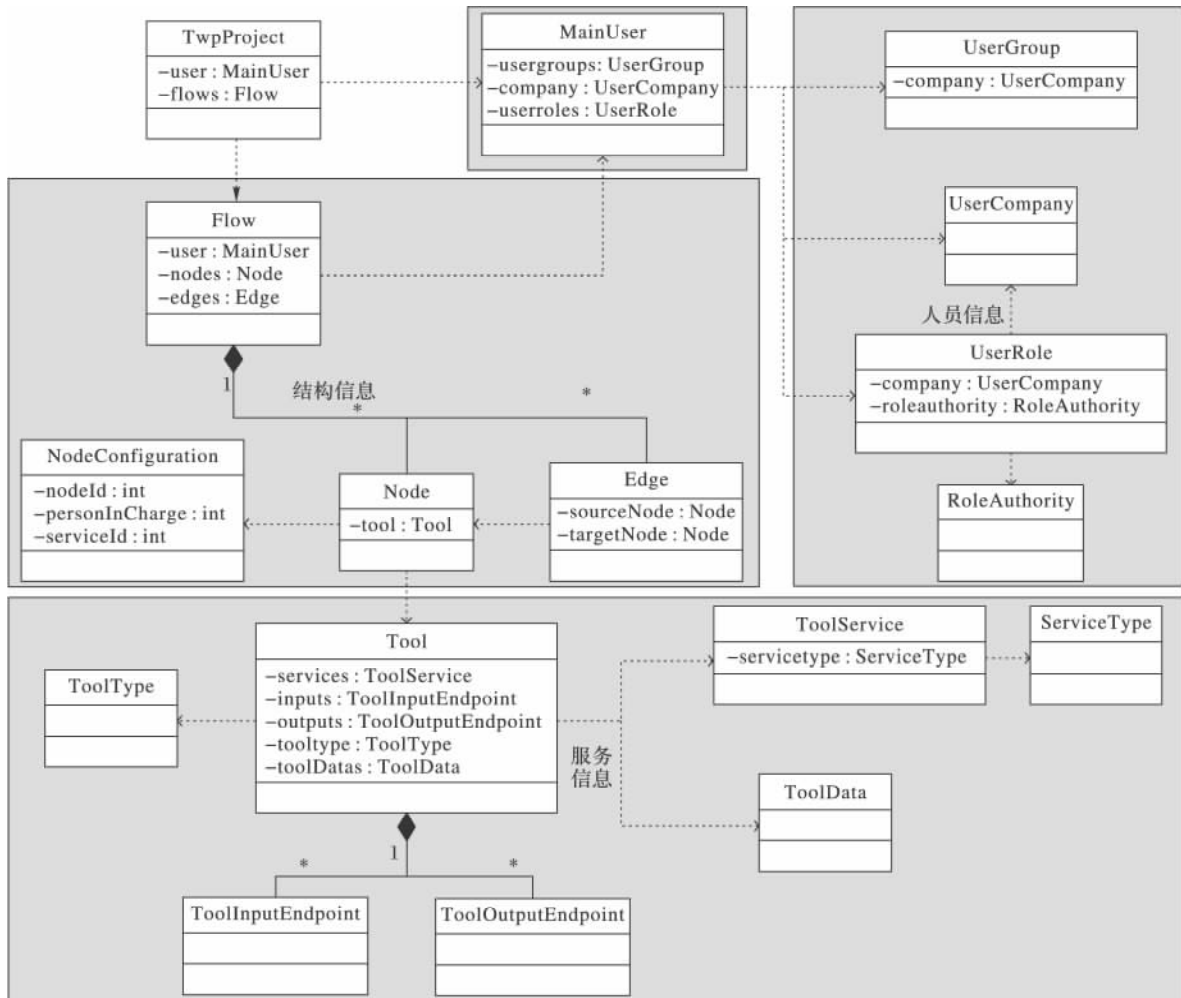


图 5 工作流模型类图

在结构信息的节点(Node) 类中关联人员信息和服务信息中的实体类,使得服务信息和人员信息可依托结构信息进行配置;通过结构信息中的节点配置(NodeConfiguration) 类记录节点的服务配置和节点配置等。编码实现图 5 所示类图即可实例化工作流模型。类图中给出了模型的关键要素及其之间的关联关系。用户定制的业务流程所包含的三部分信息可以分别对应到实例化后的实体类中,然后持久化到数据库中进行存储。当查看或者编辑现有业务流程时,首先从数据库中取出三部分信息,然后根据信息之间的关系重构业务流程,最终以图形化的形式展现给用户。

