

文章编号: 2095-2163(2022)10-0189-12

中图分类号: TP391.9

文献标志码: A

# 基于扩展 BPMN 的“家园互动”式儿童健康管理系统架构

王新康<sup>1</sup>, 倪枫<sup>1</sup>, 刘姜<sup>1</sup>, 杨帆<sup>2</sup>, 郭悦<sup>3</sup>

(1 上海理工大学 管理学院, 上海 200093; 2 上海市金山区阳光城幼儿园, 上海 200540;

3 河南理工大学 财经学院, 河南 焦作 454000)

**摘要:** 针对现在对业务流程管理(BPM)与面向服务架构(SOA)相融合使用的研究不足,基于模型驱动架构(MDA)四层次架构提出一种BPM与SOA相互映射融合的建模方法。根据BPM和SOA的特点和优势,以及BPMN和SoaML的语法规则,通过在语义层建立本体元模型和在语法层建立标记元模型与服务元模型,并分别在语义层和语法层定义模型间映射规则,实现BPM和SOA相互融合。最后通过一个“家园互动”式儿童健康管理系统的建模案例,验证建模方法和映射规则的准确性和有效性。

**关键词:** 业务流程管理; 面向服务架构; 模型驱动架构; BPMN; 健康管理系统

## The “Home-Kindergarten Interaction” child health management system architecture based on extended BPMN

WANG Xinkang<sup>1</sup>, NI Feng<sup>1</sup>, LIU Jiang<sup>1</sup>, YANG Fan<sup>2</sup>, GUO Yue<sup>3</sup>

(1 Business School, University of Shanghai for Science and Technology, Shanghai 200093, China;

2 Shanghai Sunshine City Kindergarten, Shanghai 200540, China; 3 School of Finance and Economics,

Henan Polytechnic University, Jiaozuo Henan 454000, China)

**[Abstract]** In view of the insufficient research on the integration of business process management (BPM) and service-oriented architecture (SOA), a modeling method for mapping and integrating BPM and SOA is proposed based on the four-level architecture of model-driven architecture (MDA). According to the characteristics and advantages of BPM and SOA, as well as the syntax and rules of BPMN and SoaML, the ontology meta-model is established in the semantic layer, the tag meta-model and service meta-model are established in the syntax layer, and the mapping rules between the models are defined in the semantic layer and the syntax layer, so as to realize the integration of BPM and SOA. Thereafter, the accuracy and effectiveness of the modeling method and mapping rules are verified by a modeling case of “home-kindergarten interaction” child health management system.

**[Key words]** business process management; service-oriented architecture; model-driven architecture; BPMN; health management system

## 0 引言

近年来,信息化发展逐渐成为国内企业的一个发展方向。相应地,业务流程管理(Business Process Management, BPM)和面向服务的架构(Service-Oriented Architecture, SOA)等信息系统也随着计算机技术的迅猛发展不断由概念、理论的研究而陆续经由项目研发、并最终转化为实用系统。众所周知,目前的市场环境变化迅速,企业之间的竞争日益激烈,企业为了确保自身生存就必然要提升工作效率,所以在不久的将来,数字系统也将成为各家企业生

存发展中不可或缺的重要组成部分。

模型驱动架构(Model Driven Architecture, MDA)在2001年7月由OMG发布,其核心思想在于先构建计算无关模型(Computation Independent Model, CIM),随后形成了平台无关模型(Platform Independent Model, PIM),再设计平台相关模型(Platform Specific Model, PSM)。其中,元对象设施(Meta Object Facility, MOF)是MDA的核心技术之一。MOF是OMG为帮助销售商、开发者和用户更好地使用元模型和元数据技术而制定推出的技术规范,并已于2005年成为ISO标准<sup>[1]</sup>。

**基金项目:** 国家自然科学基金(61403255, 11701370); 上海市“系统科学”高峰学科建设项目; 上海市“大学生创新创业训练计划”资助项目。

**作者简介:** 王新康(2000-),男,本科生,主要研究方向:系统建模架构;倪枫(1982-),男,博士,副教授,主要研究方向:系统科学、系统分析与集成;刘姜(1983-),女,博士,副教授,主要研究方向:复杂系统理论与方法、符号代数计算;杨帆(1988-),女,本科,幼儿二级教师,主要研究方向:幼儿健康管理。

**通讯作者:** 倪枫 Email: nifeng@usst.edu.cn

收稿日期: 2022-05-18

目前,国内外的众多学者都在致力研究如何拓展 BPMN 使其更有利于企业的后期使用。代飞等人<sup>[2]</sup>使用 Petri 网定义 BPMN2.0 编排的语义,并以此分析 BPMN2.0 的编排结构和控制流错误。陈蕾等人<sup>[3]</sup>融合 if-else 语句以及 IDEF 建模方法与 BPMN 模型建立元模型映射,实现各描述模型之间语义、粒度的对齐。Valderas 等人<sup>[4]</sup>通过定义一种允许在 BPMN 模型中创造微服务组合的方法,最终实现组合的整体图和拆分图共存。Zafar 等人<sup>[5]</sup>提出了一种支持通过 BPMN 对交换和转换过程进行建模的框架,将 BPMN 模型转换为 SoaML 模型后生成可执行 Java Web 服务,实现从 BPMN 模型自动生成可执行 Web 服务。

随着 SOA 技术的日臻成熟,有效运用 SOA 也逐渐成为了一个重要的研究方向。陈娜等人<sup>[6]</sup>针对传统 SOA 效率低下的不足,将 Petri 网应用于 SOA 服务识别中。王畅<sup>[7]</sup>针对传统的工作流管理系统设计了一种基于 SOA 的工作流管理系统的 5 层框架。Kumar 等人<sup>[8]</sup>提出一种基于 Petri 网的表示依赖关系规范的 SOA 的依赖分析,并在概念层次上对依赖关系进行建模。

同样地,MDA 的模型间也有多种转换,当下的研究主要集中在 PIM 至 PSM 的转换与 PIM 至 CIM 的转换方面,而 CIM 至 PIM 转换的相关研究较少。曹晓夏等人<sup>[9]</sup>通过将 CIM 特征分层得到层次特征需求模型,并将模式分为不同层级形成分层体系结构,通过识别特征是在哪个层级发生变化,从而实现 PIM 的相应变化。Li 等人<sup>[10]</sup>通过扩展 Petri 网模型元素作为桥梁连接 CIM 模型和 PIM 模型。Betari 等人<sup>[11]</sup>提出一种使用 UML 的核心建模概念将 CIM 转化为 PIM 的方式。还有一些学者借助于 MDA 架构在不同模型间形成映射。Wang 等人<sup>[12]</sup>采用 MDA 方式实现模型转化,将 UML 模型映射到基于 Web Service 模型中。

