

基于面向服务业务流程管理的系统架构建模

李润晔, 倪 枫, 刘 姜, 钟贤欣, 全启宸, 周彦琪
(上海理工大学 管理学院, 上海 200093)

摘要: 信息时代对业务流程管理结合面向服务架构的需求愈发显著, 针对现阶段该领域研究中, 概念与建模语言相脱节、忽视计算无关模型到平台无关模型的转换、难以贯穿系统多个视图等不足, 将业务流程管理 (BPM) 与面向服务架构 (SOA) 方法结合应用到面向服务系统架构的业务流程建模。通过区分本体元模型与标记元模型的概念, 在两种元模型之间分别建立映射关联: 语义层映射和语法层映射, 并以 BPMN 和 SoaML 为具体建模语言分两步推理实现。最后将所提出的建模方法及 BPMN 和 SoaML 之间的映射规则应用于一个游乐园智能手环系统架构建模案例中, 验证了方法的可行性。

关键词: 业务流程管理; 面向服务架构; 模型驱动架构; 元模型映射
中图分类号: N 94 **文献标志码:** A

System Architecture Modeling Based on Business Process Management and Service-Oriented Architecture

LI Runye, NI Feng, LIU Jiang, ZHONG Xianxin, QUAN Qichen, ZHOU Yanqi
(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: With the increasing demand of combining business process management (BPM) with service-oriented architecture (SOA) in the information age, and focusing on the deficiencies of researches in this field currently, such as disjunction between concepts and modeling languages, ignorance of conversion from computing independent model to platform independent model, and difficulty in penetrating multiple views of a whole system, the business process modeling of service-oriented architecture was introduced based on the combination of BPM and SOA. Then by means of distinguishing the concepts of ontology meta-model and notation meta-model, the mapping relationship between two modeling languages was established, including the semantic layer mapping as well as the syntactic layer mapping, and was implemented by two-step-inference taking BPMN and SoaML as specific modeling languages. Finally, the modeling method as well as mapping rules between BPMN and SoaML were applied to a case study in order to prove the feasibility.

Keywords: *business process management; service-oriented architecture; model driven architecture; meta model mapping*

1 BPM+SOA 架构方法

信息时代，企业业务环境瞬息万变，对信息系统的复杂性、敏捷性等要求越来越高。虽然业务流程管理 (business process management, BPM) 能促进业务流程的优化，但无法对齐不断变化的业务流程和 IT 系统。而能衔接 BPM 与企业后端并封装应用系统功能的面向服务架构 (service-oriented architecture, SOA) 可以通过其松耦合、粗粒度的特点，弥补 BPM 固化于各系统的缺陷，还能将业务流程转化为可复用组件，利用接口和协议相互联系，促进系统间信息以及业务逻辑的共享。两者结合，融合信息技术与管理技术，确保业务流程的高效性以及 IT 架构的敏捷性，充分利用现有资源，促进业务间协作。

对象管理组 (object management group, OMG) 所提出的模型驱动架构 (model driven architecture, MDA) 为 BPM+SOA 系统架构所涉及的多层次建模提供参考，其开发软件主要从计算无关模型 (computing independent model, CIM) 到平台无关模型 (platform independent model, PIM)，再到平台相关模型 (platform specific model, PSM)，最后转换成代码。由此，开发各阶段的工作成果得到保护，过程由一系列模型驱动，并经过迭代使其不断完善^[1]。如今基于 MDA 的复杂系统建模主要有通用方法^[2]、并行方法^[3]、对象-过程法^[4]和状态分析法^[5]这 4 种。

近年来，国内外对 BPM+SOA 的研究众多，如 Beimborn 等证明了两者的互补性^[6]；王玉娟促进了中石化科技管理系统复杂业务逻辑的可维护性与可扩展性的实现^[7]；Zhao 等以 BPM 为出发点，结合 SOA 与企业服务总线实现系统的集成^[8]；Alam 等对面向服务架构方法的发展与影响进行系统评估^[9]；Laumer 等通过调查，得出信息系统与不同业务流程管理组件结合，将产生不同流程绩效的结论^[10]；Hachicha 等为面向服务架构中的协作业务流程提供了分析与评估的方法^[11]。然而，这些研究与建模语言的联系不够紧密，限制了通过 BPM+SOA 方法进行系统架构设计与建模的工程应用。

另外，BPM+SOA 的系统架构需要将计算独立的业务模型转换成平台独立的服务模型。模型转换是模型驱动架构的关键技术之一，可被用于将质量较低的模型重构成质量较高的模型^[12]，或将高抽象层的模型转换为低抽象层的模型。转换技术主要有直接法、目标结构驱动法、关系法、图转换法等^[13]。近年来，曾一等通过元模型实现 PIM 到 PSM 的自动转换^[14]；苏红军等验证了通过编程实现模型驱动转换的灵活性和功能实现的多样性^[15]。然而，目前国内学者的相关研究主要围绕 PIM 与 PSM 间的转换，甚少涉及 CIM，难以满足 BPM+SOA 的系统架构建模的需要。

元对象机制 (meta object facility, MOF) 中，元模型是模型的上层抽象，能满足模型自身的集成、存储以及管理等需要^[16]。在元模型应用中，本体元模型较为常见。如 Henderson-Sellers 结合元模型和本体论的概念，提出“领域本体论” (domain ontology)^[17]；如 Béhé 等所提出的本体元模型描述了多代理模拟概念^[18]；又如闫雪峰等所构建的本体设计元模型能够有效地转换并实例化成为虚拟样机仿真模型^[19]。部分学者还注意到系统架构中，建模语言和元模型之间的关系，提出元建模方法，如何啸等所提出的表示法定义元模型^[20]，又如 Pardillo 等对扩展 UML 配置元模型 (UML profile meta-model) 的研究^[21]。但元建模对语义的描述不够完善且在需要多侧面描述的复杂系统建模过程中的作用相对有限^[22]。

本文将在前人研究的基础上，利用 protégé 软件进行语义层和语法层两次推理，探究双方各自的建模语言之间的映射规范，并通过案例证明方法的可行性。

2 建模语言

2.1 BPMN

由业务流程管理倡议组织创建并由 OMG 发布的业务流程建模与标注 (business process model and Notation, BPMN) 在建模领域的优势主要在于：提供了一套易被业务用户理解的标记语言，所构建的模型注重分析与描述业务流程，有助于对流程的

可视化管理, 为其设计与实现提供标准化桥梁。BPMN2.0 语言则进一步为存储、交换与执行等问题提供解决方案。它将图形化标记分为 5 种基本类型, 即流对象(flow objects)、连接对象(connecting objects)、泳道(swim lanes)、人工制造物(artifacts)、数据(data)。