在模型实现之后,使用该模型可以作业务流程的定制与执行。其流程如图 6 所示,共由 5 步构成: 1) 定制业务流程,流程信息由本文提出的工作流模型进行表示; 2) 对业务流程

进行验证,保证流程定义的正确性; 3) 对于验证通过的业务流程,可以根据工作流模型与 JPDL 的映射关系进行流程翻译,将流程信息翻译成工作流语言; 4) 通过工作流引擎部署业务流程; 5) 调用工作流引擎接口执行工作流。

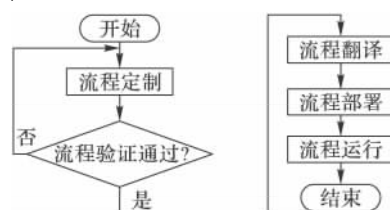


图 6 业务流程执行顺序

为了确保业务流程执行的正确性,在业务流程执行前,需要对业务流程进行简单的正确性验证。本文有效地解决了包

括开始节点、结束节点、顺序节点和分支节点的业务流程验证。对于更为复杂的业务流程验证,在未来的研究工作中会作进一步研究探讨。由上文可知,workflow 模型的结构信息中存储了节点约束信息,服务信息中存储了服务顺序信息。针对两类信息,业务流程的验证可以分为两部分:面向结构信息的验证以及面向服务信息的验证。

面向结构信息的验证保证了业务流程不存在异常结构。基于流程的结构信息,可以构建一个有向图。在该图形结构上可以通过图的遍历算法以及拓扑排序进行异常结构分析。其分析点主要包括:

- 1) 是否具有开始和结束节点;
- 2) 是否存在循环结构;
- 3) 是否存在不可达节点;
- 4) 节点使用是否满足节点约束和服务顺序约束。

面向服务信息的验证即保证业务流程中所关联服务的正确使用。基于服务信息中的服务顺序信息以及服务描述信息,可以判断服务间的连接是否正确。服务顺序信息是实际业务流程中抽象出的服务使用规则。例如创建缺陷类服务后面不可以关联创建需求类服务,那么在验证时就要对流程中各节点的服务进行该规则的判断。

本文中 workflow 模型只是提供存储 workflow 信息的机制,业务流程在定制过程中产生业务流程数据,该数据根据 workflow 模型分成三部分信息进行持久化存储。workflow 执行仍然依托于现有的 workflow 引擎。对于通过验证的 workflow,首先将其翻译成 workflow 描述语言——JPD,然后再由 JBPM workflow 引擎执行。其过程如图 7 所示。



图7 业务流程翻译与执行过程

根据 workflow 模型与 JPD 的映射关系进行流程翻译时,其伪代码描述如下。

算法 流程翻译算法——BFS_Translation(Node)。

宽度优先搜索业务流程节点,从开始节点执行。Q 为待检查队列;数组 s 将用来标示各顶点是否曾被放进待检查队列,是则标记为 1;否则标记为 0;JPD_FILE 表示翻译后的流程文件。

输入: 节点(nodes)。

节点数据结构包含元素: 后续节点(nextNodes);前置节点(preNodes);节点服务(ToolService);节点配置(NodeConfiguration);节点类型(.nodeType)。

输出: JPD workflow 语言。

方法:

```
AddQ( node Q); //首先访问开始节点 将 Q 初始化只包含一个元素
while Q 非空 do
    current_node = GetHeadNode( Q);
```

```
if current_node 为开始节点 then
    JPD_FILE.append( < start-state > ,transition);
elseif current_node 为结束节点 then
    JPD_FILE.append( < end-state > ,transition);
else
    if current_node 为 fork 节点 then
        JPD_FILE.append( < state > ,transition2fork);
    else
        JPD_FILE.append( < state > ,transition2join);
    endif
endif
if current_node.nextNodes.size > 1 then
    JPD_FILE.append( < fork > );
endif
if current_node.preNodes.size > 1 then
    JPD_FILE.append( < join > );
endif
for next_node in current_node.nextNodes do
    if s[next_node] = 0 then
        AddQ( next_node ,Q);
        s[next_node] = 1;
    endif
endfor
endwhile
```