近年来,已有为数可观的国内外学者相继开始探寻 BPM 和 SOA 相融合的方式。邓子云等人<sup>[13]</sup>运用 SOA 和 BPM 结合架构了第三方物流信息系统集成平台。尹裴等人<sup>[14]</sup>提出以提升信息系统柔性为方向,采用 BPM 和 SOA 相结合的方式来实现。倪枫<sup>[15]</sup>提出基于 TOGAF 的外层和内层迭代建模方法,并在内层进一步探讨了 BPM+SOA 建模方法。李润晔等人<sup>[16]</sup>借助于模型间映射将 BPM 和 SOA 相结合来进行建模架构。Hachicha 等人<sup>[17]</sup>提出一种面向服务架构中协同业务流程的分析和评估方

法,以保持企业在竞争市场中的竞争力。

基于前人对于 BPMN 和 SOA 的技术成果,本文拟使用 MDA 建模架构和 MOF 元模型规范对 BPMN 与 SoaML 之间互相映射的方式和规范展开研究。

## 1 业务流程主流建模方法

### 1.1 业务流程建模与标注

业务流程建模与标注 (Business Process Model and Notation, BPMN)<sup>[18]</sup>提供了一套易于被所有业务用户理解的符号,在业务流程设计和流程实施间构建了标准化桥梁。BPMN 基于业务流程图,通过规定和定义一些标准化图形元素,并将其进行组合可以得到一个标准的 BPMN 流程<sup>[19]</sup>。BPMN2.0 包含 5 类元素,分别是:流程对象 (Flow Objects)、连接对象 (Connecting Objects)、泳道 (Swim Lanes)、工件 (Artifacts)、数据 (Data)<sup>[16]</sup>。

### 1.2 SOA 架构

#### 1.2.1 面向服务架构

研究可知,随着 XML 和 Web Service 等技术的快速发展,SOA 已然从概念逐渐转向于现实应用,并日渐成为软件工程的实践方法之一<sup>[20]</sup>。服务注册库、服务请求者和提供者是 SOA 包含的主要 3 个角色,三者之间的关系如图 1 所示。SOA 为企业的软件系统提供了一种通过对服务流程化来构建分布式系统的标准和方法,能够大幅度减少信息业务的冗余。SOA 具有松耦合性、粗粒度性、标准化接口等优势。

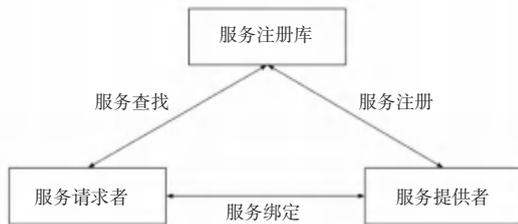


图 1 SOA 基本组成

Fig. 1 Basic composition of SOA

#### 1.2.2 面向服务架构建模语言

面向服务架构建模语言 (Service - Oriented Architecture Modeling Language, SoaML) 是对 UML2 的拓展,集成了建模功能,以支持在不同级别和不同方法上使用 SOA。SoaML 主要在 6 个主要领域拓展 UML,分别是:参与者、服务接口、服务合约、服务架构、服务数据、能力。

## 2 BPMN 模型与 SoaML 模型映射

### 2.1 BPM 和 SOA 关系

BPM 是一种业务流程协调的管理模式, 偏向于

管理方面, 而 SOA 则是面向服务的一种架构, 更偏向于是一种建模体系和思想, 两者看似毫不相关, 但各自却在经过数年发展和演变后, 迄今为止已经具备了良好的整合契机。BPM 和 SOA 的关系如图 2 所示。

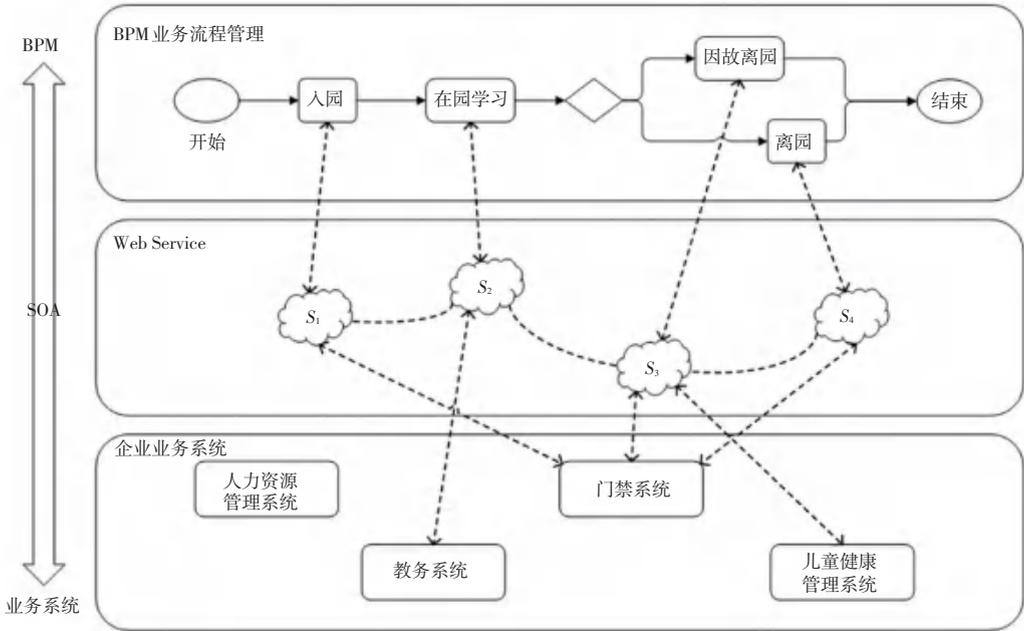


图 2 BPM 与 SOA 关系

Fig. 2 Relationship between BPM and SOA

从图 2 中可以看出在业务流程中, 每项任务和流程都对应着具体的业务系统, SOA 作为中间的桥梁可使 BPM 能够更为高效、便捷地使用各业务系统, 而 BPM 则让 SOA 可以更清晰地对各业务系统进行有机整合。BPMN 和 SoaML 分别作为 BPM 和 SOA 的一种建模语言同样具有这些特点, BPMN 使用各种图形化符号体现业务流程, 能够较为宏观地展现不同参与者之间的信息传递和工作与任务等相关业务的交互, SoaML 通过把业务活动中任务和信流传递打包封装成具有描述功能的作用的服务, 从而展现不同服务之间关系和数据流, 因此本文的研究关键就在于探索 BPMN 与 SoaML 之间相互映射和联系。

### 2.2 元模型建立及集合定义

MDA 四层元模型层次架构为元模型的扩展提供体系结构基础。MDA 四层次架构的联系和关联如图 3 所示。图 3 中,  $M_3$  层为元元模型层, 是元模型体系结构的基础结构, 用于定义元模型, 包含关联、类、属性等元素。 $M_2$  层为元模型层, 是  $M_3$  层的实例, 用于描述元模型。 $M_1$  层为模型层, 是  $M_2$  层的

