

学校代码: 10246  
学 号: 17212010060

復旦大學

硕 士 学 位 论 文  
(专业学位)

人机物融合应用的工作流设计与执行  
**Workflow Design and Execution of CPSS Application**

院 系: 软件学院

专业学位类别(领域): 软件工程

姓 名: 胡超群

指 导 教 师: 沈立炜 副教授

完 成 日 期: 2019 年 09 月 09 日

## 复旦大学 学位论文独创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。论文中除特别标注的内容外，不包含任何其他个人或机构已经发表或撰写过的研究成果。对本研究做出重要贡献的个人和集体，均已在论文中作了明确的声明并表示了谢意。本声明的法律结果由本人承担。

作者签名：\_\_\_\_\_ 日期：\_\_\_\_\_

## 复旦大学 学位论文使用授权声明

本人完全了解复旦大学有关收藏和利用博士、硕士学位论文的规定，即：学校有权收藏、使用并向国家有关部门或机构送交论文的印刷本和电子版本；允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。涉密学位论文在解密后遵守此规定。

作者签名：\_\_\_\_\_ 导师签名：\_\_\_\_\_ 日期：\_\_\_\_\_

## 指导小组成员名单

赵文耘	教 授
彭 鑫	教 授
沈立炜	副教授
吴毅坚	副教授
陈碧欢	青年副研究员

# 目 录

摘 要.....	V
Abstract.....	VI
第一章 绪论 .....	1
1.1 研究背景.....	1
1.2 研究目的和意义.....	2
1.3 主要工作和创新点.....	2
1.4 篇章结构.....	3
第二章 相关工作及技术背景 .....	5
2.1 人机物融合技术.....	5
2.1.1 HomeAssistant 平台 .....	7
2.1.2 众包技术.....	7
2.2 工作流技术.....	8
2.2.1 工作流执行引擎.....	9
2.2.2 流程定义工具.....	10
2.2.3 BPMN2.0 建模标准 .....	11
2.3 面向用户的服务组合技术.....	12
2.3.1 基于工作流的静态服务组合.....	13
2.3.2 基于 SOA 架构的动态服务组合.....	14
2.3.3 基于 MashUp 的服务混搭组合 .....	15
2.4 本章小结.....	15
第三章 人机物场景案例分析与方法描述.....	16
3.1 人机物融合动机场景案例分析.....	16
3.1.1 人机物资源一体化整合需求.....	17
3.1.2 面向用户的场景活动编排需求.....	17
3.1.3 流程活动自动化执行需求.....	18
3.1.4 现有技术的挑战与不足.....	19
3.2 研究方法描述.....	20
3.2.1 人机物融合应用的研究框架.....	20
3.2.2 人机物资源仓库构建方法.....	22
3.2.3 用户流程定义建模方法.....	24
3.2.4 用户模型映射与校验方法.....	26
3.2.5 自动化与自适应执行方法.....	29

3.3 本章小结.....	31
<b>第四章 人机物融合应用的设计与实现.....</b>	<b>32</b>
4.1 人机物应用系统工具架构.....	32
4.2 人机物资源仓库构建.....	33
4.2.1 人机物资源服务抽取.....	33
4.2.2 人机物服务语义抽象.....	34
4.2.3 语义仓库最终构建.....	35
4.3 用户流程定义工具构建.....	36
4.3.1 用户流程定义工具设计.....	37
4.3.2 流程定义工具模型构建.....	38
4.3.3 用户模型映射.....	40
4.3.4 用户模型校验.....	41
4.4 运行时环境构建.....	44
4.4.1 人机物应用自动化执行.....	45
4.4.2 人机物应用自适应执行.....	46
4.5 本章小结.....	48
<b>第五章 场景实例实验与分析 .....</b>	<b>49</b>
5.1 实验工具.....	49
5.2 性能评估.....	49
5.2.1 效率.....	50
5.2.2 易用性.....	51
5.3 案例分析.....	51
5.3.1 场景实例背景.....	51
5.3.2 结果与分析.....	52
5.4 方法的局限性.....	55
5.5 本章小结.....	56
<b>第六章 总结与展望 .....</b>	<b>57</b>
6.1 工作总结.....	57
6.2 工作展望.....	58
<b>参考文献.....</b>	<b>59</b>
<b>致 谢.....</b>	<b>62</b>

## 摘 要

物联网技术逐渐已融入到人类生活的方方面面,为人类生活提供着各种智能化、自动化的设备管理体验。近年来推进信息化建设的同时,各类信息服务的“信息孤岛”被打破,进而衍生出智能空间下的人机物融合应用。然而,目前的人机物融合相关技术仍存在以下两个方面的不足。首先,人机物三元融合存在异构特性,目前存在设备资源的服务语义建模技术对物理设备进行描述,但并未有相应技术对人机物资源进行一体化描述;其次,目前的融合应用普遍基于 ECA(Event-Condition-Action)规则,融合应用模式设定固化,不支持或者难以支持用户轻松地自定义需求模式,难以满足即时性的个性化定制的需求,灵活性较差。

针对上述存在的问题,本文提出了一种人机物融合应用的工作流设计与执行的技术方案。该方案适用于比较常见的基于流程类的生活服务场景,旨在支持用户即时地自定义场景流程活动,用户可以可视化地拖拽带有抽象语义活动描述的模型图元代表活动来自定义流程,系统执行引擎自动化和自适应调度相应地最优的人机物资源来完成用户流程活动。具体而言,该方案分为三个部分:人机物资源仓库的构建阶段对人机物实例资源进行服务抽取和服务抽象,自底向上抽象构建包含人机物实例服务模型、服务能力特征模型和空间人机物实例资源关系模型的语义仓库;用户流程设计工具的构建阶段基于所提出的用户流程元模型构建用户流程设计工具模型元素,支持用户以拖拽方式对流程活动建模并进行用户模型映射和校验;运行时环境构建阶段基于扩展的 Activiti 执行引擎支持自动化并基于可用性、空间距离和用户使用习惯三个维度的服务动态绑定策略自适应调度最优的人机物资源执行完成用户自定义的流程活动。

最后,基于该方案实现了一个人机物融合应用的工作流设计与执行的系统工具。本文基于该工具在实验环境下选取真实场景进行了实验,实验结果表明,该方案能显著增强用户体验,降低用户操作难度,提高用户的时间效率的同时能有效满足用户流程类场景中地个性化定制需求。

**关键词:** 物联网; 人机物融合; 用户建模; 自动化; 自适应

**中图分类号:** TP311

# Abstract

The Internet of Things technology has gradually integrated into all aspects of human life, providing various intelligent and automated device management experiences for human life. In recent years, the development of information technology has been promoted. At the same time, the “information islands” of various information services have been broken, and the practical life applications of smart space and CPSS have been derived. However, the current technology of ternary integration of human in social space, machine in cyber space and physical objects in physical space still has two shortcomings in the following aspects. First of all, the CPSS has heterogeneous characteristics. At present, the service semantic modeling technology of devices resources describes the physical devices, but there is no corresponding technology to describe the CPSS resources in an integrated manner. Secondly, the current practical application modes are mostly based on ECA (Event-Condition-Action) rules. The human-machine-thing fusion mode is set to solidify, which does not support or is difficult to support users to easily customize the demand mode, and it is difficult to meet the impersonal customization requirements of instantness, and the flexibility is poor.

In view of the above problems, this paper proposes a solution for workflow design and execution of CPSS application. The solution is applicable to the more common process-based life service scenarios, allowing users to customize the scenario process activities in real time. The user can visually drag and drop model entities with abstract semantic activity descriptions to represent the activity customization process. The system's execution engine automation and adaptive scheduling correspondingly optimal resources of human, machine and physical object to complete user process activities. In general, the solution is divided into three stages: the construction stage of the CPSS resources depository performs service extraction and service abstraction from human-machine-thing's instance resources, and the bottom-up abstract construction includes semantic depository of service capability feature model, spatial human-machine-things's instance resource relationship and the instance service model. The construction stage of the user process design tool builds the user process design tool model element based on the proposed user process metamodel, and supports the user to model the process activity in a drag-and-drop manner and supports user to perform model mapping and verification. The construction stage of runtime environment is

based on an extended Activiti execution engine that supports user define process activities automates and adaptively schedules optimal human-machine-thing's resource to execute based on the policies of availability, spatial distance, and user usage habits.

Finally, based on the solution, we realized a tool of workflow design and execution of CPSS application. Based on the tool, the actual scene is selected in the laboratory environment to experiment. The experimental results show that the solution can significantly enhance the user experience, reduce user operation difficulty, improve user time efficiency, and effectively meet the needs of personalized customization in user process scenarios.

**Keywords:** Internet of Things; CPSS; User Modeling; Automation; Adaptation;

**Class Number:** TP311



# 第一章 绪论

## 1.1 研究背景

物联网（IoT）是包含电子、软件、传感器和执行器的物体集合组成的网络[1]，代表了可以互相连接的对象。它允许这些物体连接、交互和交换数据，用户，传感器和网络生成大量数据[2]。物联网是一种新颖的范例，正在塑造未来互联网的发展，根据 Gartner 报告（<https://www.gartner.com>）的预测，到 2020 年将有大约 210 亿个可连接的设备将被使用[3]。它通过提高舒适度和提供节省时间的应用极大地影响人类生活[4]。物联网的优势是家庭自动化，更好的医疗保健，智能交通和智能城市的应用[5]。当前的物联网还没有形成一个被广泛认可的体系结构[6]，但大体而言，物联网架构可以由三层组成，分别是感知层，网络层和应用层。其中应用层主要涉及软件服务，用于实现智能家居，城市，学校等的应用[7]。

人机物融合是物联网领域下新兴的一种社会物理信息系统（CPSS）[8]，人机物的三元融合指的是将存在于信息、社会、物理三个相互独立的空间之中的人、机（信息服务）、物三者融合，表现为人、信息服务或带有信息服务的超集的机器、智能设备通过显式或隐式的融合范式，达到人机物智能的协作与增强。然而，物联网的三层体系架构在应用层以单纯的软件服务难以达到人机物融合。用户在现实生活中的需求通常和线下的物品和物理时空等要素有关，这时候各类计算资源对应的软件服务无法完全满足用户需求，往往需要进一步构建一种新的人机物交互方式，构建人机物融合的一体化应用。

人机物融合的研究是物联网架构应用层之上的人、机、物三元要素的深度融合。与“人”相关的研究主要集中在社会空间中的人机交互技术、众包技术，社会空间中人这一角色在人机物融合中既可以是场景活动的发起者也可以是社会协助者。与“机”相关的主要集中在智能机器人和信息服务的相关研究，智能机器人的研究主要应用人机交互和人工智能技术提供智能机器服务，结合机器学习，云端数据挖掘，分析用户行为习惯模式，设计实现智能服务[9]；信息服务的相关研究主要有服务组合和编排技术，包括传统的工作流技术、基于 Agent 技术的服务联动[10]、基于 SOA 架构的服务动态组合以及基于微服务架构的服务编排方式。“物”的研究主要是在智能设备和设备中间件平台方面的研究，智能设备的研究集中在智能传感器的研究，这也是当前国内外都竞相抢占制高点的前沿技术热点之一[11]；而设备中间件平台国内外成熟的平台包括 RestThing, EcoDiF, Carriots, SmartThings, HomeAssistant, 这些平台可以抽取 web 服务，具有向开发人员和用户隐藏硬件、网络协议、操作系统和异构型的特点，可以更加轻松地

管理资源并提高顶层应用的可行性[12]，特别是可以处理动态适应、具有上下文关系的相互影响。

随着当前信息技术应用构造方式从以技术为中心转变为以用户为中心的发展趋势，以人为中心的人机物应用构造成为了一个重要的研究方面，主要包括用户拖拽方式[13]、语音交互方式[14]和手势动作识别方式[15]等方法的研究内容。

## 1.2 研究目的和意义

人机物融合是物联网技术领域下的一个新的研究热点，同样具有提高舒适度和提供节省时间的应用来极大地影响人类生活的特点。但受限于当前技术水平的发展，日常生活中和线下的物品和物理时空等要素有关时，仅仅依靠感知设备所提供的软件服务无法满足需求，人机物融合可以解决三元之间日益凸显的异构问题，实现人机物融合的一体化应用。

针对当前的人机物智能空间应用下存在各类资源异构特性，联动模式固定单一且不支持自定义，满足需求场景有限等不足。本文的核心目标是提出一种针对流程类的日常生活场景，基于用户活动语义的简单可视化的场景活动流程个性化定制设计和人机物资源自动化和自适应调度的技术方法。该技术首先需要构建人机物资源仓库对人机物资源进行一体化描述，建立用户自定义流程设计中活动模型与人机物资源调度执行的关联，然后在用户流程定义工具基于拖拽方式组合带有活动语义的图元模型定义活动流程，借助于人机物资源仓库的知识关联可以在系统运行环境中自动化和自适应的调度最优的人机物资源完成用户流程活动。

本文的意义在于，支持用户通过利用该技术方法，利用流程定义工具以一种可视化拖拽简单语义活动模型组合流程活动的流程定义方式来满足即兴的自定义定制场景流程设计的需求，完全不用关心活动地完成，用户定义的整个场景活动将由系统自动化并且依据当前环境自适应地调度合适的人机物资源来完成，将用户和场景活动下的人机物资源隔离开了。该工具适用于智能空间下流程类的生活场景需求，可以大大提高用户的时间效率，增强了用户在智能空间下的生活服务体验，能够提供简单便捷和可个性化定制的生活服务。

## 1.3 主要工作和创新点

本文以人们实际生活中常见的具有流程特征的生活场景为研究对象，提出了一种面向用户的人机物融合应用的工作流设计与执行技术，基于该技术方法实现一个人机物融合应用工具。该工具系统包括人机物资源仓库、用户流程定义工具和运行时环境三个核心组件。人机物资源仓库对人机物资源进行了一体化描述，解决了三元异构特性，对人机物实例资源进行服务抽取和抽象向上抽象构建了以

Neo4j 图数据库存储的空间人机物实例资源关系模型，人机物实例服务模型和服务能力特征模型。基于该仓库，完成了用户流程定义工具和系统运行时环境的构建。用户进行个性化自定义的流程活动模型与人机物资源仓库关联，支持用户以可视化的拖拽方式组合编排用户流程活动，并支持用户模型映射和检验。运行时环境支持自动化执行用户流程活动，基于人机物资源仓库可关联相应人机物资源，并采用动态服务绑定策略自适应调度合适的人机物资源完成用户活动。

该工具支持用户以一种比较简单和易于理解的方式可视化地对流程活动进行个性化定义并支持流程定义错误提示，丢弃了直接关联人机物资源或者资源服务的缺乏环境应变的方式，采用了基于人机物资源仓库自动化和自适应调用最优的人机物资源实例服务调度相应的人机物资源完成用户设定场景流程活动的更加灵活可靠的技术方法。

综上所述，本文设计提出的技术方法以及实现工具的主要工作和创新点在于：

1.提出了一种人机物资源仓库构建方法。该方法对人机物实例资源进行服务抽取和服务抽象，构建了包含人机物实例服务模型、服务能力特征模型和空间人机物实例资源关系模型的仓库，建立了和用户场景流程定义活动模型以及人机物资源调度执行的知识关联关系。

2.提出了一种人机物融合应用的工作流设计方法。该方法涵盖用于描述用户活动流程的元模型，基于该元模型构建面向用户的流程设计工具，支持用户可视化拖拽简单语义活动模型组合流程活动来满足即兴地自定义流程活动的需求，并支持用户模型的有效性校验和映射，校验通过将映射为执行引擎可解析模型，否则向用户反馈错误信息支持用户流程重定义。

3.提出了一种人机物融合应用的流程活动自动化和自适应执行方法。该方法基于扩展的工作流执行引擎自动化支持用户流程活动的自动化执行，通过人机物资源仓库的关联关系可以直接调度人机物资源，并且基于资源可用性、空间距离关系和用户习惯三维衡量标准的人机物服务动态绑定策略支持实时环境下自适应调度最合适的人机物资源来完成用户活动。