2.2 SoaML

面向服务架构建模语言 (service-oriented architecture modeling language, SoaML)是对标准建模语言(UML)的拓展, 关注并描述参与者的需要与功能, 并将他们在服务价值链中联系起来。其主要建模元素有参与者(participant)、接口(interface)、服务合约(service contract)、服务体系结构(service architecture)、服务数据 (service data)以及能力 (capability)。SoaML 的主要优势在于: 关注整个系统架构的实现, 为业务模型到 IT 模型搭设桥梁, 提供变化的可追溯性, 只要服务定义维持不变, 业务流程或信息技术方面就能进行相对独立的变更, 由此提升系统的敏捷性。

2.3 融合机制

在系统架构过程中, 融合 BPMN 与 SoaML 的建模能力。由于 BPMN 能够较宏观地展现多个参与者间的业务交互, 描述业务活动的实现过程, 确保系统能够紧密匹配持续优化的业务流程, 故以 BPMN 为主要建模语言的业务模型对应系统需求描述而独立于技术实现的 CIM; 由于 SoaML 将具体的业务活动封装成描述功能的服务, 并展现

服务间关系和数据流, 故以 SoaML 为主要建模语言的服务模型对应重视整个系统架构的实现而忽略平台相关部分的 PIM。以此为原则建立融合机制, 不但衔接业务层与技术层, 更符合模型驱动架构的要求。在 DoDAF 2.X(The Department of Defense Architecture Framework 2.0)框架下, BPMN 满足业务活动模型 OV-5b 的要求, 而 SoaML 满足展现参与者和服务以及两两相互关系的 SvcV-1 模型与描述服务功能以及功能中服务数据流的 SvcV-4 模型等的要求。

本文通过 No Magic 公司的 Cameo EA 产品构建 BPMN 和 SoaML 模型。Cameo EA 为系统架构提供强大而健全的标准, 促进同一系统不同视图模型数据元素的一致性, 有助于 BPMN 与 SoaML 的结合。

3 基于元模型映射的模型关联

3.1 元模型的分类与作用

结合上文所述, 将元模型分为本体元模型和标记元模型。本体元模型是基于概念所处语境以及自身语义的, 属于语义层的元模型。而为图形化建模语言服务的标记元模型则由于抽象图形在模型中本身带有一定的规则和语法, 属于语法层的元模型。图 1 体现了两者在 MOF 中的运用以及区别。

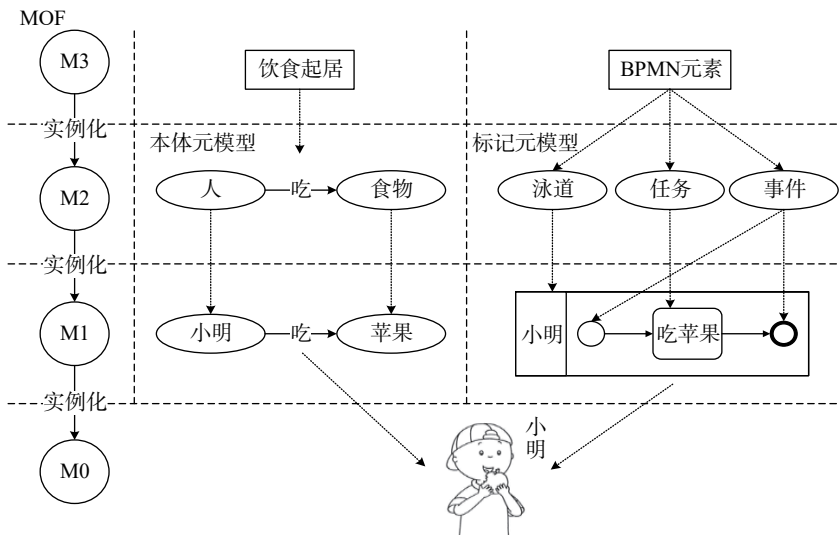


图 1 本体元模型和标记元模型在元对象机制中的运用及两者区别

Fig.1 Application of the ontology meta-model and notation meta-model in MOF and the differences between them

基于 Gerber 等的观点：“若一个本体模型能降低抽象级而建立实例化的模型，则这个本体模型就是该模型实例的元模型^[23]”。可以如此理解本体元模型：基于语义构建的，能够将其所包含的概念实例化成具体的、彼此间存在关系的词汇并构成模型的本体模型。本体元模型用集合语言表示为

$$MM_O = \{E_O, A_O, R_O\}$$

(1)

式中： MM_O 表示本体元模型； E_O 表示本体元素； A_O 表示相关属性； R_O 表示类与类或类与属性间的关系。

基于何啸等的表示法定义元模型^[20]、Pardillo 等的 UML 配置元模型扩展版^[21]、可视化建模语言使用指南中的元素分类图等，可以如此理解标记元模型：基于语法构建的，能将其所包含的标记实例化成具体的、彼此关联的符号并构成图形化模型的模型。标记元模型用集合语言表示为

$$MM_N = \{E_N, A_N, R_N\}$$

(2)

式中： MM_N 表示标记元模型； E_N 表示标记元素； A_N 表示相关属性； R_N 表示标记与标记或标记与属性间的关系。

图 1 中，M2 层的“人吃食物”是 M3 层的“饮食起居”这个高度抽象的概念性词语的一个实例。在这个本体元模型中，“人”和“食物”都是实体 E_O ，它们都具备自身的属性，即 A_O ，而“吃”成为了这 2 个实体之间的关系 R_O 。至于 M2 层的标记元模型，是对 M3 层的“BPMN 元素”这一抽象概述的一种实例化。不同于偏重语义的本体元模型，该标记元模型中“泳道”、“任务”、“事件”3 个实体 E_N 是 BPMN 模型中的 3 种图形符号，它们自身所具备的属性 A_N ，是由建模语言 BPMN 的语法要求所赋予的。比如，按照 BPMN 语法规定，“泳道”和“任务”的符号形状与用途都不相同，“泳道”有“一大一小两矩形框”与“体现行为主体”的属性，而“任务”有“圆角矩形框”与“表示流程中某个活动”的属性。实体间的关系 R_N ，也需要结合 BPMN 的语法要求。比如，“泳道”和“任务”的关系 R_N 是“任务”实体必须嵌在“泳道”实体中才能体现该“泳道”所代表的主体进行该“任务”。因此，本体元模型是可以基于自然语言语义直接构建，而标记元模型的构建必须考虑其上层，即某种可视化建模语言的语法规定。

由于系统架构涉及多个视图表述，其建模语言各不相同，并需要保持语义的一致性和语法的正确性，因此，代表语义层的本体元模型将维持不变，代表语法层的各建模语言标记元模型间能够通过本体元模型实现映射，而且标记元模型所对应的概念，是本体元模型中不同视图的交集 O ，所以，两者关系可如图 2 所示。