3 案例研究

3.1 软件可信评估平台介绍

软件可信评估平台是一个 Web 应用工具集成系统,主要用于集成软件生命周期中的管理工具,如需求管理工具 Jira、缺陷管理工具 Bugzilla、测试工具 TestLink 等,从而实现软件开发全生命周期的集成管理,并在生命周期中统一采集数据,评估项目可信等级。在集成系统中可利用工具已有服务构建业务流程,在业务流程执行过程中搜集项目开发数据,最终基于数据评估项目可信等级。本文的 workflow 模型便应用到该平台的业务流程定制与执行过程中。系统的整体架构如图 8。

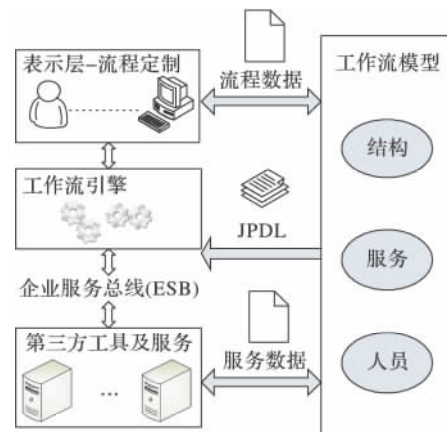


图8 系统整体架构

系统分为 4 层结构,如下所示。

1) 第三方工具及服务。Bugzilla、Jira 等第三方工具,可以通过 API 对外提供服务。

2) 企业服务总线(Enterprise Service Bus, ESB)。主要用于将工具服务连接到系统,并且规范服务数据格式以及服务调用方式。

3) 工作流引擎。开源工作流引擎 JBPM 用于任务分配、

状态转移以及流程执行。 workflow 执行过程中调用服务总线上的服务。

4) 表示层。分为业务流程定制界面和业务流程操作界面。其中,业务流程定制界面是基于 Web 的可拖拽界面,用于定制业务流程;业务流程操作界面用于展示业务流程执行时界面。

业务流程数据依据 workflow 模型中的三部分信息持久化到数据库中;根据 workflow 模型与 JPD L 的映射关系,流程数据被翻译为 workflow 语言,交由 workflow 引擎执行;在流程执行过程中,服务调用信息和产生的服务数据依据模型进行存储。 workflow 模型保证了业务流程的正确定制和执行,属于 workflow 参考模型^[1]中的流程定义模块。

3.2 软件可信评估平台系统验证

在可信证据采集集成平台上存在如下业务:针对 Jira 上面的某一个需求,通过 Qone^[14]对其分配任务,在任务执行后利用 TestLink 构建测试用例,对于测试出的问题要记录到 Bugzilla 中。在业务流程定制界面上,定制 workflow 如图 9。

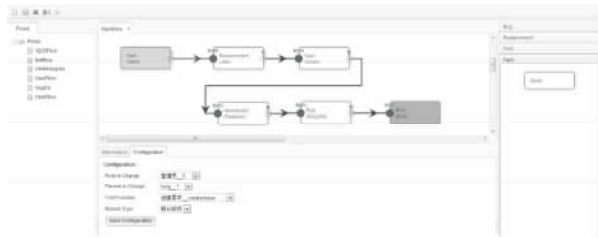


图 9 业务流程定制

图的右边是根据工具类型进行划分后的工具列表,用户根据描述便知道第三方工具的所属域。对于所拖拽的流程,可以对其节点进行配置。配置项包括操作者类型、操作者 ID、服务和分支类型。操作者类型和操作者 ID 对应模型中的人员信息,用于描述任务的执行者;服务用于确定节点所关联的工具服务,对应于 workflow 模型的服务信息,包含服务类型信息和具体服务信息;分支类型用于定制复杂业务流程,确定后续流程的分支结构。

对于定制好的业务流程,可以对其进行流程验证,验证通过后,系统自动翻译为 JPD L。最终调用 workflow 引擎接口部署并执行业务流程,执行过程中用户可以查看自己的任务。如图 10 所示。

尚未开始的任务:			
任务名称	任务开始时间	任务结束时间	备注
创建项目	尚未开始	尚未结束	
尚未结束的任务:			
任务列表			

图 10 用户任务列表

用户执行任务时,便可以调用相应的工具界面以及服务,如图 11 所示。

3.3 软件可信评估平台系统评估

现在系统所拥有的关键元素及数目如表 3 所示。

可信评估的基本单位是项目,平台上现有 2 个项目,分别隶属于 2 个公司,每个公司都拥有 2 个开发组,平台中的用户分别隶属于这 4 个开发组。通过公司和分组信息限制了用户的作用范围,通过角色限制了用户的权限。这些实际数据有效地证明了流程定制的规范性。