4.设计并实现了人机物融合应用的工作流设计与执行工具，并在实验环境下进行了实际场景的使用，验证了该技术方法和工具的可行性和有效性。

## 1.4 篇章结构

本文围绕人机物融合应用的工作流设计与执行的研究与实现，基于以上主要工作点，将分以下六个章节来具体阐述具体内容。

第一章，绪论部分。本章主要介绍本文的研究背景、目的和研究意义，从人机物三元要素分别介绍了相关研究背景，然后介绍了人机物三元融合的研究方法

的现状，最后介绍了本文的主要工作内容和创新点。

第二章 相关工作和技术背景的研究。本章主要介绍了本文提出的技术方法所涉及到的相关工作和技术背景，其中主要对人机物融合技术、工作流技术和面向用户的服务组合技术三方面的技术和本文相关知识进行介绍。

第三章 人机物融合场景分析与方法描述。本章首先对人机物融合动机场景案例进行了分析，然后介绍了总体的研究架构以及对核心的方法设计进行了描述。

第四章 人机物应用工具的设计与实现。本章介绍了系统工具的设计架构，并从人机物资源仓库构建、用户流程设计工具构建和与运行时环境构建三个方面详细介绍了其核心组件的方法和算法的具体实现。

第五章 场景实例实验与分析。本章对本系统进行了面向用户的工具使用的性能评估，并基于实际场景案例对系统各模块进行了实验结果的验证与分析，并总结了本文存在的方法局限性

第六章 总结与展望。主要对本文研究内容做了总体的总结，结合方法局限性对未来工作进行展望，对下一步工作做出了规划。

## 第二章 相关工作及技术背景

人机物融合课题是多技术融合的研究课题,当前的人机物融合的研究切入点、应用构造方式也不尽相同。本章节先大致介绍当前人机物融合技术的应用构造的相关工作,然后在此基础上提出了人机物融合设计与执行的应用构造方法,并在本章其他章节重点列举和描述了所涉及的相关技术和工作。

### 2.1 人机物融合技术

人机物融合技术是在物联网快速发展进入快车道,在与工业界结合落地过程催生的产物。由于人们在日常生活中,需要涉及的人、机、物分别被锁定在社会空间、信息空间和物理空间,三者间内部和三者间存在异构性,无法统一融合应用来满足人们活动需求。在这种状况下,人机物融合技术应用而生。

人机物融合是以实现无缝对接、协同计算为目标,人与机、机与物、物与物以及人与人的三元资源融合的综合应用技术看方案。可以展开想象,在人机物融合技术的推动下,将形成以人为中心,人类社会、虚拟空间和机器物理空间智能交互、数字融合的人机物三元融合新社会形态。

人机物融合技术,是在应用层之上的通过一种统一的人机物服务组合和编排方式实现的三元融合。具体概念关系如图 2-1 所示:

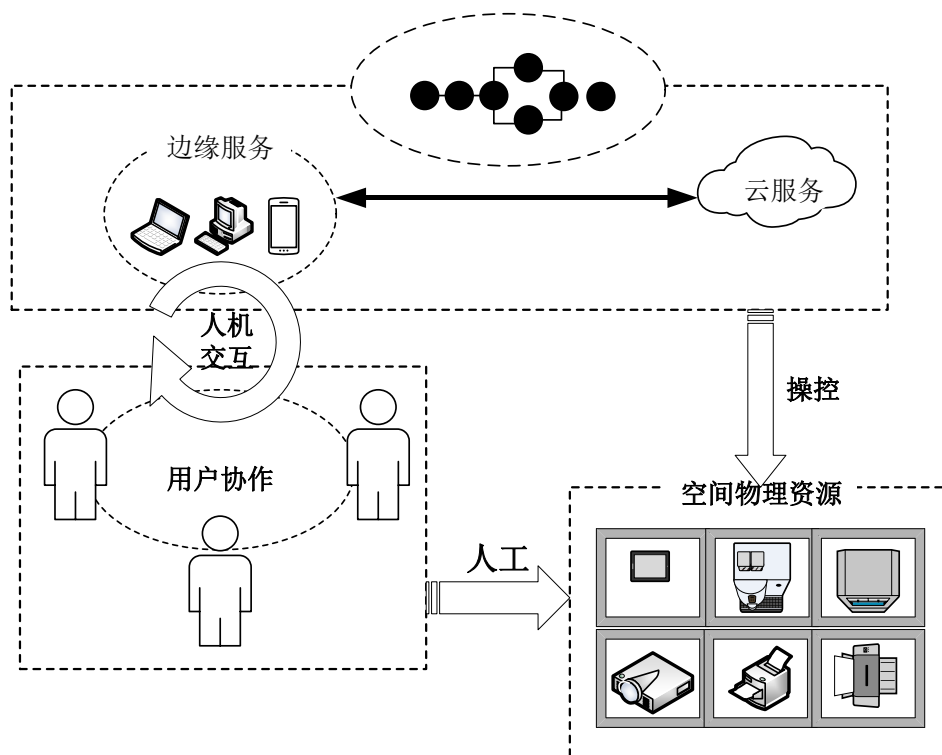


图 2-1 人机物三元融合示意图

在本文中，“人”作为人机物融合的中心，既可以是活动需求的发起者，如人机交互中人作为场景活动发起者，也可以是活动协作者，比如众包协作服务。

“机”则是信息服务和信息服务超集组成的概念，在本文中就是整个工具服务或者单个服务对应的机器的具体体现，比如说咖啡机，投影仪控制器等。而“物”的概念就比较清晰，对应的就是物体空间中的各种传感器或者简单设备。简而言之，人机物融合的一个最直接体现就是“人”通过“机”来操控、使用物资源。

人机物融合技术是一项可跨学科的多技术融合的综合应用。就研究内容而言，当前的人机物融合应用研究大多面向应用服务，可分为以下两大类：

### 1.以场景为中心的事件驱动应用模式

当前的研究和场景落地现状多以这种应用形式为主，是一种比较成熟和稳定的可商业化应用模式。这种融合方式是一种基于已有中间件平台或者一种统一的交互方式来实现的基于特定场景下的可靠应用。最近苹果打造的智能空间，统一了苹果产品系列设备，设定好用户的一些具体场景，比如用户进入房间内通过和 Siri 交互，自动打开苹果相关产品和软件服务。另外，基于设备统一的中间件平台如 HomeAssistant，内部打破设备间的异构对外则统一以 Restful 服务提供能力，平台本身就可以通过编写配置文件的形式提供基于 ECA[16]的服务能力。

### 2.以用户意图为导向的推荐应用模式

这一类人机物应用形式是分析学习或者直接与用户交互来以用户意图为导向的应用模式，具有面向用户、强流程性服务的特征。目前这种模式研究更加满足人们迫切的智能化融合需求，但落地产品相对减少。一种研究手段是结合人工智能相关技术分析学习用户行为模式构建决策系统来预测和推荐服务，缺点在于系统庞大，需要严格的计算能力，当前实际可用性不高。如 Zaineb Liouane 等人[17]基于机器学习模型训练学习用户生活模式予以服务推荐，Thomas Reichherzer 等人[18]基于多 Agent 技术通过传感器数据采集分析推理用户行为。另一种是通过面向用户的可视化人机交互方式来提供自动化的、推荐性的服务，如 [19-21] 在基于 web 内容服务和应用的面向最终用户的混搭融合和 Holm Smidt[22]等人基于嵌入式元件模型的用户可视化编程应用构造。

本文采用的是第二种应用构造方式，用户通过可视化编排定义的流程活动在系统运行环境时中基于构建好的人机物资源仓库直接关联人机物实例服务从而转化为以软件服务为基础单元的服务组合求解问题来实现人机物融合应用。人机物服务中机的服务主要是指第三方信息服务，也可以是自定义服务及其服务集合，物的服务是指设备提供的功能服务，人的服务可以是人机交互，也可以是协作服务。人机物融合中，HomeAssistant 提供管理和控制代表“物”的物理设备的能力，众包技术用来支持人机物融合应用中代表“人”的社会协作能力。人机物三元融合技术来连接人和物，实现人、机和物的三元融合的应用构造。

### 2.1.1 HomeAssistant 平台

HomeAssistant 也可以称为“家助物联”，是一个在 2016 年开源的一款智能家居管理平台，致力于打造智能信息空间的平台架构。基于 HomeAssistant，我们可以轻松的连接和管理空间内的设备，并且手动地控制或者基于自己需求地通过配置来自动化地联动它们，通过对这些设备地控制和统一的管理可以构建随心所欲地信息空间。如下图 2-2 所示，Hass 以组件为单元的方式来控制和管理物理设备，对设备的发现和控制都对应的是一种配置文件的编写和生成，特别是对于设备的联动需要编写符合 Hass 语法的基于 ECA 规则的配置语言。而在服务管理模块和状态管理模块可以对外提供原生的所有服务信息。

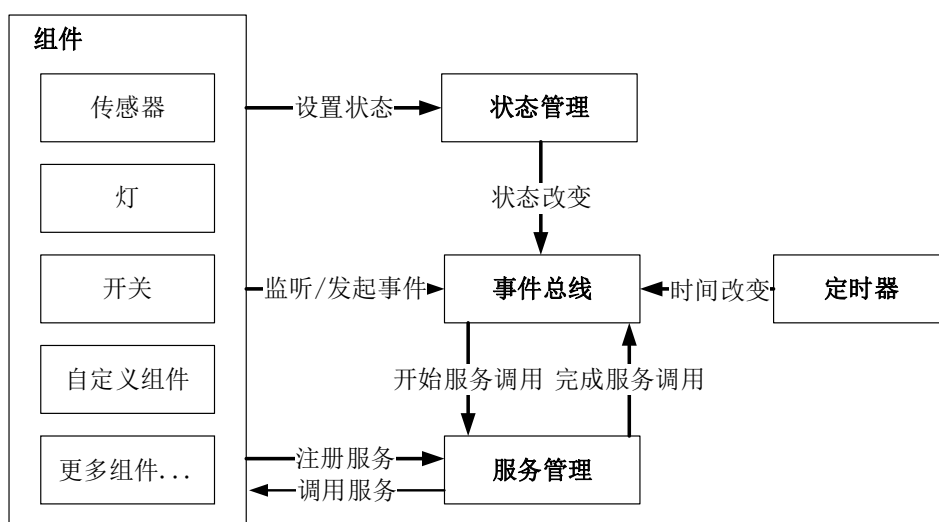


图 2-2 HomeAssistant 设备控制原理图

### 2.1.2 众包技术

众包[23]是以社会大众协作为基础的来替代机器服务完成机器难以单独完成的任务模式，它是在互联网快速发展但受限于当前技术水平的限制而无法完全由机器来代替人类手动的环境诞生的解决手段。当前，众包技术覆盖了人机交互、人工智能和信息检索等众多研究领域，成为了一个崭新的研究热点。

众包技术利用社会资源，发动社会大众来完成计算机无法完成的部分活动任务，其基本特征包括：众包任务一般来说是计算机或者机器无法直接独立完成的；面向的是一定环境下的大众，具有一定的公开特性；参与者可以多人协作或者单独的承担并完成任务；可能包括多阶段的解决方案。依据以上特征，我们可以将众包任务归类为两类：一种是多人协作的众包任务，另一种是竞争式的单人众包任务，完成众包任务就可以获得一定的奖励或者激励。本文人机物融合技术采用竞争式的众包任务将资源送达至需求人的位置，体现了人机物中人的协作和参与。

## 2.2 workflow 技术

本文的研究对象是具有流程特性的生活服务场景，结合采用了 workflow 技术 [24] 思想，工作内容涵盖了其相关技术。workflow 技术最早是应用于企业管理业务流程自动化的过程自动化技术，其最终目标是基于定义的规则，把任务、信息按照描述规则在参与者之间传递以达到需求目的。下面本文从 workflow 模型的角度来简单描述相关技术和工作，其 workflow 参考模型描述如下图 2-3 所示，它由 workflow 管理联盟引入和提出。该模型规范和定义了 workflow 由图中六大模块和统一的应用编程接口。本文关注的重点以下核心部分：流程定义工具、workflow 格式转化、workflow 执行引擎、执行应用调用和客户端交互几个部分。workflow 完成一整套完整的流程步骤分为三个阶段：流程定义、流程部署、流程执行。流程定义阶段的流程定义工具是和用户交互的可视化界面，输出是一种中间描述集，workflow 格式转化处理中间结果集转化为可执行文件，过渡到流程部署和执行阶段，交由 workflow 执行引擎执行，并调用相关应用或者服务顺序执行，最后返回客户端结果。总而言之，成熟有效的 workflow 产品必须包含流程设计和流程引擎两大关键组件，它们是本文的 workflow 方面的研究重点部分。流程设计负责流程定义阶段工作，流程引擎负责流程部署和执行的流程流转。

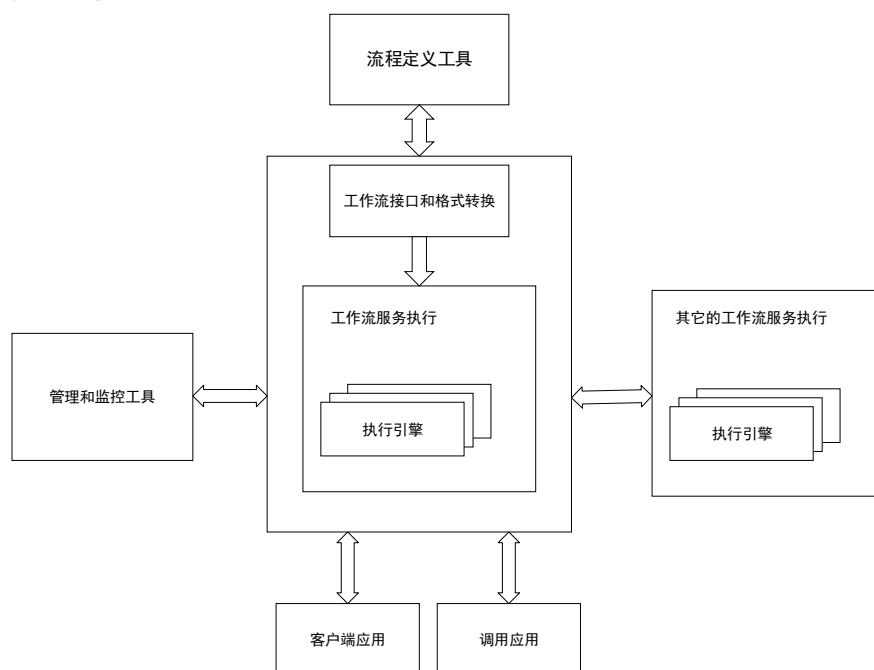


图 2-3 workflow 参考模型

目前 workflow 技术相对比较成熟，本文的技术方案所涉及的工作流模型组件主要是流程设计和流程引擎两大核心部分。一般来说，workflow 具有以下四个特性：（1）具有 SOA 架构特性；（2）支持分布式集群；（3）具备一定的可扩展能力；（4）具备相应的业务嵌入功能和开放特性。workflow 通过流程定义工具和 workflow



引擎可以实现流程服务的执行等一系列的部署、启动到实例化执行等操作。其工作流服务流程如下图 2-4 所示。

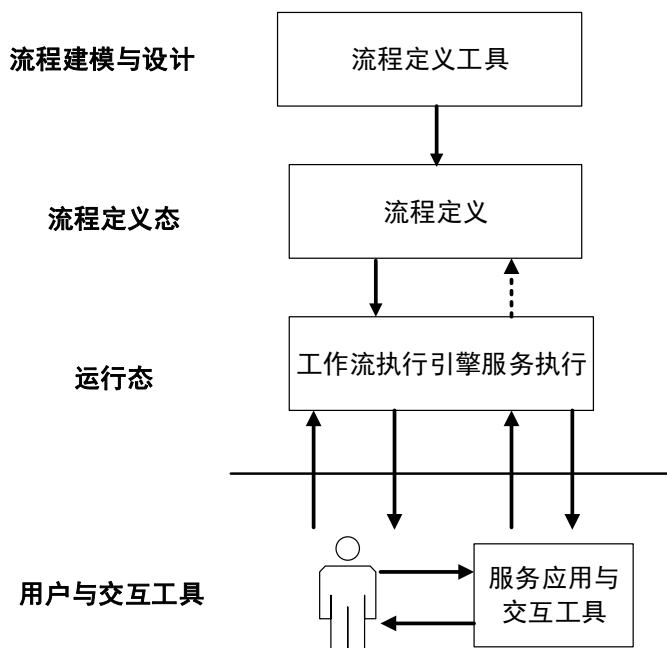


图 2-4 工作流服务流程

## 2.2.1 工作流执行引擎

执行引擎是工作流系统实现的核心组件，也是工作流驱动流程过程的决策运行的核心解决方案。它主要负责部署以及解析流程定义的描述文件为工作流执行模型，工作流启动运行阶段创建流程运行实例，并在执行阶段按照设计规则调度 and 流转。工作流就是将业务过程迁移到计算机应用下自动化的体现，工作流引擎的出现则是在对用户不可见的下层进行功能封装而提供的一种更好、更方便、快捷在实际应用中使用工作流的解决方案。它的作用在于解放了开发者的部署、执行编码，提高了软件质量、加速应用的开发，可以直接地解决用户的很多业务问题，是业务自动化的核心技术。