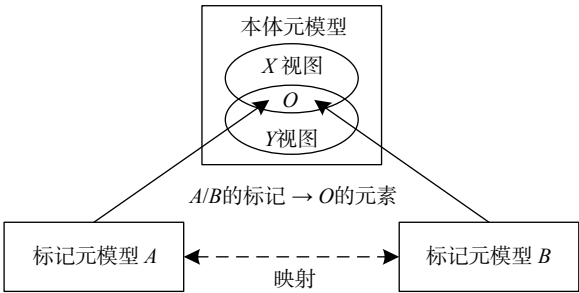


图 2 本体元模型和标记元模型的关系

Fig. 2 Relationship between the ontology meta-model and notation meta-model

3.2 本体元模型的构建

为了描述系统架构建模过程中所涉及到的概念，联系不同建模语言，需要构建一个本体元模型。首先，由于本文主要研究 BPM+SOA 的系统架构建模，因此，该本体元模型以运用 BPM 思想的业务视图和代表 SOA 理念的服务视图为主。其次，参考美国国防部提出的核心体系结构数据模型以及 DoDAF 元模型中所涉及的概念，提出适合各视图并具有代表性的本体元素。再次，为这些元素赋予功能属性，实现不同建模语言的标记元模型间的映射。最后，基于语义明确它们的关系，进而得到图 3 的本体元模型。

由图 3 可知，式(1)中的 MM_O 为图 3 所示的 BPM+SOA 系统架构的本体元模型；本体元素 E_O 有“节点”、“活动”、“信息”、“规则”、“资源”、“能力”等；相关属性 A_O 则是矩形框内的功能属性；“组成”和“是属性”则代表本体元素间以及本体元素和属性间的关系 R_O 。其中，业务视图和服务视图中的“信息”主要指模型中被传递的信息实例，该实例是活动输入、输出的来源，与活动是多对多的关系；而服务视图中的“资源”则描述系统在活动的执行过程中所涉及到的相关数据实体。另外，结合图 2 和图 3，可得当标记元模型 A 表示 BPMN 的标记元模型，而标记元模型 B 表示 SoaML 的标记元模型时，

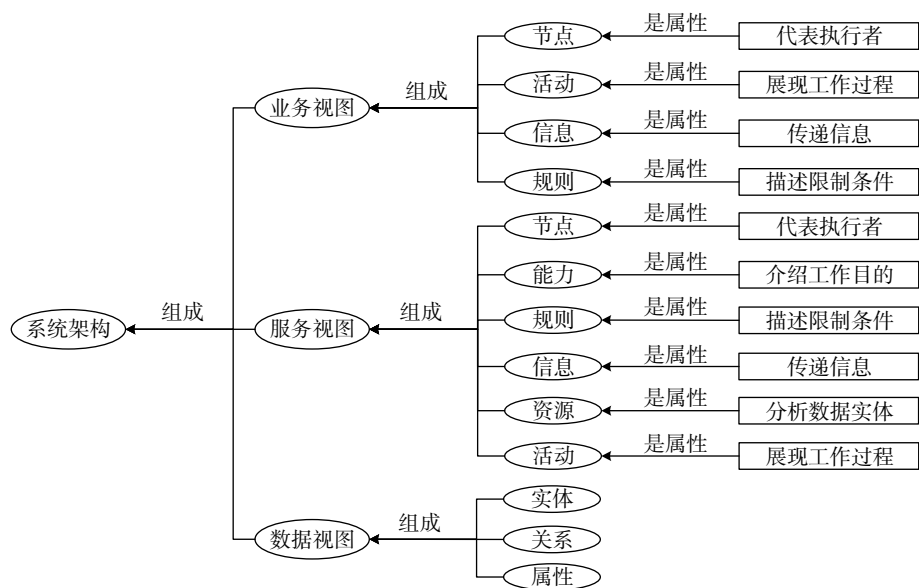


图 3 业务流程管理结合面向服务架构的系统架构的元模型

Fig.3 Ontology meta-model of system architecture based on BPM+SOA

A 属于业务视图, B 属于服务视图, 交集为

$$O = \{n, a, i, r\} \tag{3}$$

式中: n 表示节点; a 表示活动; i 表示信息; r 表示规则。

3.3 标记元模型的构建

建模语言的语法可视为诸多标记的语义的集

合, 所以, 标记元模型的构建方法与本体建模相似。而式(2)中的 MM_N 为图 4 所示 BPMN 的部分标记元模型以及图 5 所示的 SoaML 标记元模型; 图 4 和图 5 中的椭圆框代表标记元素 E_N ; 矩形框代表相关属性 A_N ; “属于”、“组成”和“是属性”则代表标记元素间以及标记元素和属性间的关系 R_N 。

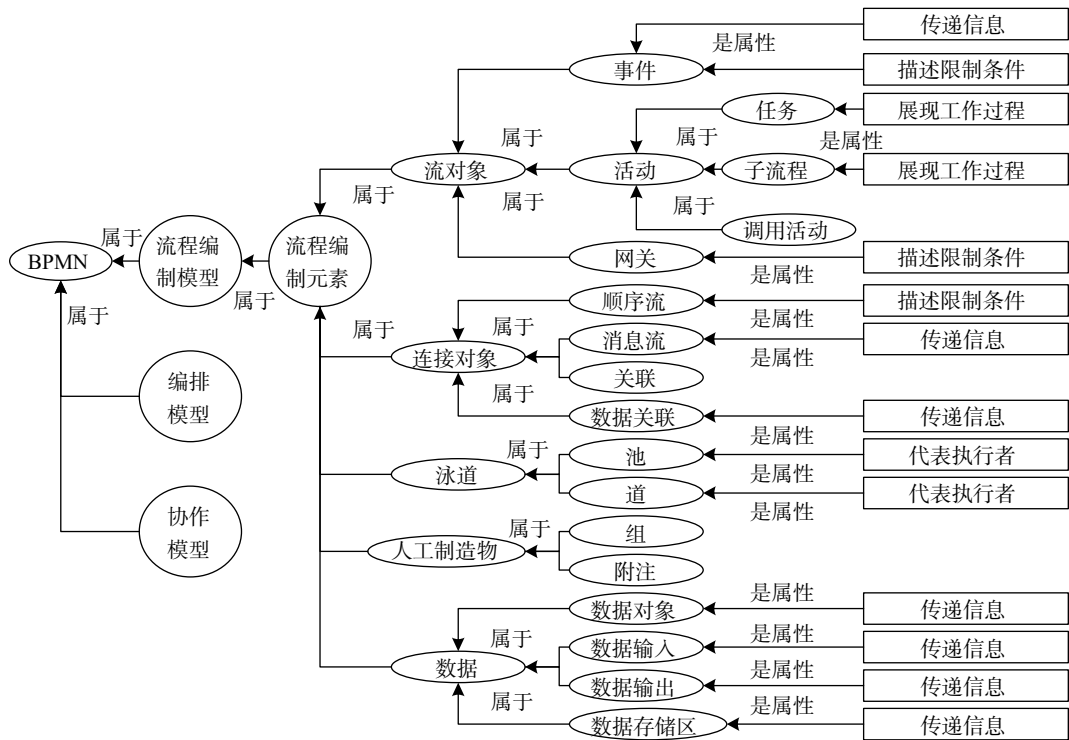


图 4 业务流程建模与标注的部分标记元模型 (以流程编制模型为主)