实例, 用于描述数据和对象, 包含各类模型。 $M_0$  层为实例层, 一般是对象和数据, 主要体现了现实世界中的事物对象。

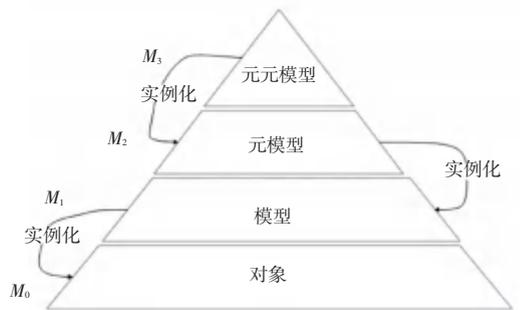


图 3 MDA 层次架构

Fig. 3 MDA levels architecture

本文的系统架构以具体对象或数据为基础建立的元模型, 这些元模型自身一般具有不同的语义, 因此首先要在语义层将 BPMN 和 SoaML 元模型映射至同一个元模型中。系统架构中所使用的 2 种建模语言均遵循一定的语法, 故稍后在语法层定义映射规则并以该映射规则根据语义层的映射结果在语法

层实现模型间映射。在映射前,先根据一定的标准对元模型进行定义,本文构建的元模型包含有元素、属性以及关系,其中属性一般是元素在建模过程中所赋予的功能和在建模中的表现方式,而关系包含元素与元素和元素与属性之间的关系。

### 2.2.1 本体元模型建立及集合定义

为了将系统架构所使用的2种建模语言建立起宏观联系,首先在语义层对整个系统构建本体元模

型。考虑到本文主要研究BPMN和SoaML之间的映射关系,所以根据整个系统需求和运行流程以及2种模型的特点选择合适的本体元素。其次,根据各元素在建模过程中所需要的功能和表现方式赋予相应属性,有助于定义映射规则使2种建模语言向标记元模型进行映射。最后,基于定制的语义明确2种不同建模语言之间的关系,得到本体元模型如图4所示。

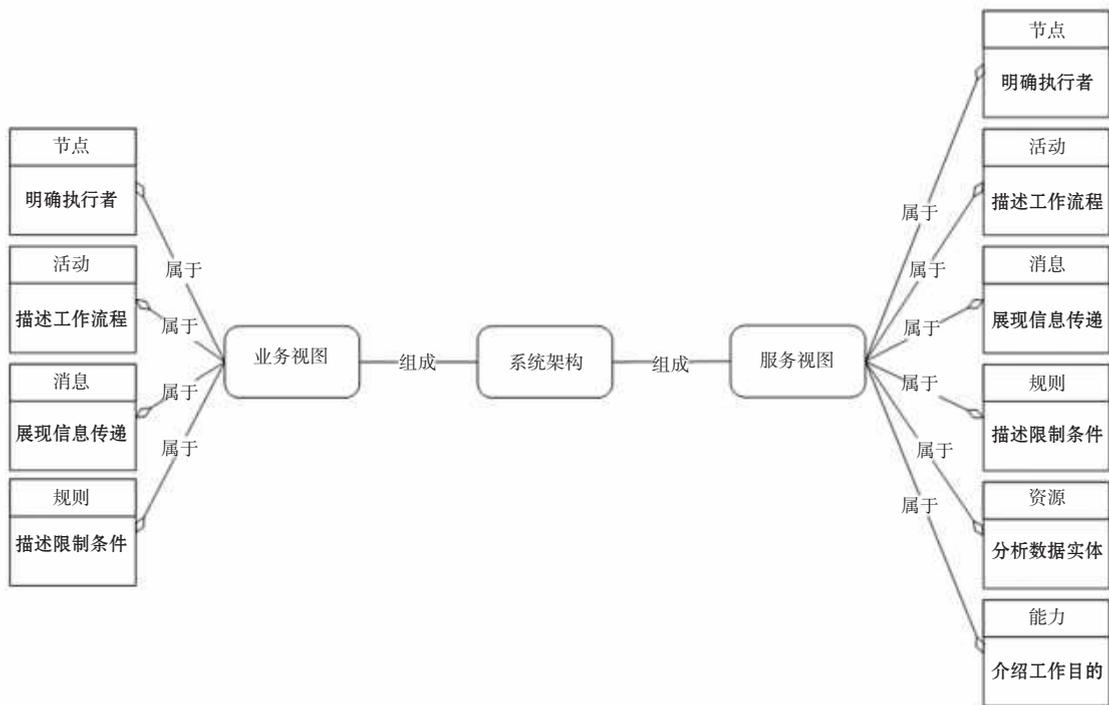


图4 系统架构的本体元模型

Fig. 4 Ontology meta-model of system architecture

从图4中可以看出,本文的本体元模型是基于BPMN的业务视图和基于SoaML的服务视图所建立的。图4中,圆角矩形框内的本体元素和矩形框内分隔上半部分为本体元素;属性包含矩形框内分隔下半部分的功能属性;“属于”、“组成”以及矩形框内的分隔体现了本体元素和本体元素间以及本体元素与属性之间关系。

### 2.2.2 标记元模型建立及集合定义

构建模型使用的建模语言的语法可以视为许多标记的语义所组成的集合,因此可以参照本体元模型的建模步骤对标记元模型进行建模。根据BPMN2.0包含的至关重要的5类元素,在明确其在建模过程中的具体功能属性后,即可推得BPMN元模型,如图5所示。

本文的标记元模型基于BPMN模型建立,图5中圆角矩形框内的标记元素和矩形框内分隔上半部

分是标记元素;属性包含了矩形框内分隔下半部分的功能属性;“属于”、“组成”以及矩形框内的分隔则表明了标记元素之间和标记元素与属性之间的关系。

### 2.2.3 服务元模型建立及集合定义

类似于标记元模型,服务元模型的建模步骤也可以类比本体元模型的建模步骤。研究可知SoaML主要在6个领域拓展了UML2,所以根据这6个主要领域提取适合的服务元素,当明确各服务元素在建模过程中的具体功能属性后,即可得到SoaML元模型如图6所示。

本文的服务模型基于SoaML模型建立。图6中,圆角矩形框内的服务元素和矩形框内分隔上半部分为服务元素;属性包含矩形框内分隔下半部分的功能属性;“属于”、“组成”以及矩形框内的分隔则表明了服务元素之间和服务元素与属性之间的关系。