本文的研究对象是人们日常生活中流程类生活服务，所以需要选择最适用于人机物融合应用的工作流引擎，在此基础上进行扩展实现。目前国内外市面上存在不同类型的常见地工作流引擎：

### (1) OSWorkflow 执行引擎

OSWorkflow 是一个轻量级的工作量框架，其突出的特点在于灵活性，支持的设计标签元素较多，可以嵌入到企业的应用程序中，和与 Spring 等框架集成，提供了很多持久化 API，包括 EJB, Hibernate, JDBC。但是提供了许多优势的同时，它严格面向开发者的缺点也很明显。OSWorkflow 支持手工编码 XML 文件定义规则，很容易被实施人员代码侵入和破坏，不是快速“即插即用”的解决方案。

### (2) jBPM 执行引擎

jBPM 是一个可扩展的、灵活并且开源的工作量引擎，可以嵌入到 java 应用或者运行在独立的服务器上。从 jBPM 到 jBPM5 不断增强其流程可视化能力和管理能力。遗憾的是 jBPM 的版本迭代使得功能逐渐增加的同时系统变得十分冗余，面向的是工业界办公业务，协作功能不够完善。

### (3) Activiti 执行引擎

Activiti[25]是基于 jBPM 在 BPMN2.0 上做了整合的自动化业务 workflow 引擎，它基本应用在业务流程管理方面，工作流方面以及服务协作方面等诸多领域，更加灵活、易扩展。相比于前两种，activiti 的最大特性在于它的协作能力上和用户体验上，拥有简单的可视化建模器和监控管理组件，提供了基于 Restful API，增强了分布式服务的调用，消除了表单的抽象，底层封装全局变量、流程变量、任务变量，更加友好易用。

相比之下，Activiti 工作流引擎更加接近本文的最终设计意图，本文工具的实现正式基于 Activiti5。Activiti 完美兼容了 BPMN2.0 规范，将用户流程建模和流程引擎部署、执行解耦，使得用户关注流程建模而底层开发人员关注流程引擎的部署执行部分，可以提高工作效率。

Activiti 工作流引擎 ProcessEngine 包括了七大组件，开发人员在部署执行编码通过调用各个服务组件接口，完成工作流的运转。该七大组件接口分别是 RuntimeService（引擎运行时服务接口）、ManageService（引擎异步执行的查询和控制接口）、RepositoryService（引擎实例仓库接口）、TaskService（引擎任务管理接口）、HistoryService（流程实例的任务、数据的查询接口）、IdentityService（用户和用户组查询接口）和 FormService（表单数据的读取管理接口）。

## 2.2.2 流程定义工具

流程定义工具具有可视化、图形编程的特征，是一种面向开发者或者用户的交互工具。它根据面向对象的不同输出结果集也不相同，面向开发者的输出结果集可以直接是工作流执行引擎的可解析、执行文件，面向用户则输出的是的一种中间结果集，需要通过映射转化为工作流执行引擎的可解析文件。

一般来说，工作流引擎会配套相应的流程设计工具，Activiti5 提供了协作工具，其核心组件如图 2-5 所示。Activiti5 提供了两个流程定义工具：Activiti Modeler 和 Activiti Designer[26]。Activiti Modeler 是 Activiti 针对表单设计的流程建模器，对那些支持 BPMN2.0 规范的表单提供图形化设计，也可以进一步将其整合到 Activiti Explorer 支持 Web 浏览器的图形化建模。而 Activiti Designer 是以 Eclipse 或 IntelliJ 插件的形式支持 BPMN2.0、支持扩展的流程设计工具。

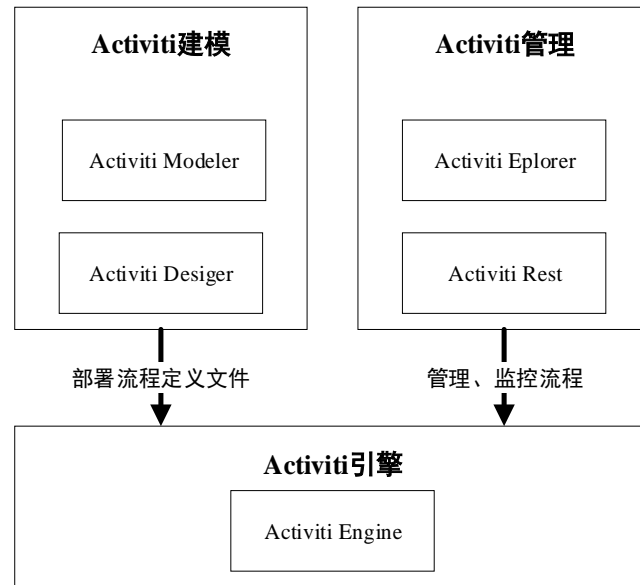


图 2-5 Activit 核心组件

### 2.2.3 BPMN2.0 建模标准

BPMN 是一种通过准确的语义来描述业务操作的业务流程模型符号建模标准[27]，以 XML 格式文件为载体，以标准图形注解可视化业务，旨在向流程参与的用户提供一种易于理解、交流的统一建模符号，打破面向用户的业务流程设计和面向开发者的业务流程实现二者间的技术壁垒。BPMN2.0 是在 BPMN 标准的基础上提出的，解决了流程过程间的流转交互的标准定义，是一种自上而下的组织建模语言。BPMN1.0 存在一些缺陷，与 BPMN 相同的作为一种统一建模标准的还有 JPD L、XPDL 和 BP EL，也被各大供应商用来表示 BPMN 流程模型，而 BPMN2.0 的出现解决了图交换过程中的交换格式问题，一跃成为了最主流的建模标准。

BPMN 的提出目标是为符号、元模型和交换制定单一的规范，而且 BPMN2.0 已经逐步扩展实现针对流程模型层面的编制和编排。本文提出的设计方案是基于 BPMN2.0 这一标准建模规范语言来实现最终的人机物组合和编排的。BPMN2.0 在整个方案从用户到底层实现的转化之间起到了从用户抽象到开发实现的关键作用。下面简单介绍对其中的主要模型。

BPMN 主要由流对象、连接对象、数据、泳道以及工件这五个主要元素组成。流对象（Flow Element）是业务建模的核心元素，包含了活动、事件和网关三大类元素。活动（Activity）是流程执行的核心任务，包括了 UserTask、ServiceTask、MailTask、ManualTask 等不同具体形态的任务划分。事件（Event）是用来描述流程过程发生的事情，可以分为捕获和触发事件，也可以分为开始事件（StartEvent）、

结束事件(EndEvent)和中间事件(IntermediateEvent)。网关是流程的控制结构，可以分为排他网关、包含网关和并行网关。连接对象用于将流对象连接在一起，控制流程活动之间的顺序和整体流程的过程流转。连接对象可以分为三大类型：顺序流（Sequence Flow）、消息流（Message Flow）和关联（Association）。数据可以说是一种半抽象的元素，是数据对象、数据输入、数据输出和数据存储的总称。例如，它可以 UserTask 的分配对象、流程变量或者属性变量等。泳道的功能主要是用来进行模型分组，可以池（Pools）以及道（Lanes）组成。工件可以用来描述流程或者模型元素的信息，提供信息备注的功能。

## 2.3 面向用户的服务组合技术

面向用户的组合服务技术的关键在于面向用户使用工具的简单化和服务组合的灵活性和有效性。人机物融合场景中三元要素可以以 Restful 服务或者 Web 服务封装其基本能力和信息，所以追根到底，人机物的融合在于如何把具有异构、存在“高技术壁垒”的三元世界元素转化为信息服务的融合。

本文基于人机物服务提出的人机物融合应用的工作流设计与执行技术目标是为用户打造一个个性化服务式的人机物交互环境。服务组合属于面向服务计算的领域范畴，是基于 Web 服务以及软件复用提出的一种区别于传统服务应用的构造方式。它是以基础服务作为基本单元进行服务的组合来实现具体应用系统的松耦合开发方式，相较于传统应用开发而言，开发效率和可用性大大提高。

然而，值得注意的是，从以技术为中心慢慢向以用户为中心倾斜的应用构造方式已经成为近些年来信息技术的一种重大的思想革命。而这也恰恰对应了人机物融合研究热点从以场景为中心的构造方式过渡到以用户为导向的应用构造方式，更多考虑的是如何让用户按照自身的意图和需求来组织和配置可达资源。

面向用户的服务组合应用主要包含以下几个特征：

### （1）服务语义抽象[28]

首先面向用户是以用户为导向的一系列解决方案，而其中的基础服务单元是具有技术特性的，所以需要服务进行一种抽象，变成用户易于理解和使用的基础单元。它是一种逐级抽象，向下屏蔽底层计算资源的技术人员和用户之间的中间步骤，实现的是角色视角表现层的一种转换。

### （2）执行与构造同时进行

在进行开发和部署过程中，表现层上层进行了语义抽象，下层的的服务支撑可能就是一对多的模型关系或者服务组合本身的前置服务和下游服务之间就是一种不确定关系。所以，应用在执行期间会存在一边执行、一边构造的实现方式。

### （3）编程模型的构造

由于用户和技术人员之间存在的“技术壁垒”，系统应用需要构建编程模型，它是对整个系统应用的应用模式和具体内容的解决方案上的大致思路和特定抽象。该编程模型除了要对系统内容建模还要借助或者创造一种统一建模标志语言，给用户在不同情境下的使用需求的解决方法。

在过去几十年，针对服务组合和集成的研究一直在不断进行和优化，下面介绍几种类型的服务组合技术。本文基于以下的一些现有的服务组合技术进行分析对比优缺点，结合人机物特性提出了一种优化改造的面向用户的人机物服务组合技术方案，具体实现将在本文第四章详细介绍。

### 2.3.1 基于工作流的静态服务组合

当前基于工作流的服务组合通常是基于工作流部署的工作流流程定义和调用相应 Web 服务。Web 服务可以解决对应人机物资源的异构性方面的互操作问题，结合工作流的 Web 服务组合[29]可以用来实现服务执行流程中的静态绑定。Web 服务具有松散性特点，是基于 SOA 架构可以将不同系统平台的组件发布为服务，如下图 2-6 所示。

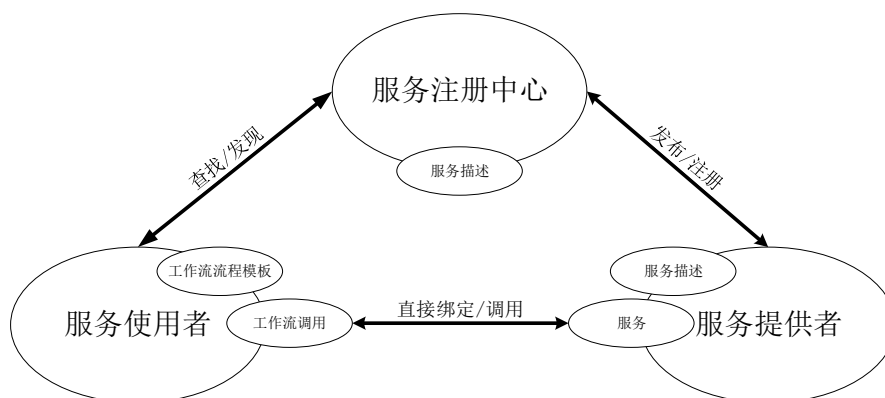


图 2-6 基于 SOA 和工作流的服务组合结构模型

SOA 架构将系统分为三种不同的角色：服务提供者、服务注册中心和服务使用者[30]，其间的工作原理是服务提供者将服务和相应的服务接口、地址等信息服务描述信息发布到服务注册中心，服务注册中心支持服务使用者的服务发现，服务使用者指的是用户部署的流程应用模板，在服务组合工作流程执行期间静态绑定、调用服务。

通常来说，服务组合流程可以直接通过 BPMN、BPEL 等建模语言以 xml 形式描述了执行过程中需要调用的服务或者将这种映射关系写入在开发实现代码中，这样的一种实现方式是一种静态的服务绑定策略。严格来说，该方式面向的是具备一定专业技术知识的用户，而非普通用户。

这种静态服务绑定的方式下将对应用组合服务对环境的应对适应能力降低

不仅仅在具体系统应用方面存在局限性,更存在着后期维护成本高、灵活性极差、服务质量得不到保证等特点。特别是服务质量的保证性上,当网络环境或者服务的可用性产生变化时造成的后果时灾难性的。

### 2.3.2 基于 SOA 架构的动态服务组合

基于建模语言和 SOA 架构的动态组合服务是对前一种静态组合的优化和改进。该方案中解除了服务使用者和服务提供者两者之间的静态关系,进而改用由服务代理来对服务进行动态绑定,组合结构如下图 2-7 所示。

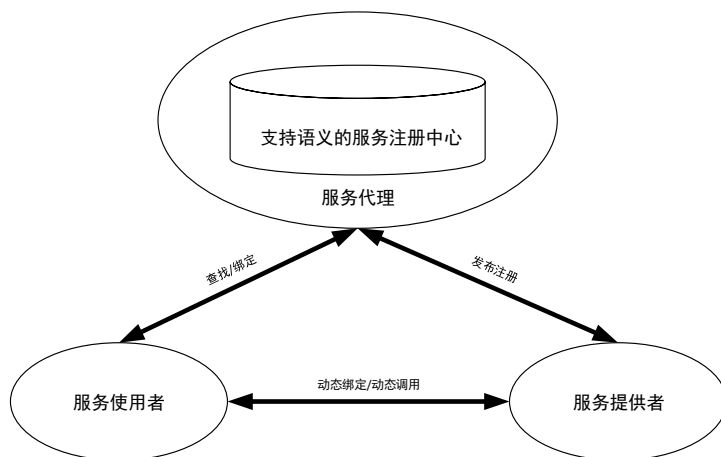


图 2-7 基于 SOA 动态绑定的组织架构

区别在于,该组织架构中引入了支持语义的服务代理作为服务注册中心,进行了服务的语义描述[31],服务使用者从服务代理中查询获取的是一个经过推理的最优服务,当应用环境发生变化,服务代理需要重新评估推理得到一个最优服务返回给服务使用者,这样就实现了服务组合的动态绑定。可以看出,在该方案下,服务的动态绑定是一个重要模块,传统的动态服务的绑定采用的是基于 Qos 属性[32] (比如服务完成的时间和完成效率)的服务评估指标作为服务评估的重要依据。其动态绑定算法流程描述大致如下图 2-8 所示下。

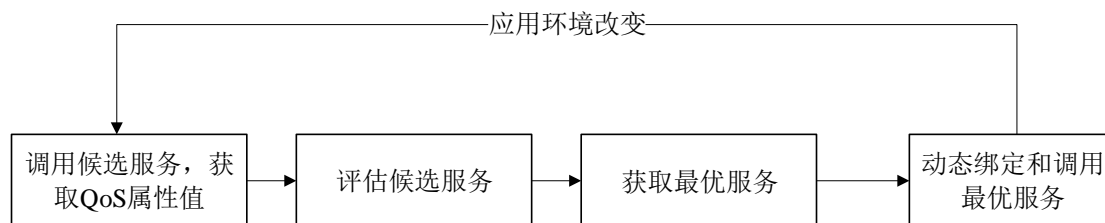


图 2-8 服务动态绑定算法

该部分的动态推理过程完全在支持语义的服务注册中心组件中完成的,是一种基于多属性的评估标准来对服务进行服务质量评估的。该算法的输入是一个查找请求,推理算法基于服务注册中心里的带有语义关系的服务集合,初步筛选得到候选服务集,在使用评估标准获取最优的服务。这是一个不断迭代循环的过程,