Fig.4 Partial notation meta-model of BPMN (mainly based on process orchestration)

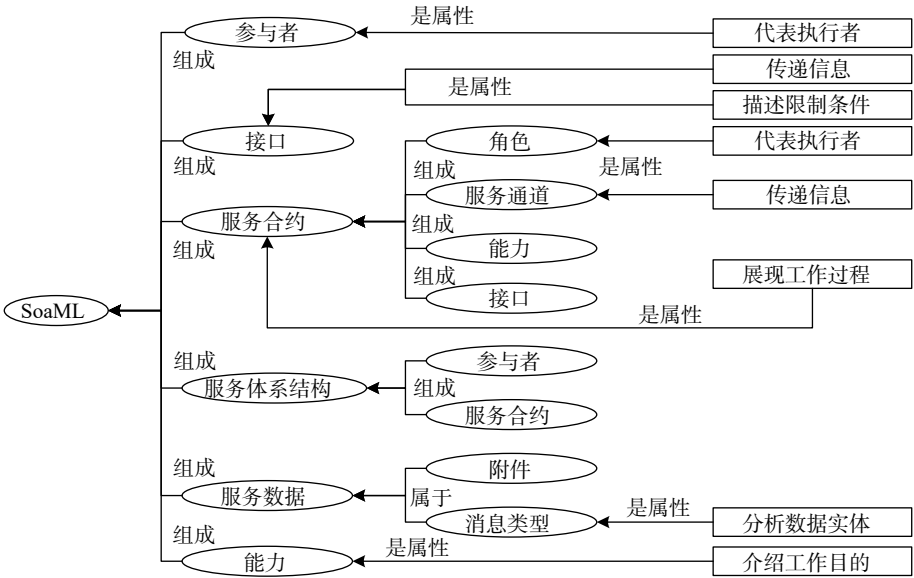


图 5 面向服务架构建模语言的标记元模型
Fig.5 Notation meta-model of SoaML

3.3.1 BPMN 的元模型

围绕业务视图中最常见的流程编制元素，在确定了重要的专业术语及其属性，并且结合语义关系明确了与 BPMN 标记元素的映射关联之后，可得如图 4 所示的以流程编制模型为主的部分 BPMN 标记元模型。

3.3.2 SoaML 的元模型

SoaML 对 UML 的延伸主要有 6 个方面，根据语义确定 SoaML 标记元素间关系并为它们赋予功能属性，可得如图 5 所示的 SoaML 标记元模型。

3.4 模型间映射

图 6 展示了本文使用 protégé 进行映射的思

路，其中，元素的功能属性是为了实现推理，之所以使用本体元模型，是因为系统架构是基于 DoDAF 框架的，需要应用该框架内的相关概念确保不同建模语言的标记元模型间映射的准确性，进而为不同层级模型间的转换始终满足 DoDAF 框架的要求提供保障。基于此思路，本文将推理映射分为两部分。

第一步是将标记元素映射到本体元素的语义层映射，当标记元素集 M 的属性值为功能属性集 F 时，便能利用推理建立 M 和属性值同为 F 相等类的本体元素集 E 之间的映射关系。第二步是为不同建模语言的标记元素构建映射关系的语法层

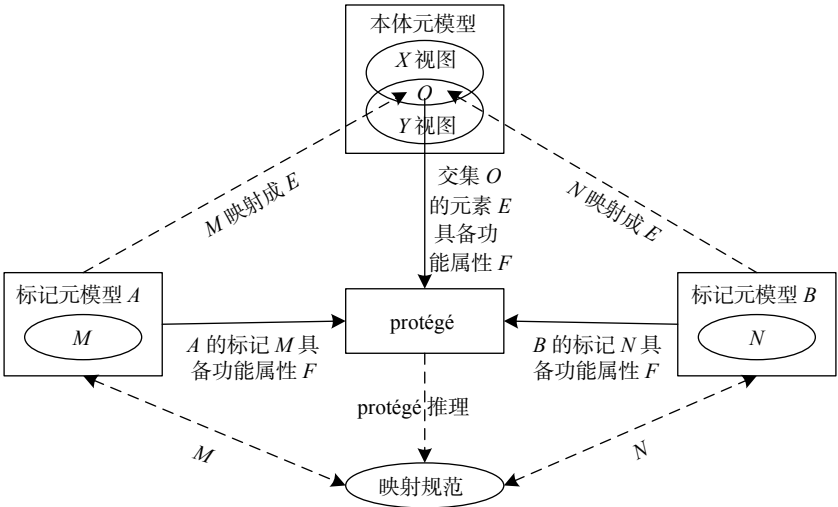


图 6 使用 protégé 进行元模型映射
Fig.6 Meta-model mapping by protégé

映射。基于上一步的推理，使属性值为 F 的标记元素集 M 和属性值为 F 的标记元素集 N 都成为了 E 的子集，进而以 E 为桥梁，生成 M 和 N 的映射规范。

语义层映射是为了确保语义间的一致性，即保障不同抽象级的模型始终处于同一个系统架构的框架下；而语法层映射是为了确保语法的准确性，即通过建立 2 个标记元模型间的映射规范，保障标记的使用规则满足当前的建模语言要求。语义层映射是语法层映射的基础，语法层映射是

语义层映射在系统架构建模中的进一步应用，所获得的映射规范则是两步推理的结论。

3.4.1 语义层映射

在 protégé 中完成本体元模型和标记元模型的构建后，使用其自带的推理机进行推理，本体元模型的各个本体元素因推理而具备子类，图 7 为借助 OWL Viz 插件展示的以本体元素“节点”为例的推理结果节选。可见，通过推理，标记元素根据其所具备的功能属性，映射到了本体元素，即语义层的映射得到实现。