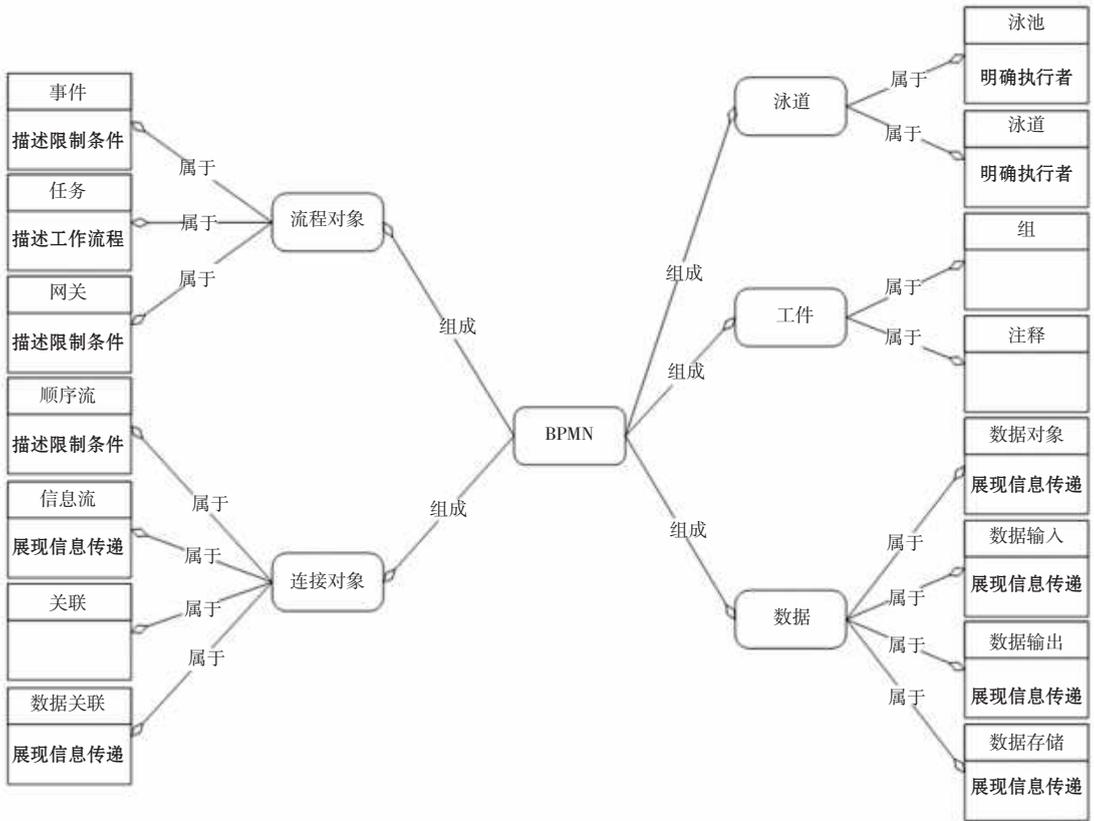


图 5 BPMN 元模型

Fig. 5 BPMN meta-model

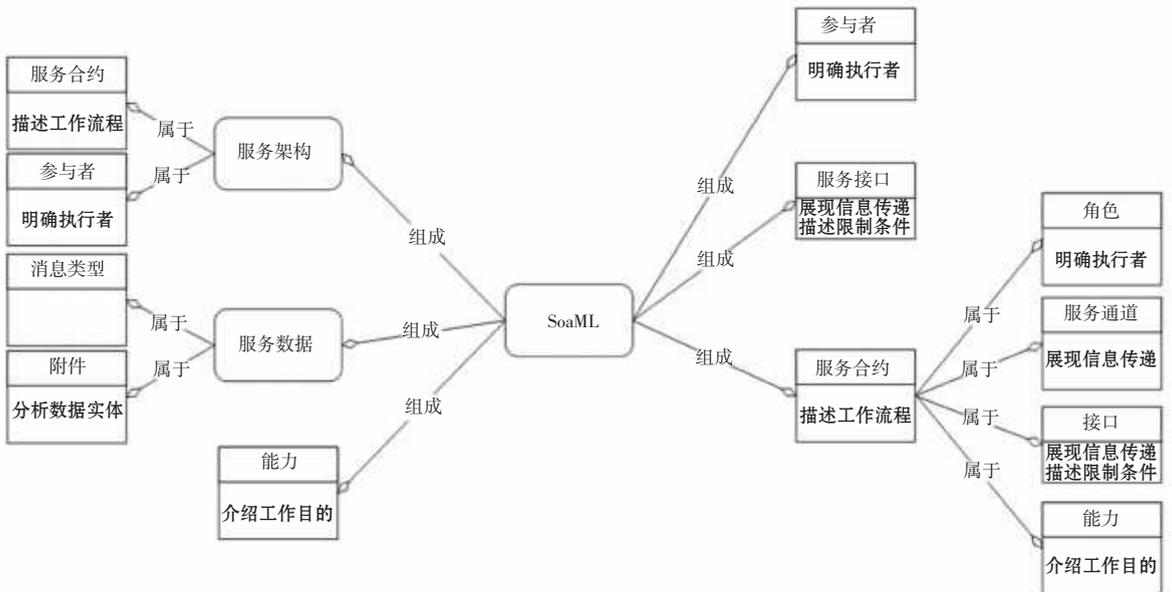


图 6 SoaML 元模型

Fig. 6 SoaML meta-model

### 2.3 模型间映射

BPMN 的标记元模型和 SoaML 的服务元模型之间的映射思路如图 7 所示。本文将模型间映射分为 2 部分。首先在语义层根据元模型的语义定义映

射规则, 将标记元模型和服务元模型中具有相同或类似的属性的标记元素  $N$  和服务元素  $S$  映射到本体元模型中具有同样属性的本体元素  $O$ 。接下来, 基于前一步的映射标记元素  $N$ 、服务元素  $S$ 、本体元

素  $O$  都有相同功能属性  $A$ , 以本体元素  $O$  及其功能属性  $A$  作为映射依据, 在语法层定义映射规则, 使

标记元素  $N$  和服务元素  $S$  之间在语法层中形成映射。

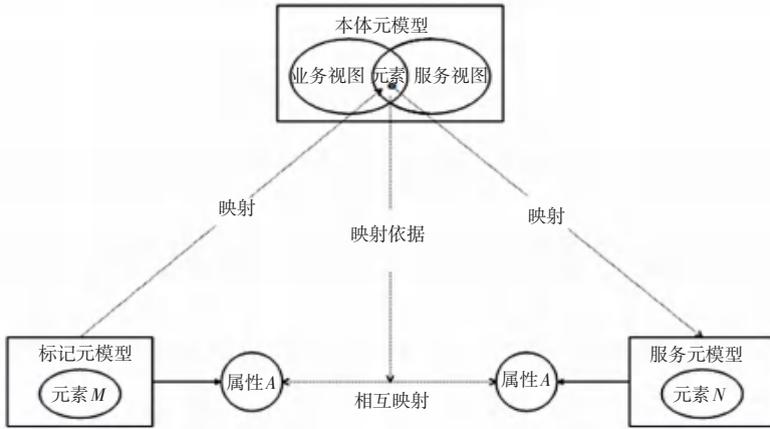


图7 元模型映射

Fig. 7 Mapping of meta-model

2.3.1 语义层定义及映射规则

语义为建模语言表达的概念和建模语言的符号之间架起了连接的桥梁, 具体表示了表达式的含义。语义主要包含2种: 静态语义和动态语义。其中, 静态语义是在模型构建、系统运行之前就已经确定的语义, 即模型的符号本身就可以表达识别相应的语义; 动态语义是在系统运行期间才能够识别认定的语义, 包括一致性和合法性等。在大部分建模语言中, 类型相关的信息属于静态语义, 并通常与一些语义规则的约束条件相对应, 用于规定一段语句或一个模型是否符合定义, 而动态语义一般来说约束上相对较弱, 在编译代码或构建模型时不受动态语义规则限制, 因此在建模时无法判断是否合法, 只有在模型运行时才会知道是否出现错误。本文研究建立的系统架构中的元模型都采用静态语义, 且使用的建模语言都具有明确的定义和功能, 因此在建立模型时就能够知晓模型是否符合建模规范和语义规