所以能够消除环境变化给系统应用带来的服务不可用问题,体现了较高的灵活性。但它在具体技术实现上各有不同,有研究者在服务动态绑定上采用基于 OWL-S 的语义描述,对 OWL-S[33]进行语义解析进一步完成推理,也有研究者用的是基于 Neo4j 图数据库进行资源关系建模完成语义描述的方式完成服务的动态绑定。动态服务组合方案目前针对的主要是 web 内容和服务,基于不同的服务动态绑定策略来实现动态地服务组合关联。

### 2.3.3 基于 MashUp 的服务混搭组合

基于 MashUp 的服务混搭组合[34]是一种新式的服务组合构造方式。该技术方案主要特点在于利用可重用的服务资源来创建系统应用程序,从某种意义上严格来说,这种方式才是真正的面向最终用户,可以实现非技术用户的可编程能力。MashUp 服务组合方式最大的特点在于它侧重于系统应用表现层的组合和集成,拥有一种可以和用户直接同步交互的服务界面或者交互方式,交互手段比较直接,对用户而言更加易于理解和操作。

该技术方案的对服务的抽象程度更高,从对服务的语义抽象上升为对用户而言更加易于理解的图形图像表示、语音接入以及姿势行为动作等最为直观的抽象表示。在该方面的研究目前也是比较火热,比如某些研究者基于 Web 页面方面做的 MashUp 的服务组合研究,是对 Web 页面进行了抽象以图像界面的形式直接向用户展现,方便用户的服务组合编排。某些研究者在移动设备上基于 app 的录制回放技术做的 MashUp 服务混搭,是以语音的形式作为与用户的直接交互手段,服务的组合在于预先完成的录制和回放技术。

但该方案相对不够成熟,技术理论储备有所欠缺,目前的主要应用还都是一些基于场景的 Ad-hoc 形式的实现和研究,针对面向用户的“即兴”定制还需要更多的技术支持。所以,综合比较来看,本文的技术方案主要是综合了前两类服务组合技术,设计和实现了人机物服务组合的技术方案。

## 2.4 本章小结

本章内容主要介绍了针对本文基于流程类生活服务的研究对象提出的人机物融合应用的工作流设计与执行方法所使用到的相关技术和内部原理,分别从人机物融合技术、工作流技术和组合服务技术进行了介绍。首先介绍了人机物融合的相关概念,它是一种比较新式的概念,涉及到人机物的三元融合,然后对人机物各类服务进行了定义和描述,并介绍了涉及到的 Hass 平台和众包技术。然后描述了工作流技术的相关组件和重要模块,接着阐述和比较了目前服务组合的几种常用技术并提出本文人机物融合应用的工作流设计与执行的技术路线。

## 第三章 人机物场景案例分析与方法描述

人机物融合是本文提出的人机物融合应用的工作流设计与执行技术的关键理论基础。本章节将首先对人机物融合动机场景案例进行分析,利用经典的人机物融合案例对本文人机物融合应用的工作流设计与执行的关键技术研究思路提出了需求挑战,并比较了目前技术存在的不足和挑战。接下来,本章将介绍本文的总体研究框架,并依据框架和方法流程的设计思路展开具体的方法设计描述。

### 3.1 人机物融合动机场景案例分析

智能空间是物联网发展至今的一个非常热门的研究热点,也是能够体现人机物融合的经典案例。在智能空间中存在着三类重要要素:人、机和物。在这三元要素中“人”可以作为空间活动的发起者,也可以是空间活动的协作者,我们认为活动发起者应该覆盖非技术人员的普通用户,活动参与者可以是具备专业技术能力或者普通的人;“机”则是一些第三方信息服务或者可交互机器;“物”是智能空间下的所有的智能设备的集合。在智能空间下,三者的最直接的关联方式是“人”通过“机”来操控或者联动“物”。

现实生活中存在一个真实场景,在一个机构单位下的办公空间,某领导带领员工开始晨会,他希望大家能够一扫早晨上班疲惫、精神涣散的精神状态,为参会者提供一些便利的生活服务。一般情况下,在该场景中存在以下一些活动步骤:打开投影仪,打开空气净化器,这时候再为参会者提供一杯浓香的咖啡就更好了,那么还需要去开启咖啡机,等待咖啡打磨好后将咖啡取回来。

结合实际生活的经历,在该场景下存在着一系列的问题使得本意是提供便利服务的过程不够流畅和舒适。假如这些活动需要发起人自己去或者安排人去完成,甚至完成者不熟悉设备的使用方式,这种方式明显不够便捷和高效。

而现有智能空间设计模式已经不足以满足人们的日趋提高的日常需求,根本上的缺点在于它还是停留在以技术为中心,是一种增量式的被动处理模式。服务提供商或者平台集成各类资源,一般是不完整的三元资源(二元、甚至一元资源),需求从市场调研分析得来,针对的是一种大众的需求场景,其缺点是 ad-hoc 式开发应用模式单一僵化、极度缺乏灵活性,应用范围有限。

针对该应用场景实例,本文希望能设计出一种技术方法支持用户以一种简单的自定义设计方式能够自动打开投影,完成后打开空气净化器,自动启动咖啡机冲咖啡,自动发送众包任务让其他空闲的人自愿性可以将冲好的咖啡接送到会议室。即支持用户个性化设计流程活动且支持自动化完成用户定义的流程活动。

针对上述类别的生活场景,本文提出了人机物融合应用的工作流设计与执行



的设计方法，目标是实现人们日常生活中存在的较为普遍的流程类生活场景下的智能化信息空间的打造。它的最终展现形态是以用户的视角，通过相对简单和易于理解的方式可“即兴式”的编排用户流程活动，然后能够自动化和自适应调度智能信息空间下的人机物资源来完成场景活动。

### 3.1.1 人机物资源一体化整合需求

本文首要解决的问题就是人机物资源的一体化描述和管理，这也是人机物资源仓库阶段的核心需求。人机物资源进行一体化描述是将物理设备、人和信息服务进行融合的关键步骤，为人机物融合应用提供知识背景支持。我们上面提到，人、机、物三元要素存在异构特性，数据信息被锁定在各自空间中，如何对人机物资源一体化进行描述和管理是人机物融合应用首先需要面对的问题。

本文执行阶段核心处理对象的基本单位是人机物资源服务，它的优势在于 Web 服务是以遵循 XML 标准描述规范的，方便建立人机物资源统一的描述和访问方式。因此，人机物资源整合就可以转化为服务层面的统一，主要包含了资源服务抽取和服务抽象和统一整合管理等内容。

资源服务抽取分别对应了人、机和物三类资源的服务抽取构建。物一般指的是物理设备，对物理设备进行服务抽取主要是基于物理设备接入的管理平台利用统一通信协议对设备的状态、提供的功能进行服务抽取和构建；机一般指的是现有的信息服务或实体，只需进行服务的封装；人在本文中主要代表了场景活动发起人和协作者，抽取众包服务代表了人的参与协作。

另外，人机物资源一体化描述需要进行资源服务语义抽象过程，该过程用来将人机物服务进一步语义抽象，建立人机物空间环境下的语义知识，向下直接关联人机物资源，向上能够关联用户流程活动，从而构建用户流程活动到人机物服务以及人机物资源的关联映射关系。

### 3.1.2 面向用户的场景活动编排需求

用户并不是直接关联人机物服务，用户关联的是流程活动组合和编排。用户定义的流程活动映射转化为服务组合编排是发生在人机物应用执行态。早期的人机物融合是预先设定好的一种或几种适用于大众通用的 Ad-hoc 式的应用模式，后来为了应对更加多元化的用户需求开始出现允许可配置和添加应用模式的融合平台，但面向的是技术人员而不是普通用户。本文我们提出的解决方法覆盖面向非技术人员的普通用户，因此需要为用户设计一种既能“即兴式”地定制人性化流程服务又能让用户易于理解和操作的交互方式。

鉴于此，我们要关注的点在于两个方面：一、场景流程活动的定义方式；二、

面向用户的简单、有效的交互方式。首先我们需要确定流程活动的定义方式，在此基础上再设计一种面向用户的交互方式来完成流程活动的编排。

### 一、场景流程活动的定义方式

针对于流程类生活场景下的用户需求，我们需要以一种计算机机器语言或者符号甚至是计算机编程思想来描述和解释，来转化为流程的定义和构造目标。在定义和构造方式可以是基于消息中间件的订阅发布模式的状态转移也可以是简单的基于规则的流程活动对应的人机物服务调用。我们采用的是和用户需求比较契合的工作流流程定义方式，在该方式下需要基于一定的语义知识—人机物资源仓库，将人机物资源服务和 workflows 活动模型节点建立映射关系。固然基于 workflows 构造方式来描述场景流程比较形象，workflows 基于标准标注语言的强规范性可以省去一些范式的定义，但是其描述具体性也限制了流程和服务的可变性和可替换性，这一点需要我们对执行引擎做一些扩展处理工作。

### 二、面向用户的交互方式

workflows 定义场景流程的应用构造方式，严格来说，它面向的是专业领域的技术用户，而非普通用户，所以还要设计用户可用的交互方式。这种交互方式可以是语音的交互，手势动作的交互，图形示意的交互等带有语义信息的交互方式。

因此，本文需要为用户提供一个交互工具，后面我们也称为用户流程定义工具，提供给用户通过拖拉拽的形式组合能够描述流程活动的带有语义的活动模型来完成流程的定义。用户流程定义的过程也是用户建模过程，是对当前场景需求下的求解过程，只不过由服务提供商的求解过程的使用权交由用户来求解，并将这个过程抽象为用户可简单使用的过程。

## 3.1.3 流程活动自动化执行需求

打造智能信息空间自动化是面向用户的基本要求之一，但对于面向用户的交互手段也应该考虑到用户的技术“小白”属性和个人因素，存在着操作失误或者操作不当甚至恶意操作的可能性，因此在活动描述的人机物服务自动化执行之前需要对用户定义的流程进行有效性检查的应用执行前的预处理。用户定义的流程经过映射和关联对于计算机来说就是编排好的组合服务，由于计算机能够处理的就是机械的根据组合服务描述来做解释处理，因此我们需要在用户定义流程后对用户流程模型进行有效性校验，并给出一定的提示修正信息。这既可以对用户起到辅助和提示的功能，也提高了系统应用的安全性和可用性。

前面我们提到，用户在定义流程的过程中不需要关注服务本身，只需要依据流程活动步骤的语义组合的带有语义抽象描述的可拖拽活动模型即可。因此系统需要提供活动模型和人机物服务的映射和绑定的功能，而且为了提高系统可用性

和灵活性,服务动态绑定是其中关键一环。具体来说,某一功能的服务对应了多个设备服务,可能是相同的多个设备,也可能是不同的设备,比如说提供饮用水功能对应了多个饮水机和饮水服务,这时候用户的视角里只有饮水的相关功能,具体服务和功能语义的映射则需要交由系统来设计和处理,选择哪个饮水机的饮水相关服务完全对应用户来说是透明的,也是不需要关注的。总而言之,用户自定义的流程自动化执行过程中既要组合服务自动化能够调度人机物资源,也需要在执行过程中依据人机物空间现实环境的变化体现出服务的自适应性,具体表现在服务的动态绑定,调度唯一适应的人机物资源上。

### 3.1.4 现有技术的挑战与不足

通过上述对人机物融合应用动机案例的分析和对方案设计关键技术的设计思路过程分析,我们发现现有的技术手段不足以满足人机物融合应用的工作流设计与执行技术的设计目标。

传统的 Web 服务组合和编排的应用构造方式的基本单元是 online 的服务计算资源,集中体现在信息空间内的信息服务。而人机物融合应用的基本组成单元是人、机和物三元要素的实例资源,既包含了 online 的联动过程,也有 offline 的由物及人的不同要素间的联动。其次,传统的 Web 服务组合和编排的应用内部描述的是服务功能的语义信息,而人机物三元要素间存在着异构,人机物融合应用需要描述丰富的三维空间的语义信息,以 Web 服务应用描述人机物融合必然造成三元空间语义信息的缺失,无法满足智能空间人机物融合场景的部分需求。最重要的是传统的 Web 服务组合和编排不支持直接面向普通用户。

基于工作流执行引擎的 Web 服务组合和编排方式是传统的应用服务构造方式,它是严格依赖计算机的解释过程,也就是,服务组合的描述必须严格依照某种描述规则才能被计算机执行引擎解释执行,这种严格的约束造成系统服务的静态绑定,环境适应能力极差,难以满足多变的现实空间环境下的人机物融合。另外,传统的执行引擎提供的用户建模工具面向的是业务人员,需要具备一定的专业领域能力,无法覆盖普通用户。

近年来研究者不断提出的服务动态组合和服务推荐技术一定程度上可以满足环境变迁下人机物资源实例的替换调用过程。以 SOA 架构和的服务动态绑定技术和微服务架构下的服务编排可以实现依据服务环境来进行服务的替换,而以大数据智能分析技术代表的服务推荐则是探索式的服务动态组合过程。遗憾的是,以上技术都是面向服务资源,不能直接作用在人机物资源。

因此,针对流程类的生活场景,我们结合工作流执行技术和服务动态组合技术作为人机物融合应用流程执行环境构建方式来进行人机物融合应用构造。整个

方案设计上还需要设计人机物资源的一体化描述方法和面向用户的场景流程设计方法，结合应用流程执行环境构建方法来实现面向用户的人机物融合应用的工作流设计与执行技术方法。

### 3.2 研究方法描述

#### 3.2.1 人机物融合应用的研究框架

本文针对的是流程类的生活服务场景下所提出的人机物融合应用的工作流设计与执行技术方法，支持用户“即兴式”的以简单操作和易于理解的方式个性化自定义场景流程，然后系统能够自动化执行流程活动，并且自适应地调度相应人机物资源完成用户流程活动。该技术方法的整体框架如下图 3-1 所示，该架构可以分为三层结构，运行层、设计层和资源层分别对应了方法设计的三大模块。

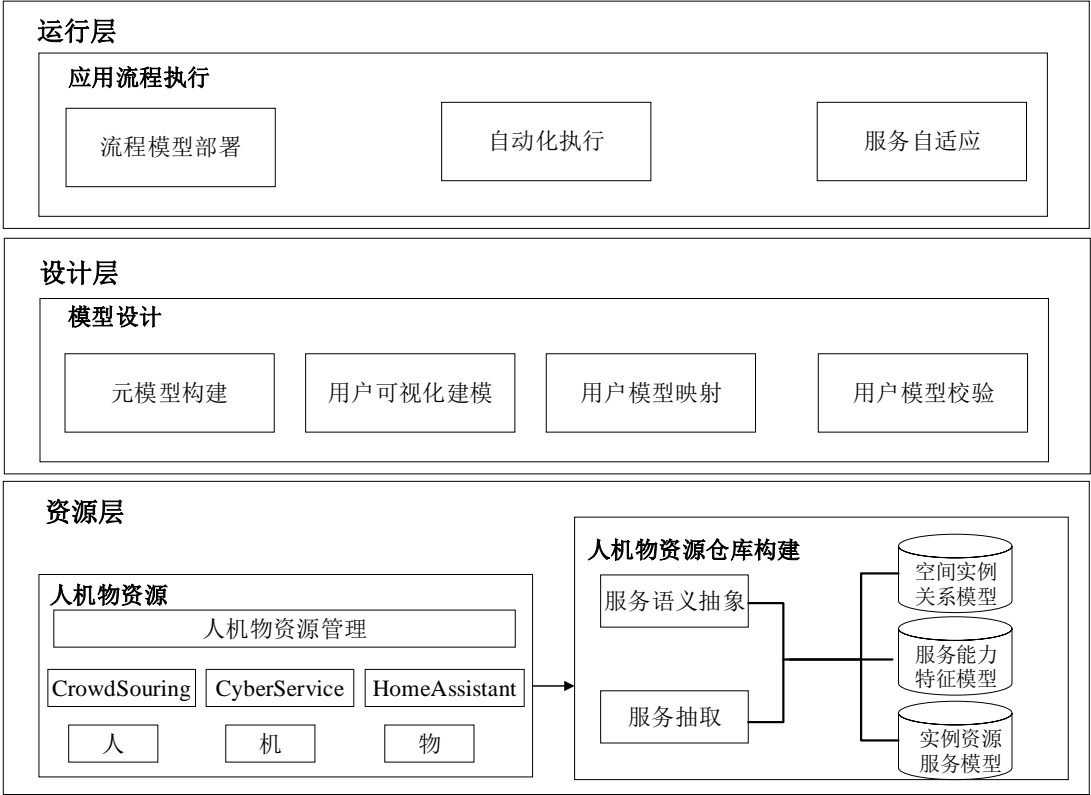


图 3-1 方法框架

如上图所示，本文提出的人机物融合应用的工作流设计与执行技术的方法架构分为以下三个核心模块的方法内容：

（1）资源层对人机物资源进行一体化描述和整合，目标是人机物资源语义仓库的构建。首先将人机物资源服务进行抽取，统一为 Rest 风格形式的服务；然后对服务进行语义抽象，对具体资源服务进行服务能力抽象和空间人机物资源