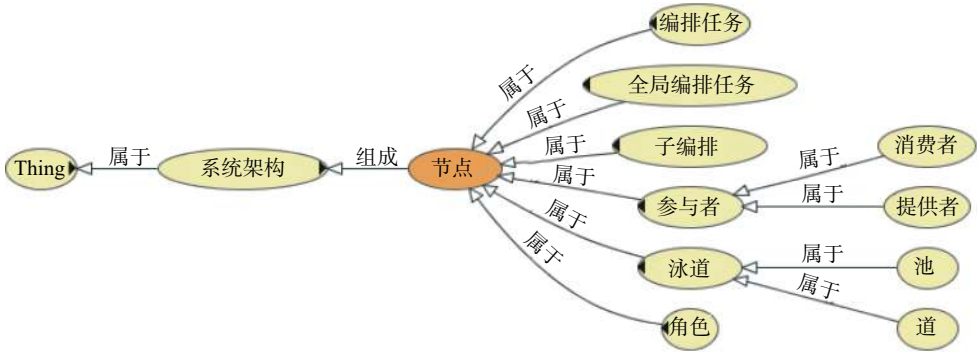


图 7 以“节点”为例的推理后模型

Fig.7 Inferred model in OWL Viz taking “node” as an example

3.4.2 语法层映射

在原有的基础上，新增一个类 mapping_rule，利用推理机进行推理，并将推理所得的映射规范整理成表 1。表 1 纵向是 BPMN 的标记元素，横向是 SoaML 的标记元素。表中●表示存在映射关系，空则表示不存在映射关系。

表 1 业务流程建模与标注和面向服务架构建模语言的映射规范

Tab.1 Mapping rules between BPMN and SoaML

BPMN 的标记元素	SoaML 的标记元素				
	服务通道	服务合约	接口	参与者	角色
任务		●			
子流程		●			
事件	●		●		
网关			●		
顺序流			●		
消息流	●		●		
数据关联	●		●		
数据	●		●		
泳道				●	●

综上所述，来自不同建模语言的标记元素通过系统架构本体元模型的不同抽象级的交集 O 生成标记间的映射规范，即语法层的映射得到实现。

在通过这两步映射所得的映射规范中，存在映射关系的 2 个标记元素必定是同一个本体元素因 protege 推理而具备的子类。又因为该本体元素所隶属的 BPM+SOA 系统架构本体元模型是基于 DoDAF 框架而构建的，同为该本体元素子类的标记元素皆在一定程度上符合 DoDAF 框架的要求，都能在各自的可视化模型中体现该本体元素的语义，所以，语义的一致性得到了保障。而本体元素和标记元素间以及标记元素之间的映射关系主要通过它们的功能属性来确立，标记元素的功能属性是由该标记元素所属的建模语言的语法规则所赋予的，因此，标记元素间的映射关系满足各自建模语言的语法要求，确保了语法准确性。

3.5 建模过程

对于 BPM+SOA 的系统架构下的模型构建，主要根据上述元模型映射的方法，并参考 DoDAF 的诸多视图。DoDAF 2.X 以数据为中心，支持

SOA, 并具备展现模型顶层特性的全景视图(all viewpoint, AV)、描述可部署能力和能力需求等的能 力视图(capability viewpoint, CV)、阐述所设想的业务和行动等的业务视图(operational viewpoint, OV)、体现执行者和服务以及交互关系的服务视图(service viewpoint, SvcV)、展现独立系统或遗留系统情况的系统视图(systems viewpoint, SV)、分析能力和业务与所实施项目的关系的项目视图(project viewpoint, PV)、描述数据关系并保证数据结构一致性的数据视图(data and information viewpoint, DIV)、展现可应用的标准或约束等的标准视图(standards viewpoint, StdV)。

建模过程如图 8 所示。第一步, 根据外界需求构建能力视图; 第二步, 构建以 BPMN 为主要建模语言的业务视图; 第三步, 按照映射规范, 构建以 SoaML 为主要建模语言的服务视图; 第四步, 构建相应的数据视图和其他视图; 第五步, 根据变化的需求, 以原能力视图模型为初始状态, 对其进行迭代, 即重复上述步骤, 完善其余视图模型, 持续优化系统, 保障系统敏捷性, 逐渐接近目标。图 8 中, 各视图下标 $n(n \geq 1)$ 表示迭代的次数, 每迭代一次, 都会使得 $n=n+1$, 由此循环迭代该系统架构建模过程。

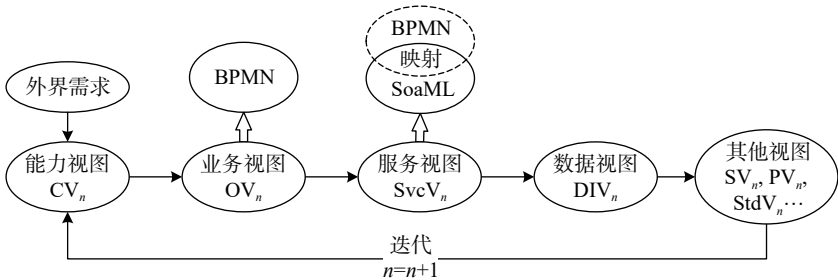


图 8 基于业务流程管理结合面向服务架构的系统架构建模过程
Fig.8 Process of system architecture modeling based on BPM+SOA

4 游乐园智能手环建模案例

按照上述建模过程, 本案例的建模步骤主要有: 第一步, 构建能力视图, 展现功能结构; 第二步, 构建业务视图, 描述节点、信息流、业务流程以及规则; 第三步, 构建服务视图, 体现系统功能的参与者、行动、服务以及交互关系; 第四步, 构建数据视图, 描述业务数据间关系。其中, 业务视图与服务视图将重点体现前文提及的元模型映射的方法。

4.1 能力视图

游乐园智能手环适用于覆盖范围广、人口流动性大、所含子项目众多的区域, 如迪士尼乐园, 其目标用户主要是园区内的游客。其设计初衷在于引导游客参与故事情景, 提升游园体验, 并设置电子围栏避免在游玩时以儿童为代表的游客在人群中与同伴走散, 还能主动就突发状况发起紧急呼救, 为园区安全提供强有力的保障。因此, 智能手环有如图 9 所示的几个主要功能。

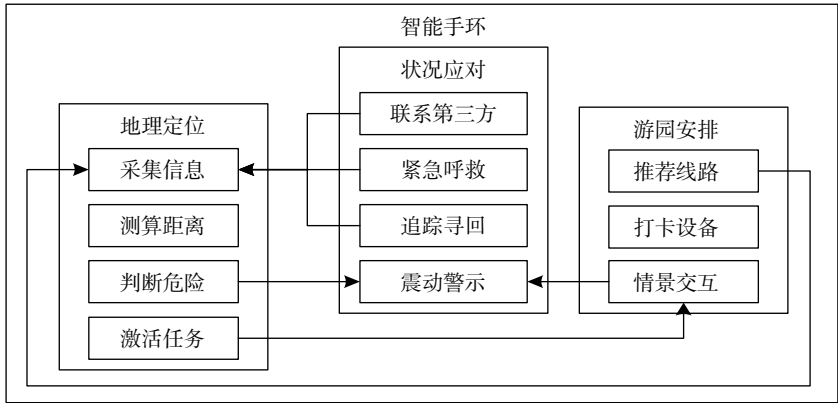


图 9 游乐园智能手环功能结构图
Fig.9 Functional structure of the smart bracelet for amusement park