则, 而在正确建立模型后也可确保系统正常运行。

参照前文所提及的模型映射步骤, 首先对整个系统架构在语义层进行映射。元模型之间的映射中都需要遵守一定语义规则, 以确保模型映射后仍然符合一定语义且具有合法性, 本文针对提出的模型和系统架构定义如下语义映射规则:

- (1) 服务元模型和标记元模型的映射对象必须是元素。
- (2) 映射的服务元素或标记元素和被映射的本体元素必须具有属性。
- (3) 映射的服务元素或标记元素的属性和被映射的本体元素的属性必须相同或相似。

基于上文定义的语义映射规则, 首先按照不同功能属性将本体元模型中业务视图和服务视图中具有相同属性的本体元素分类, 再根据语义规则将标记元模型和服务元模型中的元素映射到本体元模型中, 映射后模型如图8所示。

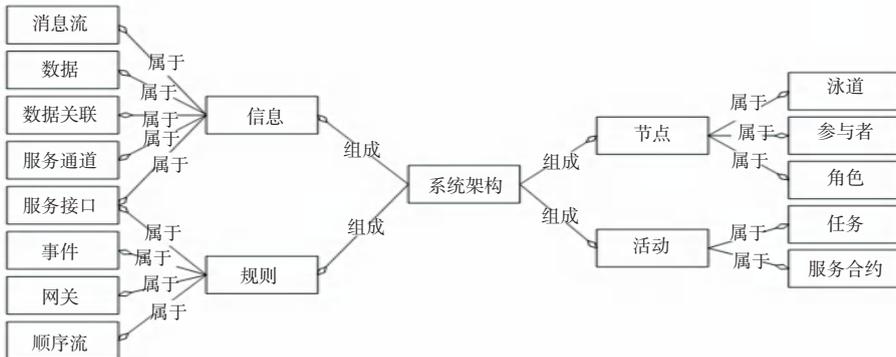


图8 语义层映射

Fig. 8 Mapping in semantic layer

本体元模型的本体元素中,“节点”、“活动”、“信息”、“规则”的属性在标记元模型和服务元模型中都有相应的元素具有该功能属性。

### 2.3.2 语法层定义及映射规则

语法是指建模语言具有精确定义的形式,语法同样可以分为 2 类:抽象语法和具体语法。其中,抽象语法是对建模语言本质性的描述,目的在于描述模型元素、元素之间的关系和联系以及模型合法性约束,通常关注的是建模语言概念和元素的抽象表达,并不关注具体将如何呈现给用户。以前述提出的标记元模型为例,模型中的标记元素仅仅是抽象表达,未能赋予相应的图形符号表达,而研究中可以通过功能属性对建模语言定义规则用于判断模型在抽象语法方面是否合法,以此控制非法模型出现。

通过抽象语法,仅仅可以知道模型的一些抽象概念和约束条件,无法进行实际建模,此时需要借助具体语法进行建模。具体语法包含 2 种形式。一种是基于文字和数学符号的文本语法,另一种是基于图形化符号的图形化语法。前者使用通俗易懂的符号,较容易理解;后者使用图形展示,模型元素之间的关系、事件或任务之间的联系和事件顺序可以较为直观地做出展示。两者各有利弊,由于本文选取 BPMN 和 SoaML 两种已经普遍使用的建模语言,因此本文主要讨论这 2 种语言中主要使用的图形化语法。表达语法的图形多种多样,大多使用的是方框、圆、实线箭头、虚线箭头等标准图形以及如信封等可以直接代表某一种元素的特殊图形,而且还可以根据不同的具体功能对图形进行一定修改,部分模型元素和图形化语法见表 1。

表 1 模型元素及其图形化语法

Tab. 1 Model elements and graphical syntax

模型元素	泳道	开始事件	结束事件	网关	顺序流
图形化符号					
模型元素	消息流	任务	消息	数据对象	子流程
图形化符号					
模型元素	服务合约	参与者	服务接口	角色	服务通道
图形化符号					

研究至此,还需要基于语法层定义一个映射规则,可以将模型之间的映射具体化,明确模型具体的

映射方法和映射规范。这里给出了一个定义为六元组的模型映射规则,重点展示了将 BPMN 元模型映射到 SoaML 模型的规则,记为:

$$MF = (B, S, C, M, A, f) \quad (1)$$

其中,  $B$  表示 BPMN 模型的标记元素;  $S$  表示 SoaML 模型的服务元素;  $C$  表示映射条件;  $M$  表示标记元素的属性;  $A$  表示服务元素的属性;  $f$  为函数,表示模型间的映射关系。

在本文的模型中,函数  $f$  表示如下的映射关系:  $f: B \times M \times C \rightarrow S \times A$ , 表明 BPMN 模型中的标记元素在满足一定条件时,转换成 SoaML 模型中的服务元素。例如  $f: B_1 \times USER \times C_1 \rightarrow S_1 \times USER$ 。其中,映射条件  $C_1$  是该标记元素的功能属性和服务元素的功能属性相同或类似且可以在本体元模型中找到具有相同或类似功能属性的本体元素,主要表示 BPMN 模型中的标记元素  $B_1$  和  $B_1$  的功能属性  $USER$  在满足条件  $C_1$  时,可以映射成 SoaML 模型中的服务元素  $S_1$  和  $S_1$  的功能属性  $USER$ 。

根据上文定义的映射规则,BPMN 中部分标记元素可以与 SoaML 中部分服务元素相互映射,相互映射关系见表 2。

表 2 BPMN 与 SoaML 映射关系

Tab. 2 Mapping relationship between BPMN and SoaML

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

从以上的建模过程和映射规则中不难看出,可以相互映射的 2 种不同建模语言中参与映射的元素一定存在相同或类似的功能属性,而且在本体元模型中也能够找到具有相同功能属性的元素。因为本文基于 BPMN 和 SoaML 两种建模语言结合建立本体元模型,保证了语义一致性。而 BPMN 和 SoaML 两种建模语言已经具有较完善的语法,提出的映射规则是基于这些语法的,所以语法的一致性也得到保证。