实例进行关联。该阶段的最终输出为空间人机物资源实例关系模型，人机物服务能力特征模型和人机物实例资源服务模型的语义仓库，以 Neo4j 图数据库为载体存储。

(2) 设计层构建用户可视化流程设计环境，对用户表现为用户可拖拽活动模型组合流程活动的流程设计。基于人机物资源仓库的语义关联，用户活动模型可直接映射关联人机物服务能力特征模型，从而实现间接关联该能力特征下的人机物服务和资源的超集。该模块基于提出的人机物应用工作流元模型进行用户流程定义前的用户流程定义工具构建，支持用户流程活动建模和用户模型的映射和校验。

(3) 运行层构建了人机物融合应用流程执行环境，基于扩展的工作流执行引擎实现人机物资源的自动化和自适应调度。应用流程执行环境关注的是人机物实例服务，通过人机物实例服务来调度其人机物实例资源。该过程包括了流程模型部署、执行和服务自适应，具体内容为将流程定义交由工作流引擎部署执行，并在自动化执行用户活动过程中基于资源语义仓库采用资源可用性、空间位置关系和用户使用习惯的推理规则动态的动态绑定资源服务，自适应调度人机物资源。

基于该设计方法，用户自定义流程的方法流程设计如图 3-2 所示。

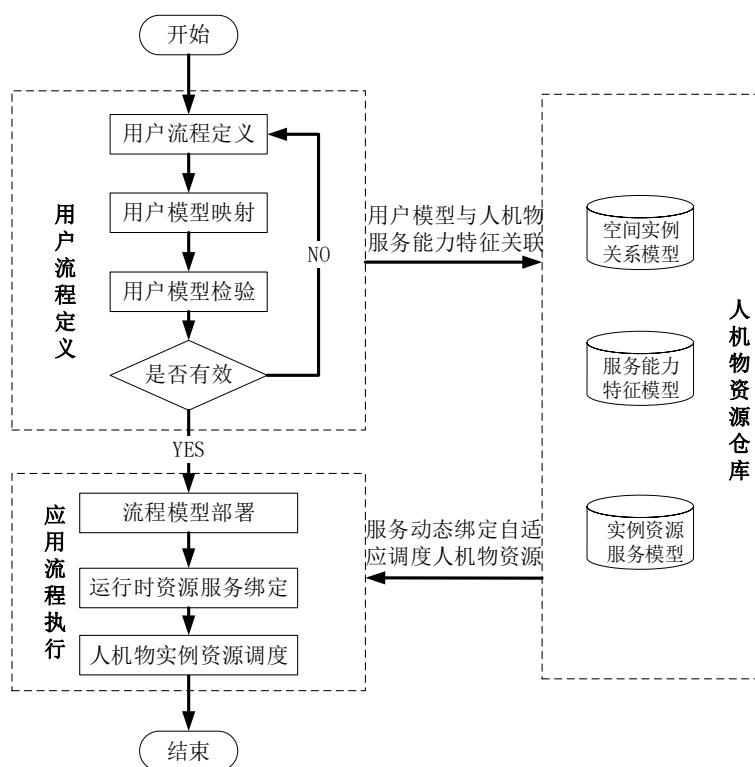


图 3-2 用户自定义流程活动的方法流程图

首先，用户流程定义和应用流程执行的方法流程都基于人机物资源语义仓库。用户在可视化用户流程定义过程中拖拽组合活动模型完成流程定义，此时用户活动模型直接关联到特定的人机物资源仓库的服务能力特征模型，间接关联到人机

物实例服务和资源候选结合。然后将用户活动模型映射为 xml 描述 BPMN 模型，并进行用户模型的有效性校验。通过校验后将流程定义部署至执行引擎自动化执行，在运行时过程中基于人机物资源仓库的资源可用性、空间位置关系和用户习惯动态绑定策略直接关联人机物实例服务，应用流程执行时自适应绑定人机物实例资源服务，通过该实例服务直接调度人机物资源完成用户活动。

### 3.2.2 人机物资源仓库构建方法

由于人机物资源异构特性，为了将人机物进行三元融合，需要将三类元素转化为资源服务统一表示。人机物资源仓库构建的目的在于建立人机物资源的语义关联关系，从而建立用户流程设计和系统人机物调度执行的关联关系。根据上述对整体流程的描述，人机物资源仓库的构建过程主要内容是人机物服务抽取和人机物服务语义抽象，提供了从人机物实例资源到用户活动抽象过程产物与人机物实例服务、人机物实例资源的关联方法。

人机物服务抽取是对人机物三种资源进行服务化和统一 Rest 风格化，人机物三类资源服务抽取定义和示例如下表 3-1 所示，具体的抽取方法将在下面详细介绍。

表 3-1 人机物资源服务描述

资源类别	抽取的服务定义	示例
人	针对“协作者”的众包服务	接咖啡众包、接外卖众包服务
机	已有的第三方服务	可直接调用接口的信息服务
	自定义流程活动的“前置”服务	活动报名服务、活动参数配置服务
物	抽取的 HomeAssistant 服务	控制投影仪、开关接口服务
	抽取的设备控制 App 的服务	APP 控制设备接口服务

“人”的服务抽取主要针对的是参与角色中“协作者”的人，具体表现形式是各类众包服务，需要注意的是这是一种自定义服务，本文抽取的是对象是自定义众包平台下的各类众包服务。

“机”的服务抽取本质上是对已有服务或者自定义服务的封装统一，比如喝咖啡参数表单填写服务和活动报名的服务等一般是流程活动的前置服务。本文因面向用户的流程设计摒弃了用户活动内参数配置的方式，前置服务一般是和相应资源的人机物服务组合而存在，用于弥补用户拖拽方式下人机物融合应用流程数据信息的完整度。

“物”的服务抽取主要集中在对 Hass 连接设备的访问控制服务的抽取。第

二章节我们提到 Hass 内核提供了状态和控制管理模块，对设备状态的访问方式为“IP:端口/api/states?Password= password”，返回的是以所有设备的冗余 json 数据串，需要进行信息提取抽取成设备服务接口。以上三类资源服务的在本文中都是以 Restful 风格的服务存在。

人机物服务语义抽象过程基于 Neo4j 图数据库对人机物服务进行建模，从底层设备资源到用户活动语义描述逐层向上进行抽象，对用户屏蔽了底层人机物资源。本文人机物资源仓库构建过程中设计的人机物资源冰山模型如图 3-3 所示。

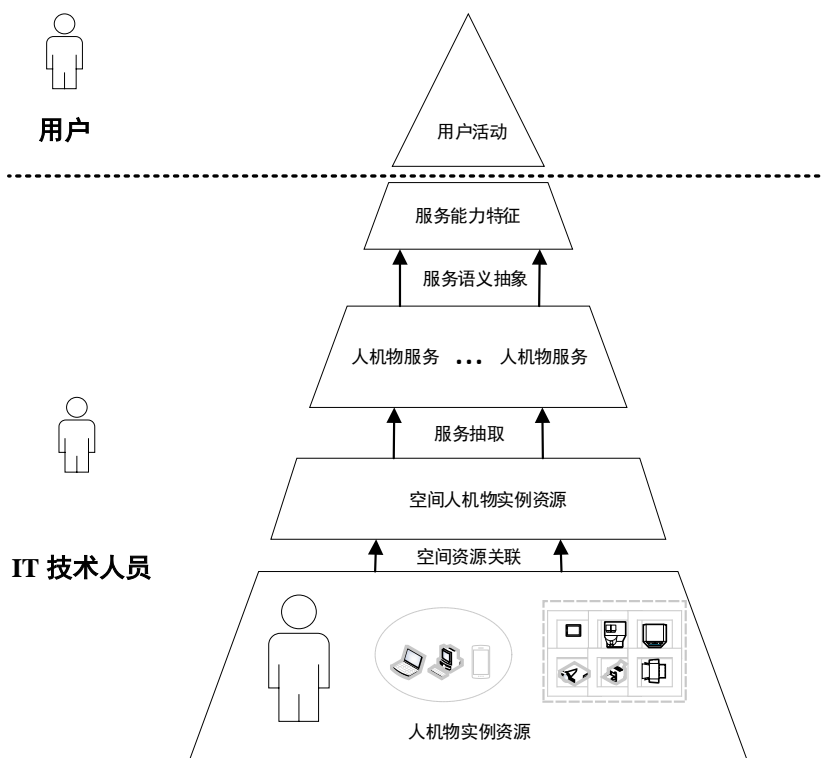


图 3-3 人机物资源冰山模型

该冰山模型自底向上分别面向 IT 技术人员和普通用户两种角色。面向 IT 技术人员阶段，上图中三个箭头所示的步骤分别对应地最终在 Neo4j 图数据库中构建了三种类型的模型：空间人机物实例关系模型，人机物实例服务模型，人机物服务能力特征模型。首先基于人机物实例资源进行空间资源关联，构建空间实例资源关系模型，它包含了空间语义信息，比如某空间 Room 下包含“has-a”某实例资源，该实例资源“located-at”空间 Room 下。针对空间下人机物实例资源进行服务抽取得到人机物实例服务，该人机物实例服务模型和下层的人机物实例资源模型是一对多的映射关系。接着针对人机物服务进行语义抽象，本文中我们针对人机物服务的能力特征进行语义抽象，遵循的语法结构为“动词+宾语”的动宾结构，形如“Open Light”，最终进一步抽象成人机物服务能力特征描述，该人机物服务能力特征模型和人机物实例服务模型也是一对多的关联关系，值得注意的是本文仅考虑的是同一种类人机物资源的服务能力特征抽象，即某个人机物服

务能力特征模型关联的该能力特征下同一类型资源的服务。最后，人机物服务能力特征模型向上直接关联映射到面向用户的流程设计的用户活动。例如两台咖啡机分别位于某机构单位的两间房间内，通过空间资源关联可以构建该机构单位到房间再到咖啡机的空间关联关系，基于空间下这两台咖啡机我们分别抽取了打磨咖啡服务，基于上述服务进一步语义抽象，以“Run Coffeemachine”服务能力特征描述了上面两个服务，场景冲咖啡的用户活动和该服务能力特征直接关联。

本文的最终目的在于通过用户自定义的流程来调度相应的人机物资源完成其活动。因此，最终面向的直接对象是人机物实例服务或者人机物实例资源（服务来调度资源）。基于上述构建的语义仓库，人机物资源仓库作为人机物语义知识提供抽象过程中任意过程产物到人机物实例资源和人机物实例服务的关联方法，比如用户、空间 room、能力特征以及用户活动能关联到人机物实例资源集合或者人机物实例服务集合方法。资源仓库构建实现过程将在第四章中详细介绍。

### 3.2.3 用户流程定义建模方法

根据上面的场景需求分析，满足用户个性化流程设计的需求需要设计一种用户易于理解和操作的用户交互方式。本文提出的人机物融合应用的流程设计方法面向的对象不仅仅是技术人员，还覆盖了普通用户，支持用户通过拖拽和组合活动模型的方式完成场景下流程活动的定义。用户通过该方法进行流程定义的过程也是一种用户流程建模过程，首先对人机物应用 workflow 进行元模型构建，如下如 3-4 所示。

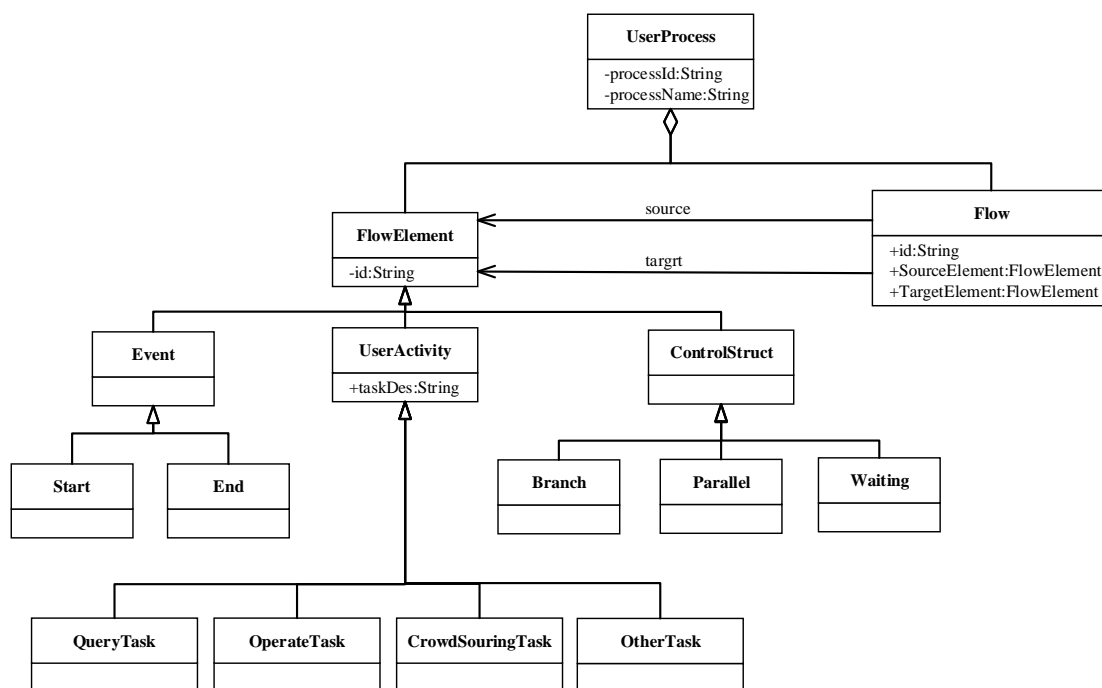


图 3-4 人机物融合应用工作流元模型



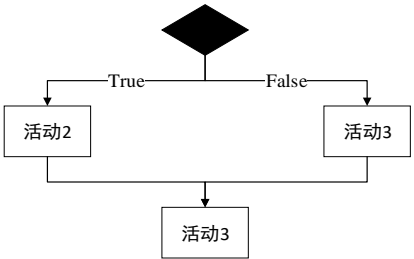
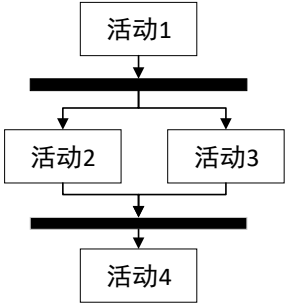
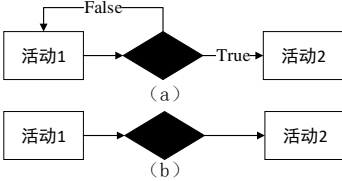
该元模型可以用来描述用户流程定义的工作流程以及模型元素之间的关联关系，用户根据需求场景自定义流程过程即基于流程定义工具组合活动模型，完成用户流程定义和用户建模过程。