图 11 创建缺陷服务

表 3 系统元素统计

元素	数目	元素	数目
公司	2	工具服务	35
开发组	4	可利用业务流程	48
角色	2	平台项目	2
第三方工具	8		

现在系统中已经接入 8 个工具及其 35 个服务。这些服务经过简单的拖拽便可以构建 48 种满足实际需求的业务流程。实验结果表明,48 种业务流程都运行正确,满足实际业务需求;业务流程从定制到运行平均时间小于 2 min;工具服务以及数据顺利集成,工具界面可在集成平台上嵌入使用。与传统开发相比,节约了为满足新需求进行详细设计、代码编写和测试的时间和资源,从而大幅度地减少为新业务需求开发软件的成本。

4 结语

本文针对服务集成环境下的业务流程定制,提出一种由结构信息、服务信息和人员信息构成的 workflow 模型,可以有效地解决现有 workflow 系统在服务集成环境下存在的两个问题:1) 现有 workflow 系统对于业务流程中所关联服务的描述信息少,无法服务集成需要;2) 现有 workflow 系统节点类型单一,节点本身不能表达业务含义,导致流程定制复杂。首先,通过三部分信息保证了业务流程的正确表达;其次,通过丰富的服务描述满足了服务集成所需信息,而且从服务描述中可对复杂多样的服务进行分类划分,支持节点类型的拓展。最终通过实际系统验证了 workflow 模型的有效性。实验结果表明,该模型的提出减少了为新需求开发软件的成本。

本文提出的 workflow 模型只是针对第三方工具服务集成场景下的业务流程定制,重点是解决该场景下服务多样性造成的问题。但是目前 workflow 模型只支持简单的业务流程定制与验证,进一步工作将针对本文的 workflow 模型,拓展业务流程类型支持范围,并提出流程验证模型并给出验证方法。

参考文献:

- [1] Workflow Management Coalition. WfMC-TC-1003: the workflow reference model [S]. Cohasset: [s. n.], 1995: 1-55.
- [2] LIU Y, ZHANG Z, ZHANG K. Overview of workflow models [J]. Computer Engineering and Design, 2007, 28(2): 448-451. (刘怡, 张子刚, 张戡. 工作流模型研究述评[J]. 计算机工程与设计, 2007, 28(2): 448-451.)
- [3] SADIQ W, ORLOWSKA M. Analyzing process models using graph reduction techniques [J]. Information Systems, 2000, 25(2): 117-134.

(下转第 2003 页)

要技术的有效性将是下一步研究工作的重点。

参考文献:

- [1] EDDY B P, ROBINSON J A, KRAFT N A, *et al.* Evaluating source code summarization techniques: replication and expansion [C]// ICPC 2013: Proceedings of the 2013 IEEE 21st International Conference on Program Comprehension. Piscataway: IEEE, 2013: 13–22.
- [2] LATOZA T D, VENOLIA G, DELINE R. Maintaining mental models: a study of developer work habits [C]// ICSE 2006: Proceedings of the 2006 International Conference on Software Engineering. New York: ACM, 2006: 492–501.
- [3] KO A J, MYERS B A, COBLENZ M J, *et al.* An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks [J]. IEEE Transactions on Software Engineering, 2006, 32(12): 971–987.
- [4] MORENO L, APONTE J, SRIDHARA G, *et al.* Automatic generation of natural language summaries for Java classes [C]// ICPC 2013: Proceedings of the 2013 IEEE 21st International Conference on Program Comprehension. Piscataway: IEEE, 2013: 23–32.
- [5] HAIDUC S, APONTE J, MORENO L, *et al.* On the use of automated text summarization techniques for summarizing source code [C]// WCRE 2010: Proceedings of the 2010 17th Working Conference on Reverse Engineering. Piscataway: IEEE, 2010: 35–44.
- [6] MCBURNEY P W, LIU C, McMILLAN C, *et al.* Improving topic model source code summarization [C]// ICPC 2014: Proceedings of the 2014 22nd International Conference on Program Comprehension. New York: ACM, 2014: 291–294.
- [7] KAJKO-MATTSSON M. A survey of documentation practice within corrective maintenance [J]. Empirical Software Engineering, 2005, 10(1): 31–55.
- [8] SAHARIA N, SHARMA U, KALITA J. A suffix-based noun and verb classifier for an inflectional language [C]// IALP 2010: Proceedings of the 2010 International Conference on Asian Language Processing. Piscataway: IEEE, 2010: 19–22.
- [9] JURAFSKY D, MARTIN J H. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition [M]. Upper Saddle River: Prentice Hall, 2000: 139–141.
- [10] CAPOBIANCO G, de LUCIA A, OLIVETO R, *et al.* Improving IR-based traceability recovery via noun-based indexing of software artifacts [J]. Journal of Software: Evolution and Process, 2013, 25(7): 743–762.
- [11] FRY Z P, SHEPHERD D, HILL E, *et al.* Analysing source code: looking for useful verb-direct object pairs in all the right places [J]. IET Software, 2008, 2(1): 27–36.
- [12] CAPOBIANCO G, de LUCIA A, OLIVETO R, *et al.* On the role of the nouns in IR-based traceability recovery [C]// ICPC 2009: Proceedings of the 2009 IEEE 17th International Conference on Program Comprehension. Piscataway: IEEE, 2009: 148–157.
- [13] ZAMANI S, LEE S P, SHOKRIPOUR R, *et al.* A noun-based approach to feature location using time-aware term-weighting [J]. Information and Software Technology, 2014, 56(8): 991–1011.
- [14] RODEGHERO P, McMILLAN C, MCBURNEY P W, *et al.* Improving automated source code summarization via an eye-tracking study of programmers [C]// ICSE 2014: Proceedings of the 2014 36th International Conference on Software Engineering. New York: ACM, 2014: 390–401.
- [15] WANG J, WU G, ZHOU Y, *et al.* Research on automatic summarization of Web document guided by discourse [J]. Journal of Computer Research and Development, 2003, 40(3): 398–405. (王继成, 武港山, 周源远, 等. 一种篇章结构指导的中文 Web 文档自动摘要方法[J]. 计算机研究与发展, 2003, 40(3): 398–405.)
- [16] SHEPHERD D, FRY Z P, HILL E, *et al.* Using natural language program analysis to locate and understand action-oriented concerns [C]// AOSD 2007: Proceedings of the 2007 6th International Conference on Aspect-Oriented Software Development. New York: ACM, 2007: 212–224.