### 2.4 建模过程

由于整个系统建模架构在 DoDAF 的框架基础

上,因此参考 DoDAF 的多个视图并根据前文元模型的映射方法来构建整个系统模型。本文研究建立的系统架构主要使用了用例图、BPMN、SoaML,对应 DoDAF 八大视图中 SV、OV、SvcV,再根据具体需求使用其它视图。

图9是基于上文的建模过程。由图9可知,第一步,根据系统的用户和需求建立系统用例图、即系统视图;第二步,建立以 BPMN 为主要建模语言的

业务视图;第三步,将 BPMN 的标记元素按照一定的映射依据和映射规则分别在语义层和语法层进行映射,建立以 SoaML 为主要建模语言的服务视图;第四步,根据系统和用户的其它需求构建相应的其它视图;第五步,后续系统功能或其中的用户发生改变,以系统用例图为基础重复上述步骤进行迭代,完善其余视图和整个系统架构,直至达到使用者要求。

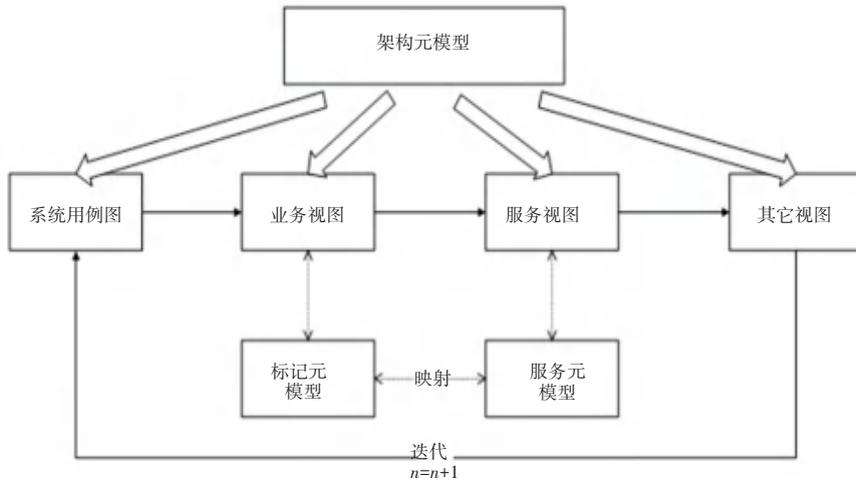


图9 建模过程

Fig. 9 Modeling process

### 3 儿童健康管理系统的建模案例

根据研究提出的系统架构建模和模型间映射关系,对家园互动方式的儿童健康管理系统进行建模。系统主要采用 BPMN 和 SoaML 两种建模语言对系统进行建模架构。对此可做分析阐述如下。

#### 3.1 用例图建模

儿童健康管理系统主要包含查看儿童实时健康数据和突发情况应对两个功能。其中,查看儿童实时健康数据功能满足了家长或幼儿园教师在儿童不在身边时能随时了解儿童的实时健康状况。突发情况应对功能确保儿童在身体状况出现变化的时候能够获得及时的关注并第一时间告知幼儿园教师和家长。参与用户主要包括2种角色:家长或幼儿园教师和幼儿园负责人员。将儿童健康管理系统的建模分析后用例图详见如下。

- (1)用例名称:儿童健康管理系统的。
- (2)系统角色:家长或幼儿园教师、幼儿园负责

人员。

(3)简要说明:家长或幼儿园教师是系统主要用户,具有查看儿童实时健康数据和接收突发消息的权限。幼儿园负责人具有维护儿童信息和处理应急情况的权限。

(4)前置条件:

① 家长、幼儿园教师和幼儿园负责人员都进行了注册和审核,有权访问儿童健康管理系统的。

② 手环获取的身体数据直接导入儿童健康管理系统的。

(5)后置条件:可以判断身体数据是否异常,及时做出反应。

儿童健康管理系统的用例图的简单描述见图10。由图10中可以看出,幼儿园负责人员主要起着维护儿童信息和处理应急情况的作用,而家长或幼儿园教师可以通过登录进入儿童健康管理系统的查看儿童当前健康情况,并当幼儿遇到应急情况时就会接收到相关信息。

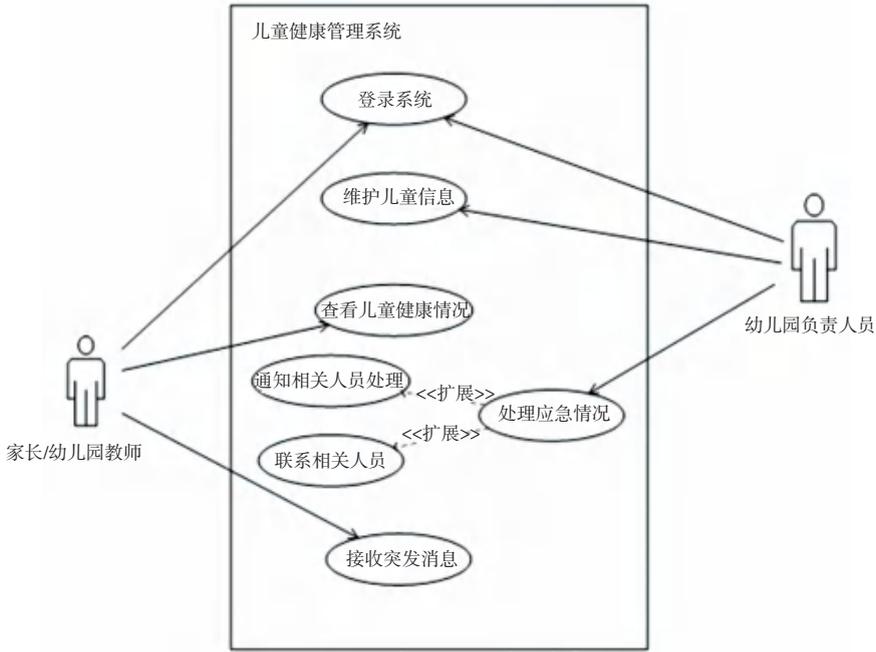


图 10 儿童健康管理系统用例图

Fig. 10 Use case diagram of child health management system

### 3.2 BPMN 模型建模

整个系统基于具有实时监控健康数据的智能手环进行建模, 主要涉及用户端、手环端和幼儿园端等, BPMN 模型如图 11 所示, 展现整个系统运行流程。用户主要是幼儿园教师和儿童家长, 在需要时可通过登录系统并输入儿童的相关信息查看儿童实时健康情况, 如果儿童遇到突发情况时, 也会通过短信或微信消息及时通知家长和幼儿园教师。同时,

一旦手环端监测到儿童健康数据有异常就会迅速响应, 通知幼儿园相关人员第一时间做出反应, 幼儿园方将根据当日入园情况判断儿童是否在园。如果儿童在园, 园方相关人员就会立刻通知幼儿园相关医务人员前去照看并迅速通知幼儿园当班教师, 在了解具体情况后告知家长相关异常情况; 如果儿童并不在园, 幼儿园相关人员就会通知家长并与其取得联系, 在得知具体情况后通知幼儿园教师。