UserProcess 就是用户场景下个性化自定义的流程，也就是用户建模后的模型。本文采用的是用户拖拽方式的建模方式，所以用户拖拽的是 FlowElement（流实体模型）和 Flow（连接模型），也反映了组合成 UserProcess 需要若干 FlowElement 和 Flow 模型。FlowElement 代表的是用户建模过程具有活动状态信息的活动模型和活动控制模型，包括了 Event、UserActivity 和 ControlStruct 三类模型。Event 模型包含 Start 和 End 模型两种活动状态表示模型，它们和普通流程图里的开始、结束状态节点表示具有相同语义。UserActivity 是用户活动模型，代表的是用户自定义流程中的每个用户活动。而上节中已经描述了的人机物资源冰山模型，在最顶层的用户活动可以以和人机物服务能力特征模型映射关联的用户活动模型来表示。该活动模型可以将继续细化划分得到具体类别的活动模型。

在用户场景流程设计过程中，我们完成一个场景流程所涵盖的活动类别可以归纳为：查询、操控、众包和其它注册、配置信息四大类活动。我们将活动类型进行细化和划分，将所有的活动按照以上四个类别生成标签来细化活动语义，将 UserActivity 活动模型派生为查询活动模型、操控活动模型、众包活动模型和其他模型，即 QueryTask、OperationTask、CrowdsourcingTask 和 OtherTask，它们各自带有相应的活动描述。用户活动模型由人机物资源仓库的人机物服务能力特征模型映射构建，所以以上四类活动模型的活动描述遵循相同的基本模型定义规则，即“动词+宾语”的动宾结构的定义规则，例如在 QueryTask 中以“Query+ 实例”命名，OperationTask 可以以“Open+ 实例”、“Close+ 实例”或者“Operation+ 实例”命名等等。

除此之外，FlowElement 模型中另外一类关键的模型是活动控制结构模型。活动控制结构模型和普通流程设计网关控制结构功能类似，用于控制整个流程的合并和分支结构，为人机物应用流程设计添加了更多的功能支持。活动控制结构模型包括了 Branch 选择控制模型、Parallel 并行控制模型和 Waiting 停等控制模型。Branch 选择控制模型即是 if-else 类型的分支结构的控制结构模型，是一类表示排他分支语义的控制结构。Parallel 并行控制模型则是并行结构的控制模型，支持并行分支活动的控制结构。这两种控制结构组合使用可以实现几乎所有的复杂流程控制结构。Waiting 停等控制模型则是一种不同于流程定义中已有的控制结构类型，它是一种自定义的用来支持人机物场景特性的停等控制模型，是符合人机物场景需求的自定义的复合控制结构模型，代表轮询等待情境下的停等语义，图中（a）是其普通的轮询语义描述，（b）是统一后的模型展现形式，上述三种控制结构模型和控制结构模型的语义示意关系如下表 3-2 所示。

表 3-2 控制结构模型图示

控制结构模型	控制语义示意
Branch 选择控制模型	
Parallel 并行控制模型	
Waiting 停等控制模型	

Flow 是连接 FlowElement 实体模型的连接模型，由于本文摒弃了用户流程设计过程中参数或者表达式手动设置的方式，采用的是面向用户的流程设计方法，所以将数据流和模型连接控制流统一为 Flow 连接模型，用来连接源 FlowElement 模型和目标 FlowElement 模型。

### 3.2.4 用户模型映射与校验方法

用户进行流程定义是通过上述方法构建的流程定义工具组合活动模型，这是用户建模过程，需要将用户模型映射为 xml 描述的 BPMN 模型。这样，才能够将用户模型转化为计算机解析读取的语言，在应用流程执行环境中得以执行。根据上一节对用户流程定义元模型的描述，用户流程定义的活动模型与 BPMN 模型存在对应的映射关系，通过这种映射关系完成了用户态和系统运行态的切换，支持用户设计的人机物融合应用流程能够自动化执行。

BPMN 文件是以 XML 描述为载体，以模型图形展示的标准格式的标注语言。如下图 3-5 所示，BPMN 的 XML 描述主要包括模型属性定义部分（第 5-8 行）、模型位置关联部分（第 9-11 行）和宏定义部分（其余数据行）。宏定义部分是通用描述部分，模型位置关联部分描述文件各个模型的位置和关联关系，模型属性

定义部分是流程解析的核心部分。本文的关键映射内容主要是模型属性定义和模型位置关联部分，而位置关联部分依赖于模型属性，它是模型属性的位置和模型大小的描述，支持 BPMN 模型以图形的形式展示。

```

1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <definitions xmlns:activiti="http://activiti.org/bpmn"           //宏定义
3  .....>
5    <process id="myProcess" isClosed="false" isExecutable="true">
6      <startEvent id="_2" name="StartEvent"/>
7      .....                               //流程模型定义
8    </process>
9    <bpmndi:BPMNDiagram id="Diagram" name="New Diagram">
10     .....                               //流程模型 diagram 描述
11  </bpmndi:BPMNDiagram>
12 </definitions>

```

图 3-5 BPMN 文件 XML 结构

用户流程模型和 BPMN 模型映射关系如表 3.3 所示，用户流程定义结束映射生成 XML 格式的 BPMN 文件，即流程模型定义部分以<BPMN 模型 属性>的格式，如<startEvent name="StartEvent" id=" StartEvent "/>的形式。除了 id、name 的基本标识属性，UserActivity 中的 class 属性是作为和所有任务的统一委托类，负责活动逻辑处理，exclusive 属性表示并发支持，async 属性用来配合活动的同步异步，Flow 中的 sourceRef 和 targetRef 表示连接的源 FlowElement 模型和目标 FlowElement 模型，conditionExpression 则是数据流条件表达式。而模型位置关联描述部分则是依据 BPMN 属性定义部分的进行位置的描述，它对于流程活动的执行不会产生影响，只是对 BPMN 文件的流程图展示产生影响。

表 3-3 用户模型和 BPMN 模型映射关系

用户模型		BPMN 模型	BPMN 属性
Event	Start	StartEvent	id、name
	End	EndEvent	
UserActivity	QueryTask	ServiceTask	id、name、class、exclusive、async
	OperationTask		
	CrowdsourcingTask		
	OtherTask		
ControlStruct	Branch	exclusiveGateway	id、name
	Parallel	parallelGateway	
	Waiting	exclusiveGateway 和 Flow 的复合模型	
Flow	Flow	sequenceFlow	id、sourceRef、targetRef、condition

用户流程定义和模型映射后,为了规避因用户疏忽或者恶意的错误操作给系统带来的稳定性和容错性风险,需要对用户流程模型进行有效性校验。本文所设计的校验规则按照模型的粒度范围由小到大,先进行单个模型的属性校验,再进行模型间的关联关系和关联规则的校验,最后是整个流程的用户模型有效性校验。单模型校验的目的在于预防用户恶意篡改用户流程设计的模型和属性的定义,通常情况下单模型的校验结果不会出现错误信息提示的。模型校验处理流程设计如下图 3-6 所示。该处理流程主要是用于避免用户因个人的失误或者错误操作使得流程无法被引擎解析执行甚至是执行引擎错误执行,支持向用户反馈校验错误信息,提示用户重定义。

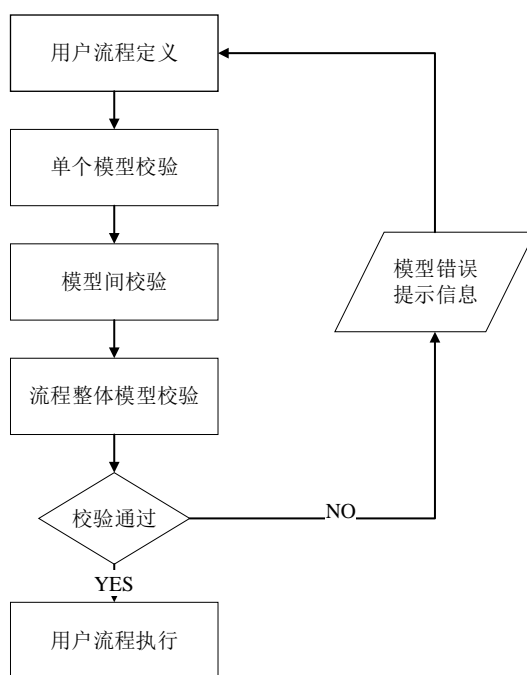


图 3-6 用户模型校验流程

用户模型校验过程单个模型校验是能最小粒度覆盖的模型校验,将对所有用户模型属性进行合法性校验,是针对单个模型的定义和属性描述进行基本的合法性校验过程。模型间校验是对人机物流程场景下用户模型间的先后顺序和语义顺序的关联关系进行校验,是针对人机物特征进行的约束性语义校验过程。流程整体模型的校验则是对整个用户模型组合成的流程进行有效性校验,针对的是整个用户自定义的流程全部分支可达路径进行的合法性校验过程,采用了基于深度遍历优先的从开始到结束状态可达路径分析的校验过程,整个校验算法设计描述下面算法伪码 1 所示。代码第 4-7 行基于执行引擎原生校验器对上一节中描述的用户模型以及其模型属性进行合法性校验,第 8-10 行基于校验器工厂自定义模型间检验器对用户模型间前后语义顺序关系进行语义检验,第 11-19 行自定义流程整体模型校验器基于构建好的模型图结构关系,采用全路径可达分析的方法对整个人机物活动流程进行有效性校验。

**算法伪码 1：用户模型校验算法**

```

1  输入： 用户流程定义模型 UsrProcess
2  输出： 校验错误信息集 Validate
3  UsrProcess 映射为 BpmnXML 经转换器转化为 BPMNModel;
4  创建模型校验器工厂，注册获取模型校验器；
5  获取 BPMNModel 校验器校验单个模型元素；
6  if 存在属性设置错误
7      将模型属性错误信息加入 Validate 错误信息集；
8  自定义模型校验器校验模型间关联；
9  if 两个模型间不符合语义和顺序规则
10     将模型间关联错误信息加入 Validate 错误信息集；
11  自定义模型校验器校验模型整体关联；
12  获取用户实体模型和带有源模型和目标模型关联的连接模型；
13  构建用户实体模型和连接模型图结构关联关系，从“StartEvent”进行深度优先遍历；
14  while 遍历构成应用流程中的所有用户模型
15      while 当前用户模型不为“EventEvent”结束模型
16          if 当前用户模型直接关联的下一个模型为 null
17              将当前模型不可达错误信息加入 Validate 错误信息集；
18          else
19              当前用户模型更新为当前用户模型直接关联的模型；
20  return Validate 错误信息集；

```

### 3.2.5 自动化与自适应执行方法

用户流程设计和模型校验完成后，将进入人机物融合应用流程执行阶段，系统自动化执行用户活动的过程中通过调用人机物服务来调度相应的人机物资源完成用户活动。通过上面我们对场景的分析，本文对 Activiti 工作流引擎进行扩展来支持人机物应用流程的自动化执行，并基于人机物资源语义仓库从资源可用性、空间位置关系和用户习惯三个方面的服务动态绑定策略来自适应调度人机物资源完成用户活动。

在 3.2.4 节中我们描述了 UserActivity 的四种标签分类下的活动模型，其中 class、exclusive、async 三个属性字段是执行引擎执行用户活动过程不可或缺的关键字段。本文 calss 属性设置用来统一以一个委托类对所有活动进行委托代理，

即用来设置用户活动的人机物服务调用的统一入口。`exclusive` 和 `async` 的结合使用用来进行流程的同步和异步实现,完成流程活动的挂起和执行操作。这也是由活动类型决定的,大部分活动关联和映射的人机物服务是一些自动化的调用即返回式的短连接服务,也有如众包和其他活动需要人确认或者注册来返回的长连接服务。后者活动类型的异步过程在基于 `exclusive` 和 `async` 属性字段在执行态表现为挂起和继续执行。人机物服务调用和执行将调度相应的人机物资源来完成用户活动的自动化执行。

完成上述用户流程设计和模型映射、校验后,人机物融合应用的流程执行过程如下图 3-7 所示。该阶段从用户态流程设计切换到了系统执行态,完成人机物服务的调用和资源的调度。应用流程执行环境下首先将流程定义进行部署,解析为工作流 BPMN Model 进行执行,执行引擎活动模型委托类监听 `UserActivity` 具体活动,该委托类作为不同活动对应的人机物服务调用的统一入口,并基于人机物服务的动态绑定策略,调用人机物服务来调度人机物资源完成活动。

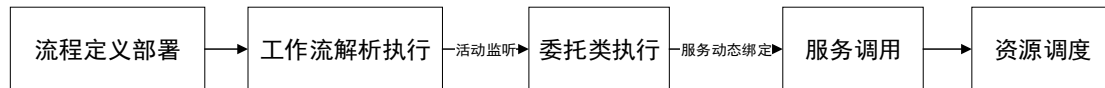


图 3-7 人机物融合应用执行过程

本文提出的人机物应用执行服务动态绑定策略类似于基于 SOA 架构的 Qos 属性的动态服务组合方法,采用了基于人机物资源语义仓库下的推理规则进行的最优服务匹配算法动态绑定了实时环境下最合适完成某活动的服务。在资源仓库构建过程中,我们分别构建了空间人机物实例关系模型、人机物实例服务模型和人机物服务能力特征模型三类模型,基于语义仓库的关联方法从实例资源服务可用性、空间位置和用户习性(人机物资源调度次数)三个方面设计规则推理算法最终关联唯一的人机物服务模型,并间接关联调度某个资源实例。该人机物融合应用自适应调度算法描述如下伪码 2 所示。该算法大致流程为:由执行引擎委托类监听作为入口,输入为用户活动模型。初步直接关联人机物服务能力特征模型(第 3 行),通过资源仓库关联方法关联获取到人机物实例资源服务模型集合,进一步从资源可用性方面筛选得到人机物实例服务候选集(第 4-6 行);再次基于空间位置关系进一步关联匹配,通过资源仓库关联方法获取人机物实例关系模型中距离用户最近的人机物资源实例的服务候选集(第 7-14 行,这里本文认为和用户在同一单位房间下的所有资源最近且相同);最后基于用户习惯方面最终关联匹配上一次服务候选集中用户调度次数最多的人机物资源服务(第 15-17 行),作为最终从用户活动模型到人机物实例服务模型一对一的动态服务绑定结果(第 18 行)。上述三次关联匹配过程中关联匹配到唯一实例服务即可以返回。它作为实时环境下自适应的人机物实例服务,最终基于资源仓库关联关系来直接调度该服务对应的人机物资源完成用户的活动。

**算法伪码 2：人机物服务动态绑定算法**

1	<b>输入：</b> 用户活动模型 <b>UsrActivity</b>
2	<b>输出：</b> 人机物实例服务 <b>Service</b>
3	用户活动模型 <b>UsrActivity</b> 直接映射关联人机物资源仓库的服务能力特征模型；
4	<b>while</b> 遍历人机物资源仓库关联方法关联获得该能力特征下的人机物服务集合；
5	<b>if</b> 当前人机物实例服务状态为 <b>available</b>
6	将该人机物实例服务加入候选集 <b>candidates</b> ；
7	<b>while</b> 遍历候选集，实例资源服务模型关联映射人机物空间实例资源关系模型和实例
8	资源，获取位置信息；
9	<b>if</b> 存在实例资源所在区域 <b>Room</b> 和用户所在区域 <b>Room</b> 相同
10	将该人机物实例服务加入候选集 <b>candidates</b> ；
11	<b>else</b>
12	基于欧式距离计算公式计算实例资源和用户的距离 <b>dis</b> ；
13	更新最小距离值 <b>dis</b> ，并记录此时的实例服务
14	将最终最小值 <b>dis</b> 对应的人机物实例服务加入候选集 <b>candidates</b> ；
15	<b>while</b> 遍历上述人机物实例服务集，关联得到人机物实例资源模型；
16	<b>if</b> 该资源实例使用次数最多；
17	将该人机物实例服务加入候选集 <b>candidates</b> ；
18	<b>return</b> 候选集 <b>candidates</b> 中唯一的 <b>Service</b> ；

通过上述描述的人机物服务动态绑定算法，输出得到和用户活动最终关联映射的人机物实例服务。该实例服务通过人机物资源仓库的关联方法可以直接关联到人机物实例资源，最后调度该资源完成用户活动，实现实时环境下的自适应。