4.2 业务视图

智能手环系统主要涉及游客、智能手环、游玩设备、第三方(如安全中心和急救中心)等节点, 故以“走失寻回”场景为例的信息流如图 10 所示。而图 11 所示的 BPMN 模型, 展现该场景中

的工作流程。
业务视图还包含描述该业务中所有预期事件执行条件的规则模型, 表 2 例举了“走失寻回”场景下的一条规则。

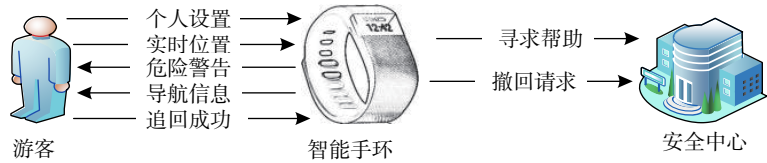


图 10 “走失寻回”场景下智能手环业务节点与信息流

Fig.10 Operational node and resource flow of the smart bracelet in “lost search” scenario

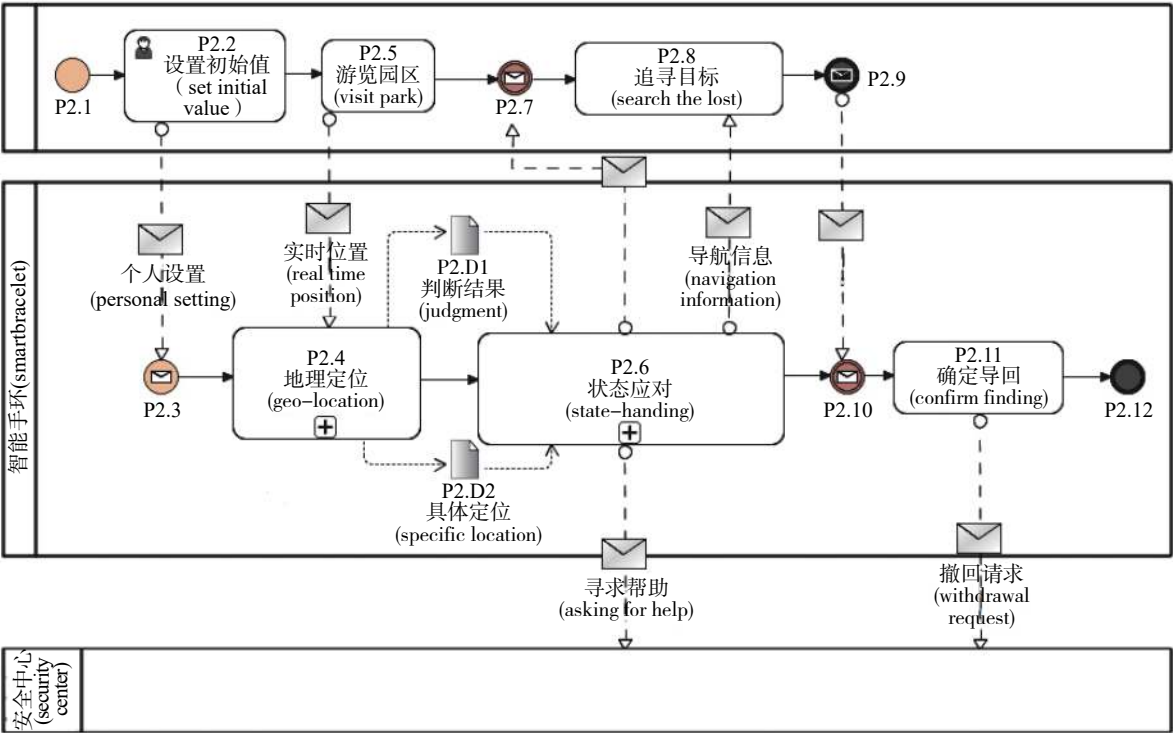


图 11 “走失寻回”场景下的 BPMN 模型

Fig.11 BPMN model in “lost search” scenario

表 2 “走失寻回”场景下的业务规则模型节选
Tab.2 Partial operational rules model in “lost search” scenario

场景	if	then
		(danger_warning.warning_method=“vibration warning”)
走失	judgment.j	and (danger_warning.danger_type=“lost”)
寻回	udgment = 1 and (asking_for_help.third_party = “security center”)	and (navigation_information.navigation_status = 1)

4.3 服务视图

首先, 将业务视图中场景“走失寻回”用服

务体系结构来展现, 用于描述各服务以及它们之间的关系, 如图 12 所示。其中, 3 个参与者 (Participant) 分别对应业务视图中的 3 个泳道, 游客、智能手环和安全中心。业务视图中的任务“设置初始值”、“游览园区”和子流程“地理定位”被打包映射成游客和智能手环间的一个服务合约 (Service Contract) “地理定位”。子流程“状态应对”和任务“追寻目标”被打包映射成游客和智能手环间的服务合约“状态应对”。任务“确认寻回”被映射成服务合约“确认寻回”。

另外，由于业务视图中，智能手环和安全中心也发生了交互，所以，将子流程“状态应对”中与安全中心相关的任务和任务“确认寻回”打包，映射成第四个服务合约“第三方帮助”。

其次，结合映射规范，定义各服务合约中的接口和角色，如图13所示，消息流都将通过接口的不同操作(operations)实现传递。由泳道映射而来的角色，相比参与者，增加了消息提供

方(provider)和消息使用方(consumer)的设定，由此体现了消息的流向。接口则由业务视图中的消息流映射而来，并在消息流的基础上，定义了信息传递的操作，为进一步实现该系统奠定基础。