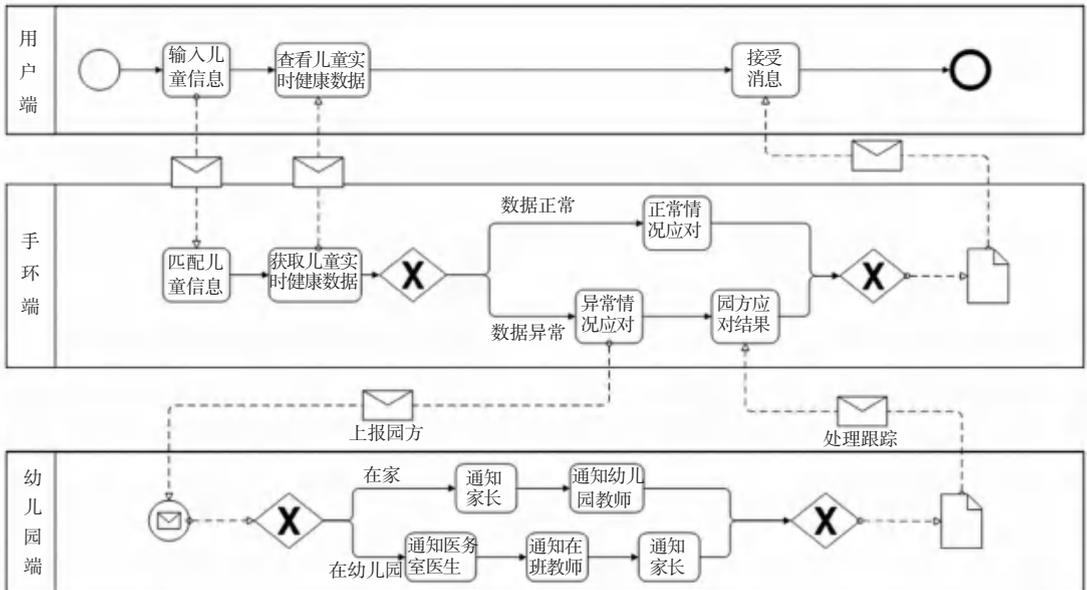


图 11 儿童健康管理系统 BPMN 模型

Fig. 11 BPMN model of child health management system

### 3.3 SoaML 模型建模

根据第2节提出的 BPMN 和 SoaML 映射方法和规则,首先定义各个服务合约中的接口和角色,如图 12 所示。每个服务接口代表一个操作,一般至少

有一个提供者和一个使用者,而提供者和使用者由简单接口定义,信息和数据则通过接口在提供者和使用者之间传递。

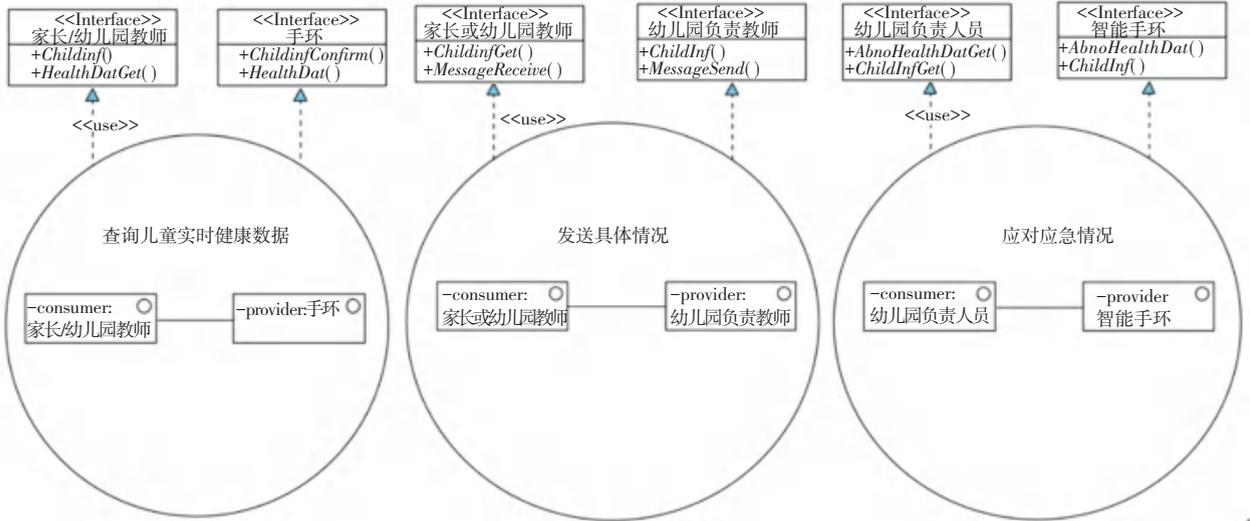


图 12 儿童健康管理系统服务接口

Fig. 12 Service interface of child health management system

紧接着,创建服务合约,如图 13 所示。服务合约中除了由 BPMN 模型中泳道映射得来的 3 个角色,还增加定义了服务提供者和服务使用者,使得服

务合约能鲜明展现服务提供者和服务使用者之间如何提供和使用服务以及信息流传递的方向。服务合约则由 BPMN 模型中任务映射而来。

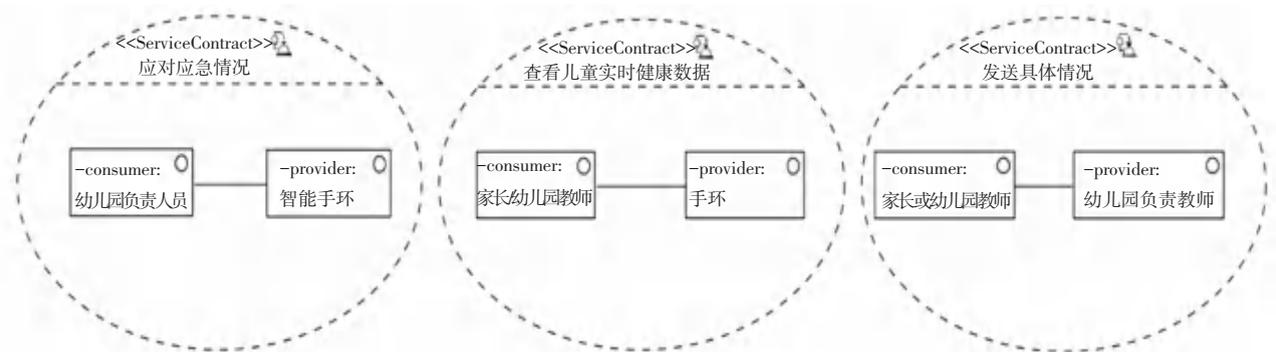


图 13 儿童健康管理系统服务合约

Fig. 13 Service contract of child health management system

以“查看儿童实时健康数据”服务为例,服务提供者“手环”链接至服务接口图中相应的简单接口“手环”;服务使用者“家长/幼儿园教师”链接至服务接口图中相应的简单接口“家长/幼儿园教师”。