### 3.3 本章小结

本章节首先从人机物融合典型场景实例进行分析，进而提出了人机物融合应用的工作流设计与执行方法，并结合场景对技术方法进行了关键技术的需求分析。然后，对本文总体研究方法和研究思路方法展开了描述。人机物资源仓库构建方法包括服务抽取和服务语义抽象，支持从人机物资源到用户活动的语义抽象过程，从而建立了用户流程设计和系统人机物调度执行的关联关系；人机物流程定义的构建方法基于人机物融合应用工作流元模型设计了用户流程定义方法，并支持模型映射、校验。人机物融合应用流程执行环境构建方法基于执行引擎自动化执行用户流程活动并基于服务动态绑定策略自适应调度人机物资源完成活动。

## 第四章 人机物融合应用的设计与实现

继上一章节重点描述了人机物融合应用的工作流设计与执行技术的基本设计思路和研究方法,本章将对人机物融合应用的工作流设计与执行工具的设计和实现展开详细的介绍。首先,介绍了本文工具系统的组成架构,然后围绕其中的三个核心组件人机物资源仓库、用户流程定义工具和运行时环境分别展开详细设计和实现的描述。

### 4.1 人机物应用系统工具架构

基于在第三章中介绍的总体研究框架和方法流程描述,我们设计并实现的人机物应用的系统架构如下图 4-1 所示。该系统组成包括了三大核心组件:用户流程定义工具、系统运行时环境和人机物资源仓库。

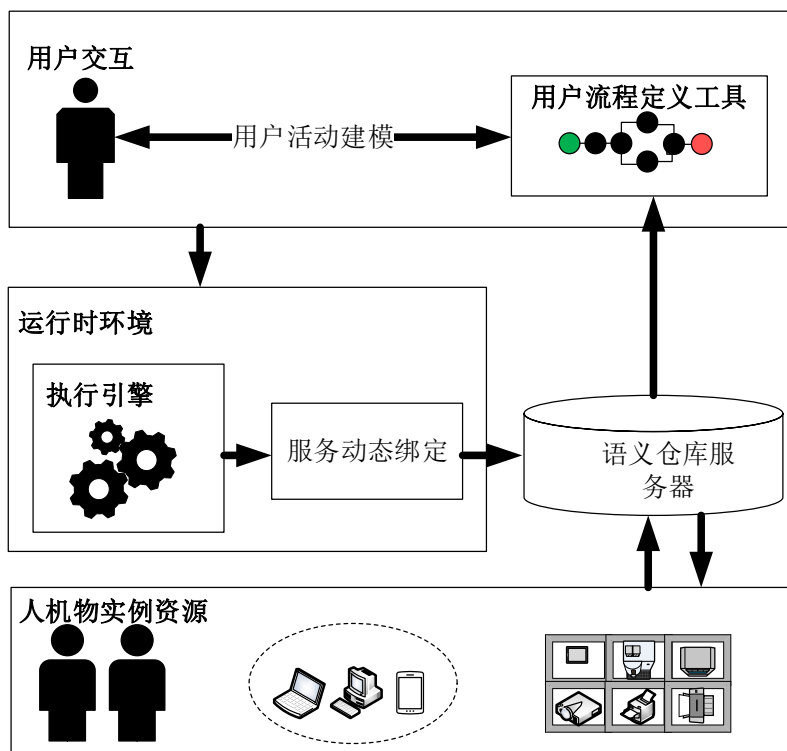


图 4-1 人机物融合应用系统架构

人机物资源仓库主要包括服务抽取和服务语义抽象,对人机物资源进行一体化描述和管理,建立了人机物资源的语义关联关系,从而建立了用户流程设计和系统人机物调度执行的关联关系。最终构建的是人机物空间资源关联模型,人机物服务能力特征模型和人机物实例服务模型集合,并提供了人机物实例服务和人机物实例资源的最终关联方法。

用户流程定义工具是人机物融合应用流程设计的核心组件工具,工具构建包



括元模型构建、工具模型构建和实现,支持用户流程个性化设计过程,支持用户流程建模后用户模型映射和模型校验,将用户流程模型映射为 BPMN xml 描述,基于 BPMN 模型对用户模型进行有效性校验。

运行时环境是人机物融合应用流程执行的核心运行组件,支持用户定义的流程活动自动化执行,自适应调度人机物资源完成用户活动。简单来说,通过用户流程定义工具的输出结果 BPMN 交由执行引擎解析自动化执行,继而在运行态基于资源可用性、空间位置关系和用户习惯的服务动态绑定策略自适应调度人机物资源完成用户流程活动。

## 4.2 人机物资源仓库构建

人机物资源语义仓库的构建是围绕着原生的人机物实例资源进行的一体化描述和管理过程,从松散、存在异构特性的人机物资源抽象关联到三元融合的用户活动,自底向上支持人机物场景下三元融合应用。该组件的作用在于分别提供人机物实例资源和服务与用户流程设计活动、系统运行时环境调度执行的基本关联方法,从而实现用户流程设计和系统人机物调度执行的关联和映射。人机物资源语义仓库构建过程分为三个构建步骤:人机物资源服务抽取、人机物服务语义抽象和人机物资源仓库的最终构建。

### 4.2.1 人机物资源服务抽取

在 3.2 节中我们描述了服务抽取的来源,人机物资源服务可用分别从 HomeAssistant、已有的信息服务和自定义众包平台提供的众包服务抽取。

HomeAssistant 是设备接入的物理设备管理平台,可以实现对设备的自动发现和接入并提供状态管理和控制管理。本文中我们针对“物”(物理设备)的服务抽取分为状态查询和设备操作两类服务,状态查询 Api 返回结果是所有服务对应的 json 数组,如下图 4-2 所示,我们需要进一步抽取其中 entityId 字段和 state 字段,封装为新的设备查询服务。

```
{
  "attributes": {"Open since": "300", "battery_level": 49.0, "device_class": "opening"},
  "context": {"id": "a60b842524454f8a8210191e92473452", "user_id": null},
  "entity_id": "binary_sensor.door_window_sensor_158d0001f37c19",
  "last_changed": "2019-09-08T02:57:09.588206+00:00",
  "last_updated": "2019-09-08T03:02:09.092912+00:00",
  "state": "on"
}
```

图 4-2 Hass 设备的 Json 信息

物理设备的操控类服务的抽取基于 Hass 组件的配置文件的 url、method 和 payload 字段封装为资源实例的操控类服务,操控空气净化器服务如图 4-3 所示。

```
@app.route('/Operation/AirPurifier')
def OperateXiaoMiAir():
    //请求参数 data 为设备的 entityId
    response = requests.post('http://ip:port/api/services/fan/toggle', headers=header, data=data)
    update_mysqlDef('fan.xiaomi_air_purifier_2s') //更新该设备使用次数, 参数为设备 id
    res = {'state': 'XiaoMiAir_sucess'}
    return json.dumps(res)
```

图 4-3 抽取的操控空气净化器服务

“机”对应的信息服务指的是自定义的 web 服务或者可调用的已有 web 服务。严格意义上来说,“人”的服务抽取来源众包平台的众包服务也是已有的特殊的信息服。人机物资源服务抽取阶段构建的是人机物实例资源服务模型,该过程将人机物资源提供的所有服务都抽取为 Restful 风格的服务。

## 4.2.2 人机物服务语义抽象

完成了人机物实例资源的服务抽取,需要对抽取的的人机物服务进行语义抽象。人机物服务语义抽象是对具有同一类服务能力的资源实例服务进行的语义抽象,是人机物融合应用技术用户流程设计和系统执行关联的关键方法。本文的语义抽象规则是基于服务能力特征进行的,并且语义描述严格遵循“动词+宾语”的命名规范,实例资源服务语义抽象如下表 4-1 所示。

表 4-1 实例资源服务语义抽象示例

资源实例	资源实例服务	抽象语义
Light1	① ip1+port+?entityId+/getLightState	Query Light
	② ip1+port+?entityId+/OpenLight1	Open Light
	③ ip1+port+?entityId+/CloseLight1	Close Light
Light2	④ ip2+port+?entityId2+/getLightState	Query Light
	⑤ ip2+port+?entityId2+/OpenLight2	Open Light
	⑥ ip2+port+?entityId2+/CloseLight2	Close Light

上表是以抽取两台 Light 相关的服务进行语义抽象的结果。表中 1 和 4、2 和 5、3 和 6 基于服务能力特征分别抽象为 Query Light、Open Light 和 Close Light,遵循上述的“动词+宾语”的命名规范。服务语义抽取阶段构建的是人机物服务能力特征模型,该模型带有能力特征服务的属性描述,如公式(4.1)所示:

$$Model = \langle AbilityDes, RequestMethod, Input, Output, ServiceType \rangle \quad (4.1)$$

该公式描述了服务能力特征模型由五元组表示，对应了五个属性描述，分为是抽象语义的能力特征描述，请求方式（get、post），输入、输出参数以及服务类型（是否支持回调）。一个服务能力特征模型由上述五个属性来描述，一个实例资源对应多个实例服务，多个实例服务可以抽象对应一个能力特征模型。

### 4.2.3 语义仓库最终构建

本文采用的是基于 Neo4j 来对人机物资源进行建模和语义关联，其好处在于图数据库在分布式系统和海量数据关系建模的实用性和高效性，尤其是可以在一个大规模的复杂的关联模型关系中抽象出一个有用的结构或者关联关系。

首先，我们对空间人机物资源实例建立空间关联关系，分别采用“has-a”和“located-at”关联关系建立空间下实例资源的关系，表示人机物实例资源 located-at 空间下，空间 has-a 实例资源语义描述，模型关系表示如下公式(4.2)所示：

$$Relation = \langle Model1, hasA, locatedAt, Model2 \rangle \quad (4.2)$$

Model1 和 Model2 可代表空间区域 room 和人机物资源实例模型，描述了人机物资源实例模型在空间下的关联联系。

接着对空间实例资源进行服务抽取，构建实例资源服务，比如上文提到的 Ligth 的状态查询服务，构建的实例资源服务模型直接以 Neo4j 节点的形式存入。该模型约束范式如下公式(4.3)所示：

$$Model = \langle EntityId, AbilityDes, Url \rangle \quad (4.3)$$

该约束公式描述的是人机物实例服务模型，由上述三元组属性描述，它和人机物能力特征模型通过 AbilityDes 属性构成多对一的关联映射关系。

最后对实例资源服务进行语义抽象，构建了服务能力特征模型，该模型在 neo4j 节点表示形式为：name 属性为抽象语义描述属性，其他四个属性分别对应同名的 neo4j 节点属性描述。基于上述构建过程空间实例资源关联关系模型、实例资源服务模型和服务能力特征模型三者之间语义范围的关系如下图 4-4 表示所示。

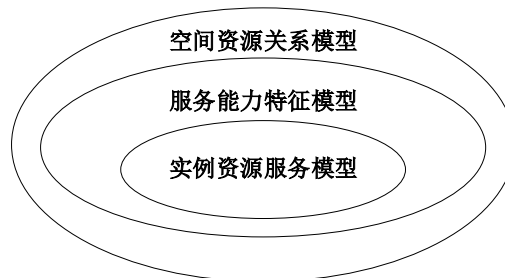


图 4-4 模型语义范围示意图

经过人机物资源服务抽取和抽象，建立了从用户活动到人机物实例服务模型，再到人机物实例资源的关联关系。人机物资源语义仓库的最终构建基于以上模型的构建，提供了一系列的关联方法。包括用户活动到人机物服务能力特征模型的关联方法，人机物能力特征模型和人机物实例服务模型的关联方法，人机物实例服务模型和人机物实例资源的关联方法，以及空间人机物资源关系模型下人机物实例资源和空间区域、用户的关联方法。基于上述关联方法，人机物资源语义仓库提供了一致性关联方法，即上述人机物仓库任意元素和人机物实例服务模型、人机物实例资源的最终关联方法。具体描述为：用户、用户活动、空间区域（room）、服务能力特征模型、人机物实例资源模型组成的五元组集合的非空子集都可以直接关联人机物实例服务模型；用户、用户活动、空间区域（room）、服务能力特征模型、人机物实例服务模型组成的五元组集合的非空子集都可以直接关联人机物实例资源模型。

4.3 用户流程定义工具构建

用户流程定义工具是和用户直接交互的用户流程活动设计工具，支持用户可视化地拖拽和组合活动模型元素的方式来个性化定义场景流程。前面 2.3.2 节中介绍了人机物资源冰山模型，原生的人机物实例资源向上逐层抽象，从面向人机物资源到面向 IT 技术人员到面向用户的用户活动模型。用户和用户流程定义工具交互构造的基本单元就是用户活动模型，即代表场景流程下的一个语义活动。

基于前面构造的元模型，本文设计并实现的用户流程定义工具的架构如下图 4-5 所示，大体可以分为四个模块：UI 模块、用户流程设计和可视化模块、以及用户模型映射模块和用户模型校验模块。

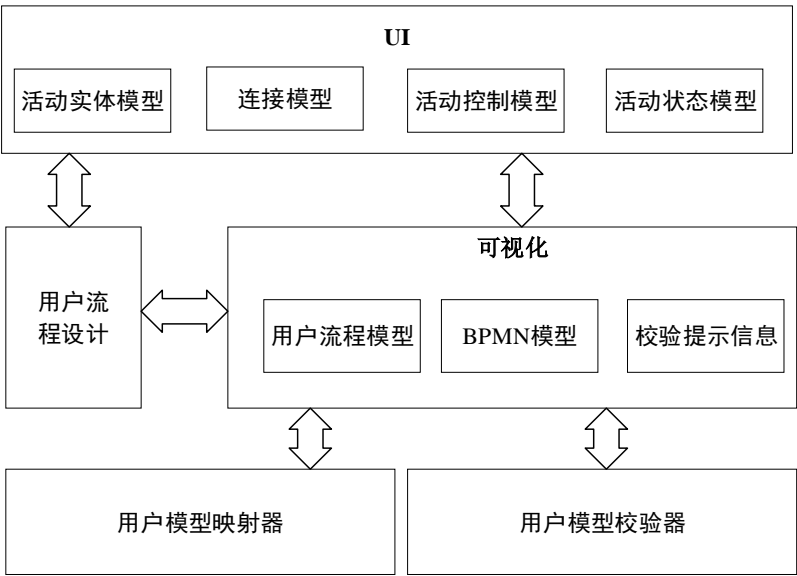


图 4-5 用户流程设计工具架构图

UI 模块主要包括工具框架界面和可拖拽的模型元素，是用户交互工具的基础元素。可拖拽的模型元素包括和语义仓库的能力特征模型一对一映射的活动实体模型，以及工具栏里的连接模型、控制模型和开始、结束的状态模型。

用户流程设计模块和可视化模块可以统一为用户模型构造器，主要用于用户对 UI 模块中的基础模型进行拖拽流程建模，并在可视化模块中展示用户流程模型、BPMN 模型 xml 描述和模型校验信息。

用户模型映射器用来将用户模型映射为执行引擎所能解析的以 XML 描述的 BPMN 模型。用户模型校验器则是对用户模型进行有效性校验，并且将校验结果包括校验通过和校验错误信息反馈显示，提示用户进行修改和重定义。

### 4.3.1 用户流程定义工具设计

基于该工具的基本框架模块，本文设计实现的用户流程定义工具的界面截图如下图 4-6 所示，支持用户基于该工具进行个性化场景流程活动的定义。

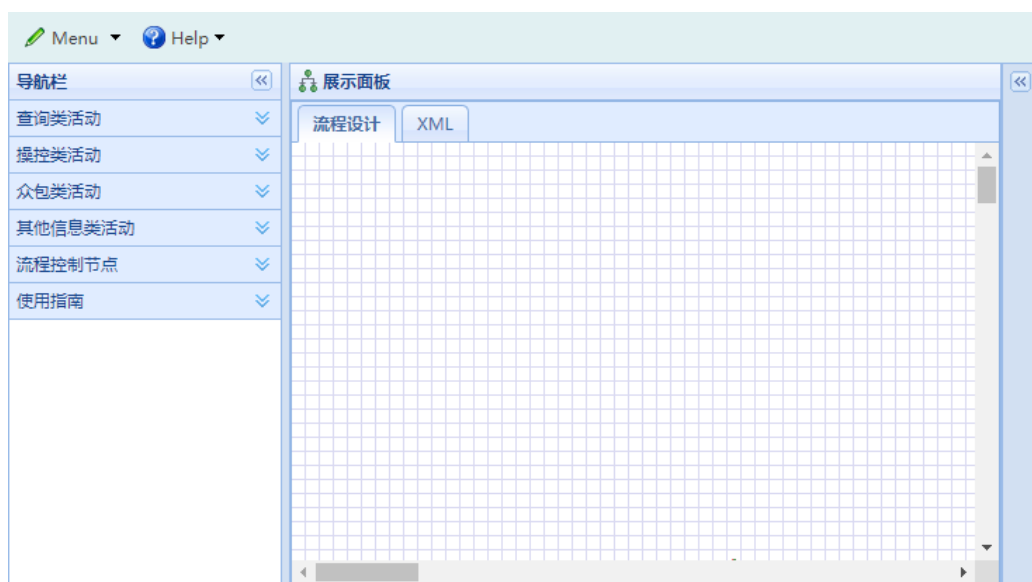


图 4-6 用户流程建模工具界面

从该图中我们可以看出，该用户流程设计工具分为四个部分：

(1) UI 部分，除了整个工具界面还包括两部分，一部分是左侧导航栏中的查询、操控、众包和其他四类标签分类的用户活动模型，另一部分是左侧导航栏中工具栏里包含开始和结束状态模型以及三类控制结构模型的非活动实体模型；