业务视图中 BPMN 模型的语义延续到服务视图中，并根据服务视图中 SoaML 的语法规则调整标记元素，以此保障了语法的准确性。

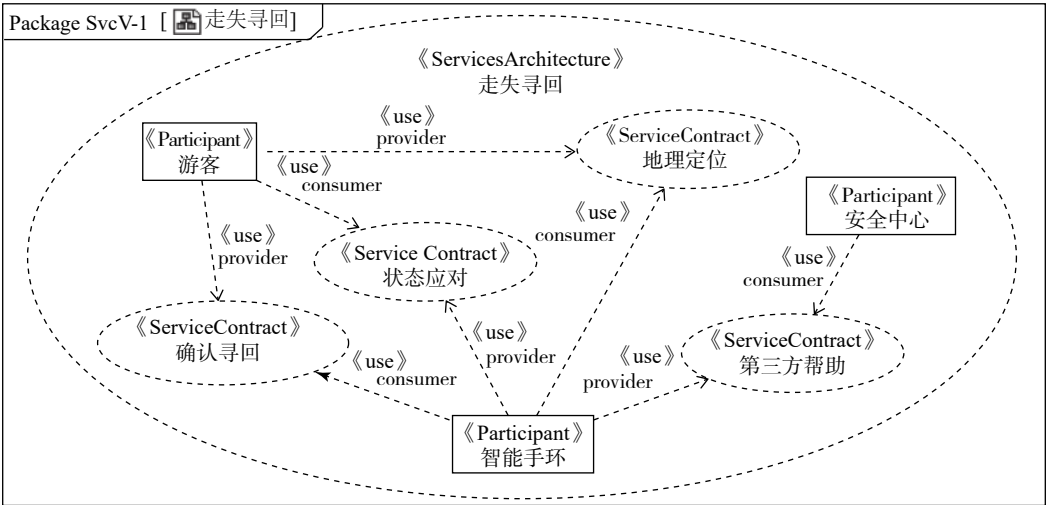


图 12 “走失寻回”场景下的服务体系结构

Fig.12 Service architecture in “lost search” scenario

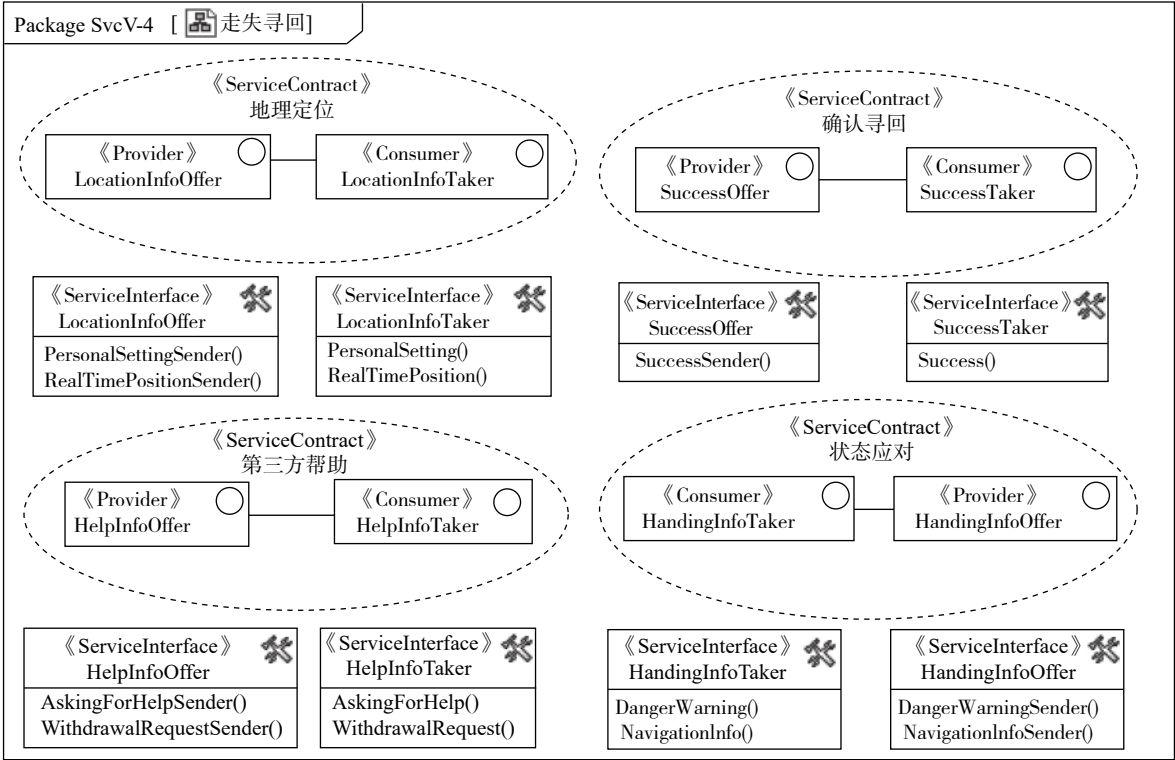


图 13 “走失寻回”场景下的服务合约

Fig.13 Service contract in “lost search” scenario

4.4 数据视图

本案例主要展现业务流程中的数据要素, 即构建逻辑数据模型。经过定义实体、定义关系和

定义属性 3 个步骤, 分解 BPMN 模型中的消息流和数据关联, 得到图 14 的逻辑数据模型。

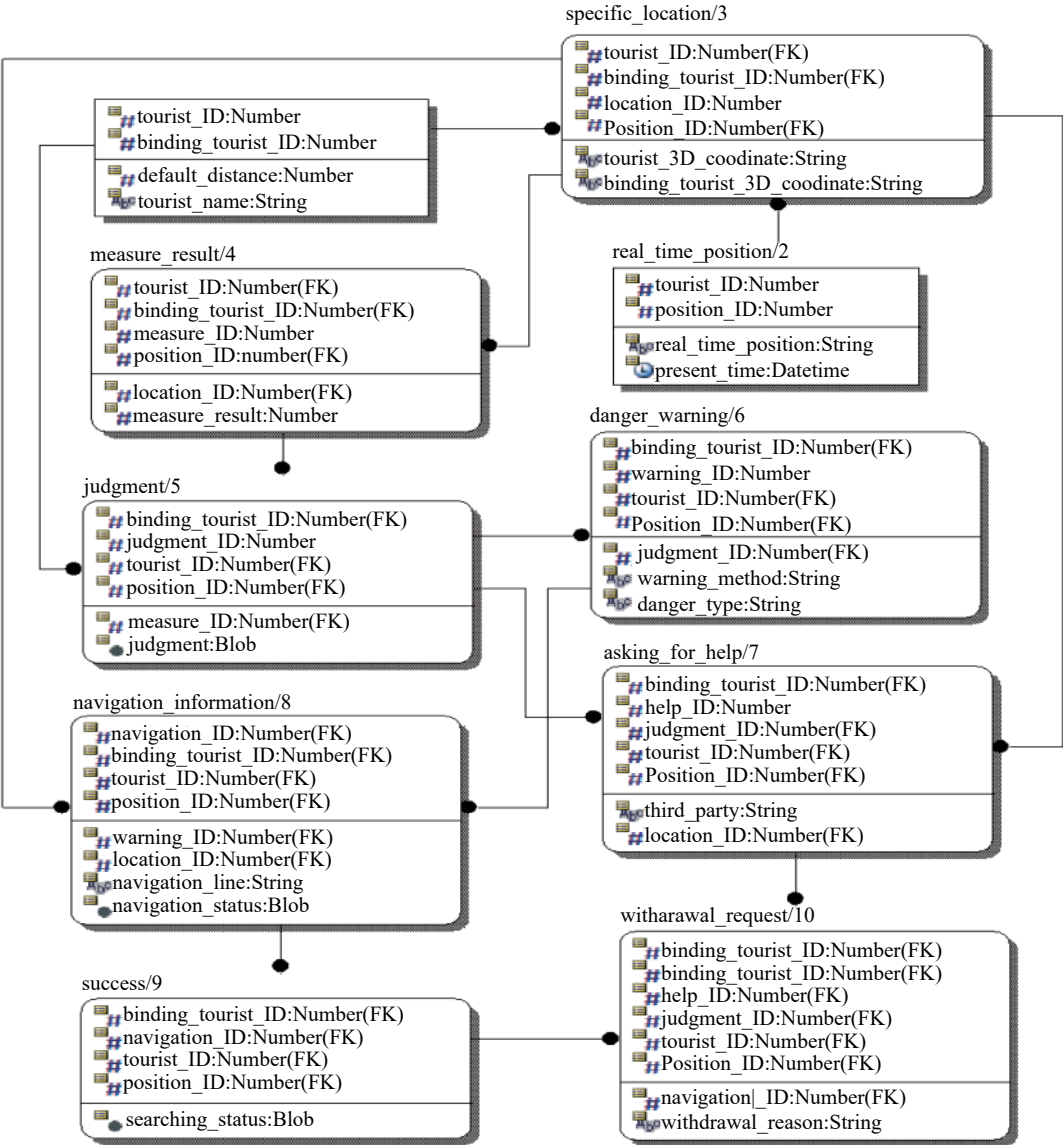


图 14 “走失寻回”场景下的逻辑数据模型

Fig.14 Logic data model in “lost search” scenario

5 结束语

本文所提出的基于元模型映射的 BPM+SOA 系统架构建模方法, 为面向服务业务流程建模理论与建模语言的融合提供了一种思路, 促进了系统架构设计过程中 CIM 到 PIM 的转换。另外, 将语义层的本体元模型和语法层的标记元模型相结合, 有助于维持系统架构不同视图中的语义一致性和语法准确性。本文的方法还存在的局限: 对于本体元模型和标记元模型的定

义有待完善; 构建系统架构本体元模型的原则和标准尚待探索; 模型间的映射规则尚待完善, 并实现自动映射和模型转换。以此作为进一步研究方向, 最终期望实现 BPM+SOA 系统架构建模的自动化。

参考文献:

[1] 王志宝, 刘丽, 常原飞, 等. 基于模型驱动的省级林业有害生物灾害监测与预警平台[J]. 东北林业大学学报, 2013, 41(7): 155-159.

- [2] LYKINS H, FRIEDENTHAL S, MEILICH A. Adapting UML for an object oriented systems engineering method (OOSEM)[J]. INCOSE International Symposium, 2000, 10(1): 490–497.
- [3] BALMELLI L, BROWN D, CANTOR M, et al. Model-driven systems development[J]. *IBM Systems Journal*, 2006, 45(3): 569–585.
- [4] DORI D. Object-process methodology: a holistic systems paradigm[M]. New York: Springer-Verlag, 2002.
- [5] WEISS K A, DULAC N, CHIESI S, et al. Engineering spacecraft mission software using a model-based and safety-driven design methodology[J]. *Journal of Aerospace Computing, Information, and Communication*, 2006, 3(11): 562–586.
- [6] BEIMBORN D, JOACHIM N. The joint impact of service-oriented architectures and business process management on business process quality: an empirical evaluation and comparison[J]. *Information Systems and e-Business Management*, 2011, 9(3): 333–362.
- [7] 王玉娟. 基于 SOA 的科技管理 BPM 平台设计与实现[J]. *计算机科学*, 2013, 40(11A): 423–425.
- [8] ZHAO D Y, YE F. ERP system implementation based SOA under SaaS model[J]. *American Journal of Industrial and Business Management*, 2014, 4(7): 355–359.
- [9] ALAM K A, AHMAD R, AKHUNZADA A, et al. Impact analysis and change propagation in service-oriented enterprises: a systematic review[J]. *Information Systems*, 2015, 54: 43–73.