及服务之间的关系和联系。

最后,根据上文绘制出的服务接口图和服务合约图,使用服务结构图展现儿童健康管理系统,如图 14 所示,用于描述儿童健康管理系统的多个服务以

在此过程中,3 个参与者分别对应 BPMN 模型中泳道“用户端”、“手环端”、“幼儿园端”。以服务合约“查看儿童实时健康数据”为例,即通过 BPMN 模型中的任务“输入儿童信息”、“匹配儿童信息”、“获取儿童实时健康数据”、“查看儿童实时健康数据”打包映射得到。

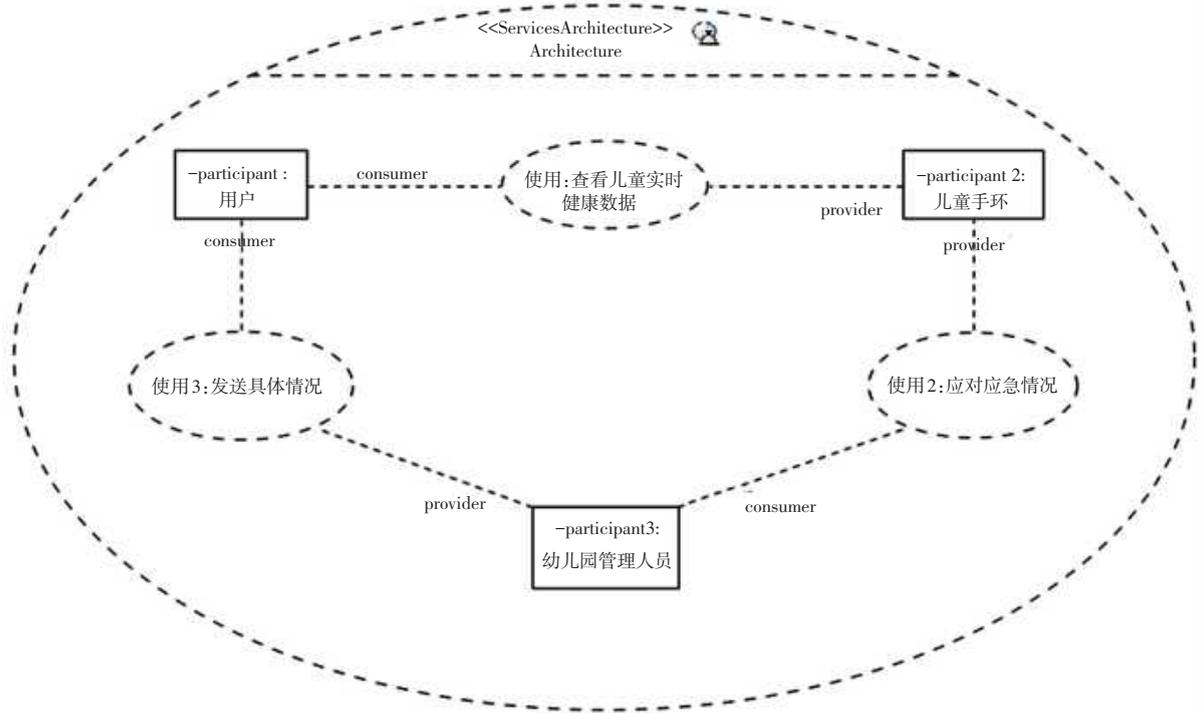


图 14 儿童健康管理系统的服务架构

Fig. 14 Service architecture of child health management system

## 4 结束语

本文运用 MDA 四层次架构和 MOF 元模型规范, 搭建出一个具有 4 个层次的架构, 定义并形成以整个系统为基础的本体元模型、以 BPMN 为基础的标记元模型、以 SoaML 为基础的服务元模型。此后, 定义映射规则明确映射规范, 将 BPMN 的元素和 SoaML 的元素之间进行映射, 形成完善的 BPM+SOA 系统架构, 达到建模的目的。

本文对于 BPMN 和 SoaML 两种建模语言以及相互之间映射做了初步研究, 但本文所给出的系统架构和映射方法仍存在一定缺陷: 输入输出操作和传递的信息流的数据类型需要进一步明确、本文并没有形成可操作的界面供用户使用、元模型的定义以及语义和语法的定义可以更加完善准确、映射规则的定义也需要进一步完善。未来还需要进一步深入研究, 从而研发出可供企业使用的技术系统。

## 参考文献

[1] OMG. Meta Object Facility (MOF) Specification Version 1.4.1 [EB/OL]. [2005-05-05]. <https://www.omg.org/spec/MOF/ISO/19502/PDF>.

[2] 代飞, 赵文卓, 杨云, 等. BPMN 2.0 编排的形式语义和分析[J]. 软件学报, 2018, 29(04): 1094-1114.

[3] 陈蕾, 倪枫. 基于 i-BPMN 的翻转课堂教学系统业务流程建模

[J]. 智能计算机与应用, 2019, 9(04): 180-186, 191.

[4] VALDERAS P, TORRES V, PELECHANO V. A microservice composition approach based on the choreography of BPMN fragments[J]. Information and Software Technology, 2020, 127: 106370.

[5] ZAFAR I, AZAM F, ANWAR M W, et al. A novel framework to automatically generate executable Web services from BPMN models[J]. IEEE Access, 2019, 7: 93653 - 93677.

[6] 陈娜, 张金娟, 刘智琼. 基于 Petri 网的 SOA 架构应用研究[J]. 移动通信, 2015, 39(06): 46-50.

[7] 王畅. 一种 SOA 的工作流管理系统的框架设计[J]. 智能计算机与应用, 2018, 8(01): 79-81, 86.

[8] KUMAR P, GUPTA R. Dependency modeling of a SOA based system through colored Petri nets[J]. Journal of Computing & Information Technology, 2016, 24(3): 253-269.

[9] 曹晓夏, 缪淮扣, 孙军梅. 一种运用模式将 CIM 转换到 PIM 的方法[J]. 计算机科学, 2007, 34(06): 265-269.

[10] LI Zonghua, ZHOU Xiaofeng, YE Zhengwei. A Formalization model transformation approach on workflow automatic execution from CIM Level to PIM level[J]. International Journal of Software Engineering and Knowledge Engineering, 2019, 29(9): 1179-1217.

[11] BETARI O, FILALI S, AZZAOU I, et al. Applying a model driven architecture approach: Transforming CIM to PIM using UML[J]. International Journal of Online Engineering (iJOE), 2018, 14(9): 170.

[12] WANG Bo, WANG Zhili, QIU Xuesong. MDA-based network management information model transformation from UML to Web Services[C]//2012 14<sup>th</sup> Asia-Pacific Network Operations and Management Symposium (APNOMS). Seoul: IEEE, 2012: 1-4.

(下转第 202 页)