(上接第1998页)

- [4] LI H, SHI M. Workflow models and their formal description [J]. Chinese Journal of Computers, 2003, 26(11): 1456–1463. (李红臣, 史美林. 工作流模型及其形式化描述[J]. 计算机学报, 2003, 26(11): 1456–1463.)
- [5] BAJAJ A, RAM S. SEAM: a state-entity-activity-model for a well-defined workflow development methodology [J]. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(2): 415–431.
- [6] YANG Y. An extended stochastic Petri nets modeling method for collaborative workflow process [C]// ICMPE 2012: Proceedings of the 2012 International Conference on Medical Physics and Biomedical Engineering. Amsterdam: Elsevier, 2012, 33: 1547–1552.
- [7] LI L, TANG J. Research of a cross-enterprise workflow model [C]// ICECE 2012: Proceedings of the 2012 International Conference on Electrical and Computer Engineering. Amsterdam: Elsevier, 2012, 12: 195–203.
- [8] LI H, ZHANG M. A dynamic workflow model based on Petri net and instance migration [C]// ICACI 2012: Proceedings of the 2012 International Conference on Affective Computing and Intelligent Interaction. Texas City: Information Engineering Research Institute, 2012, 10: 320–325.
- [9] BAI L, ZHANG X. A workflow model based on extended multi-dimensional workflow net and performance analysis [C]// ESRIS 2012: Proceedings of the 2012 International Conference on Earth Science and Remote Sensing. Texas City: Information Engineering Research Institute, 2012, 30: 291–298.
- [10] XU M, XU Q. Workflow model analysis based on colored Petri nets [C]// ICCEE 2010: Proceedings of the 3rd International Conference on Computer and Electrical Engineering. Singapore: IACSIT, 2010: 250–256.
- [11] CHENG J. Study of Web service workflow based on Petri net [D]. Hefei: Hefei University of Technology, 2006. (程俊. 基于 Petri 网的 Web 服务集成工作流的研究[D]. 合肥: 合肥工业大学, 2006.)
- [12] CHENG Q. Research on Web service workflow transaction model [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2013. (程强. Web 服务工作流事务模型研究[D]. 南京: 南京邮电大学, 2013.)
- [13] RIZESCU G. Businesses integration with workflow and Web service technologies [J]. Annals of Dunarea de Jos University. Fascicle I: Economics and Applied Informatics, 2008, 1: 87–94.
- [14] BAIKE.COM. Qone [EB/OL]. [2014-12-20]. <http://www.baik.com/wiki/qone>.