- [10] LAUMER S, MAIER C, ECKHARDT A. The impact of business process management and applicant tracking systems on recruiting process performance: an empirical study[J]. *Journal of Business Economics*, 2015, 85(4): 421–453.
- [11] HACHICHA M, FAHAD M, MOALLA N, et al. Performance assessment architecture for collaborative business processes in BPM-SOA-based environment[J]. *Data & Knowledge Engineering*, 2016, 105: 73–89.
- [12] LIU H, MA Z Y, SHAO W Z. Graph transformation based description language for model refactorings[J]. *Journal of Software*, 2009, 20(8): 2087–2101.
- [13] CZARNECKI K, HELSEN S. Classification of model transformation approaches[C]//Proceedings of the Workshop on Generative Techniques in the Context of Model-Driven Architecture. New York: ACM Press, 2003: 217–219.
- [14] 曾一, 周吉, 孙政, 等. 支持 MDA 的设计模式建模与模型转换方法研究[J]. *计算机工程与应用*, 2012, 48(1): 76–80.
- [15] 苏红军, 闫云山, 尤振华. UML 模型到 N 层 Web 模型的模型驱动转换[J]. *计算机应用*, 2014, 34(4): 1161–1164.
- [16] KÜSTER J M. Definition and validation of model transformations[J]. *Software & Systems Modeling*, 2006, 5(3): 233–259.
- [17] HENDERSON-SELLERS B. Bridging metamodels and ontologies in software engineering[J]. *Journal of Systems and Software*, 2011, 84(2): 301–313.
- [18] BÉHÉ F, GALLAND S, GAUD N, et al. An ontology-based metamodel for multiagent-based simulations[J]. *Simulation Modelling Practice and Theory*, 2014, 40(1): 64–85.
- [19] 闫雪锋, 段国林, 戴怡. 机械产品虚拟样机本体元模型转换方法[J]. *机械设计*, 2016, 33(9): 25–28.
- [20] 何啸, 麻志毅, 邵维忠. 一种面向图形化建模语言表示法的元模型[J]. *软件学报*, 2008, 19(8): 1867–1880.
- [21] PARDILLO J, CACHERO C. Domain-specific language modelling with UML profiles by decoupling abstract and concrete syntaxes[J]. *Journal of Systems and Software*, 2010, 83(12): 2591–2606.
- [22] 刘辉, 麻志毅, 邵维忠. 元建模技术研究进展[J]. *软件学报*, 2008, 19(6): 1317–1327.
- [23] GERBER A, KOTZÉ P, VAN DER MERWE A. Towards the formalisation of the TOGAF content metamodel using ontologies[C]//Proceedings of the 12th International Conference on Enterprise Information Systems. Funchal, Madeira, Portugal: Springer, 2010: 54–64.

(编辑: 石 瑛)