(2) 可视化部分，图中用户流程设计工作区中的流程设计支持用户从左侧导航栏内拖拽的模型元素进行用户流程设计。XML 工作区可视化表示的是经用户模型映射器映射得到的 BPMN 模型的 xml 描述。另外，用户模型经由用户模型校验器输出的校验结果内容会反馈显示在工作区，包括了用户定义的流程校验通过和校验不通过的具体错误提示信息，支持用户进行重定义。

(3) 用户模型映射部分, 该部分用来在用户流程设计过程中对用户构建的模型进行实时的模型映射, 即从图中流程设计工作区中的用户模型到 XML 描述可视化部分中的 BPMN 模型 xml 描述的映射过程。

(4) 用户模型校验部分, 该部分是在最上面的菜单栏中下拉菜单中的模型校验菜单对应的功能, 将用户定义的流程模型进行模型的有效性校验, 校验结果内容将反馈显示在可视化工作区。

用户完成了模型校验并通过才能将最终结果 BPMN 模型交由运行时环境的执行引擎来解析和执行, 即需要点击菜单栏中的下拉菜单中开始执行菜单来从用户流程定义过程进入系统自动执行阶段。

该工具的设计上采用的是 Eclipse 插件风格的支持 Online 的 Web 界面工具。工具实现上, 采用了 Draw2d 技术, 借助 Draw2d 这个图形库实现模型元素图形绘制, 编写 Js 来实现拖拽和其他复杂的功能。

### 4.3.2 流程定义工具模型构建

从我们提出和构建的元模型可以看出, 整个工具的用户模型可以分为四大类: 活动模型, 状态模型, 控制模型, 以及连接模型, 也就是工具界面中导航栏下的可拖拽模型元素。

本文基于 Draw2d 图形库对 Draw2d 库进行派生和扩展来构建和绘制这些工具模型。用户流程模型 UserProcess 由以上四类模型组合而成, 其中用户活动模型, 活动状态模型, 活动控制模型三类模型都是从 Draw2d 的 Node 类进行派生和扩展, 在用户流程模型中属于实体模型, 具体构建实现的结构如下图 4-7 所示。

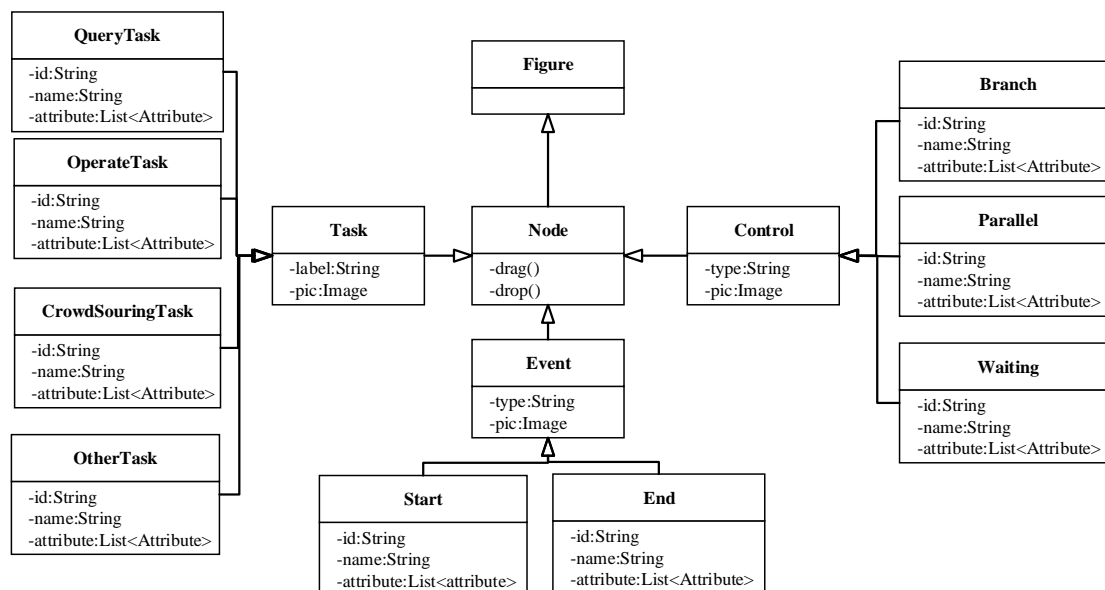


图 4-7 实体模型构建结构图

Flow 连接模型是连接实体模型的辅助模型，本文提出的人机物融合应用的流程设计方法覆盖了普通用户，将控制流和数据流统一为了连接模型，避免了用户配置连接模型的参数或者条件的方式，将数据流尽可能地封装体现在流程流转过程内部，连接模型的构建结构如图 4-8 所示。每个实体模型都有 Port 连接点，每个连接模型分别连接着源连接点和目的连接点，控制着流程的数据和顺序流转。另外，流模型内部描述的属性可以描述流数据内容，完成了和实体模型的关联。

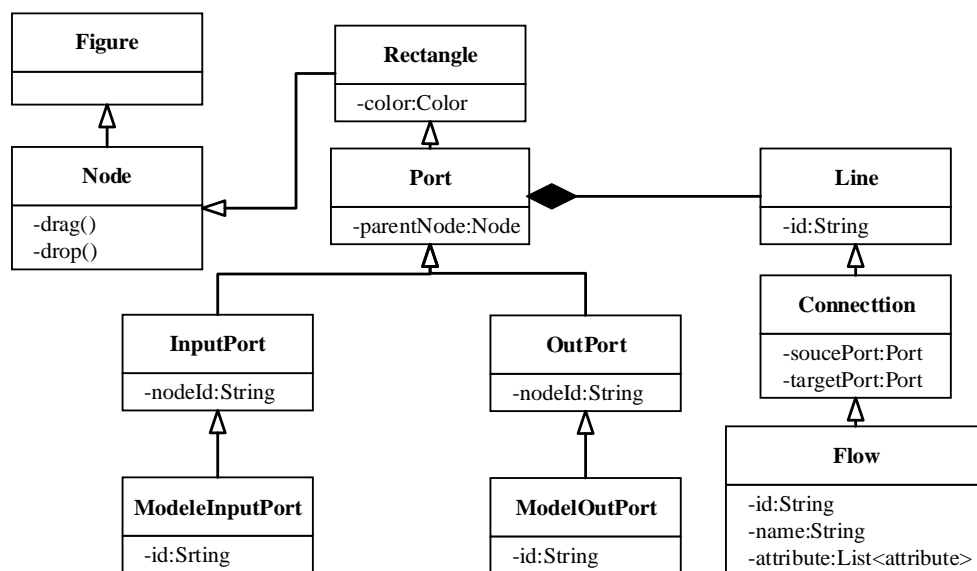


图 4-8 连接模型构建结构图

值得注意的是，在活动模型构建过程中，除了绘制图形模型来表示，还将模型的语义信息附在图形模型上来统一构建活动模型。本文的用户流程设计工具模型中的活动模型即用户活动模型下的四类活动模型，和人机物冰山资源模型的最顶层的用户活动直接映射关联，并且活动模型和人机物资源仓库中的人机物能力特征模型是映射关联的。也就是说，三者的关系描述为用户活动直接关联活动模型，活动模型通过语义仓库的关联方法直接关联人机物服务能力特征模型，继而建立了用户活动与人机物服务能力特征模型的关联关系。

构建用户活动直接关联活动模型的关联方法实现上，我们采用的是语义描述匹配方法，用户模型的语义描述和能力特征模型描述处理上一样，用户模型 id 属性的命名遵循的是“动宾”结构的命名范式，是一种类驼峰命名规则的单词首字母全大写规则，如“CloseDoor”。比如对于一个查询服务活动模型，在左侧模型栏里创建了查询活动模型图形，其模型语义描述就是取该模型 id 作为活动描述，用户活动得以直接关联语义相匹配的用户活动模型。而在人机物语义仓库从用户活动到人机物服务能力特征模型的关联方法的实现则基于不同“动宾”命名约束的语义描述字符匹配方法实现。

4.3.3 用户模型映射

用户流程定义是用户建模过程，输出结果是用户模型，需要将用户模型映射为运行时环境下执行引擎能够解析和执行的描述语言，该映射过程是一种边构造边映射的动态映射过程。本文设计的用户流程定义工具中用户模型映射器负责将可视化区域流程定义工作区的用户模型映射为以 xml 描述的 BPMN 模型，并可视化显示在 xml 面板部分。

3.2.4 节中描述了 BPMN 模型的 XML 定义规范，分为三部分：文件宏定义、模型和属性定义以及 diagrams 定义,文件宏定义主要包括一些文件头声明和命名空间，可以用人机物特征下通用的描述来填充定义；模型和属性定义是 xml 模型描述主体，描述了模型和属性信息；diaogram 定义则依赖于模型属性定义部分，描述了图示模型的 diaogram 和模型之间的关系。XML 是一种遵循 dtd 规范的树状结构，从根元素到子元素再到叶子元素，每个元素可以由若干属性或者文本描述。用户模型映射器的输出结果就是描述 BPMN 模型的 XML,可视化显示在 xml 栏下的 canvas 中，如下图 4-9 所示。

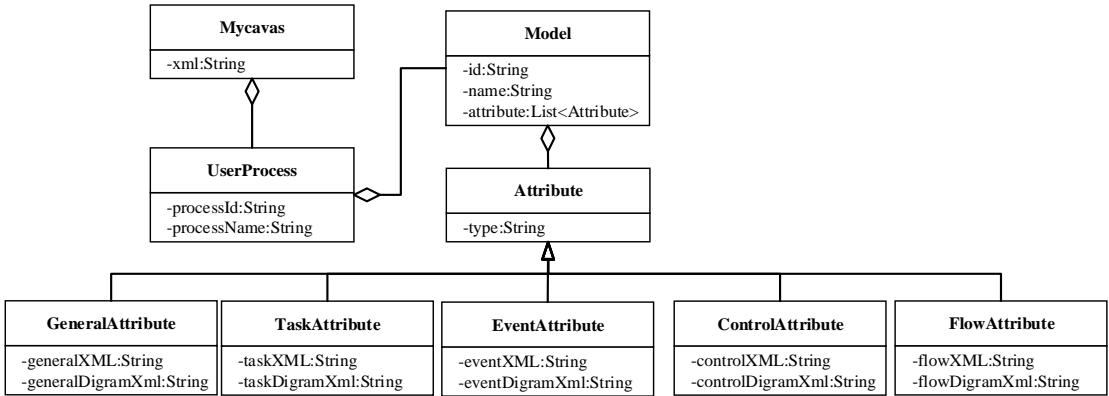


图 4-9 模型映射的关系图

上一章节流程定义所涵盖的所有模型可以统一抽象为 Model，用户流程模型由这些 Model 组合实现用户建模。每个 Model 都关联了一个描述 XML 映射关系的 Attribute 属性对象，用户映射器的核心就在于用户模型 Model 和 Attribute 之间的关联和映射关系。

GeneralAttribute 描述的是 XML 的文件宏定义部分的内容，在 BPMN 的 XML 描述里我们可以用一种通用的头文件和命名空间定义来适用在所有的 BPMN 描述里。其映射的 XML 部分为 XML 根元素<definitions>、<bpmndi:BPMNDiagram>和<bpmndi:BPMNPlane>子元素，属于通用的 XML 描述部分。后面其余四个派生部分描述的都是 XML 的模型与属性定义部分的内容和 diaogram 定义部分。

TaskAttribute 描述的是代表活动模型对应的具体的 XML 元素：<process>子元素下的<serviceTask>元素。本文研究的人机物融合应用在运行时环境流程执行



中的研究对象为人机物实例服务，鉴于此，活动模型统一映射为 BPMN 的 ServiceTask 模型，<serviceTask>元素属性有 id, name, async, class, exclusive。另外，也包括<bpmndi:BPMNPlane>子元素下的<BPMNShape>元素，用来描述 label 和坐标信息。

EventAttribute 描述的是状态模型即活动开始模型和活动结束模型对应 XML 内容：<process>子元素下的<startEvent>元素和<endEvent>元素，其属性值包含 id 和 name。依赖于实体模型，还包含对应<bpmndi:BPMNPlane>子元素下的和模型对应的<BPMNShape>元素描述。

ControlAttribute 描述的是控制模型对应的具体的 XML 元素：<process>子元素下的<exclusiveGateway>元素、<parallelGateway>元素和代表自定义的控制结构的 Waiting 模型所对应的<exclusiveGateway>元素和额外的<sequenceFlow>的复合结构，其属性定义包括 id 和 name。由于本文的人机物融合应用流程设计方案面向对象覆盖普通用户，所有连接模型上不用用户配置参数或者表达式，Waiting 控制模型是复合模型，包含 Flow 连接模型，其语义为轮询停等，所有我们规定本模型前序模型必须是查询类活动模型。本文设定轮询返回的连接模型判定表达式为“\${QueryState==false}”，通过连接下一活动 连接模型判定表达式为“\${QueryState==true}”。另外同样对应还包括<bpmndi:BPMNPlane>子元素下的和模型对应的<BPMNShape>元素描述。

FlowAttribute 描述的是连接模型对应的具体的 XML 元素：<process>子元素下的<sequenceFlow>元素，只有在含有 Branch 结构和 Waiting 结构时才包含<conditionExpression>隐式规则设定好的连接模型的数据表达式，如上文中提到的连接模型表达式。同样，还包括<bpmndi:BPMNPlane>子元素下的和模型对应的<BPMNShape>元素描述。

将用户流程模型映射为 BPMN 的 xml 描述，在 UserProcess 描述用户流程模型时添加<process>元素并最终在 Canvas 的 xml 可视化视图中展示用户模型映射得到的 BPMN 的 xml 描述，支持用户在建模过程中实时进行模型映射并可视化。

#### 4.3.4 用户模型校验

用户模型校验是在用户流程定义结束后由模型校验器进行的模型有效性校验。该过程的输入是用户模型映射后的 BPMN 的 xml 描述，输出是校验结果信息，校验结果信息也将反馈用户，支持提示用户重定义。用户模型校验器的作用在于：一、规避用户因个人流程定义的失误、错误操作和恶意修改 XML 描述造成的流程无法解析执行甚至具有语法和语义错误的流程进入运行态的隐患，可以提高系统的可用性、安全性和容错性；二、将校验提示信息显示在可视化区域能

够提示和指导用户重定义，可以提高系统的用户满意度和降低操作难度。

本文用户模型校验阶段的模型校验算法从用户流程模型粒度上划分，需要依次经历单个模型的检验，模型间的校验和整体流程模型的校验三个步骤。单个模型的检验负责校验单个模型的定义和属性描述的合法性校验，特别是在本工具符合人机物特征所必须的属性检验。模型间的校验校验的是在本工具中人机物特征下存在的模型顺序性关联关系的有效性。整体流程模型的校验则是对用户个性化定制设计的整个流程模型进行的完整性校验。下面介绍该算法的具体实现过程。

首先，将模型映射结果即 `canvs` 下的 `BPMN xml`，获取部署 `xml` 输入流后将其解析为 `BPMNModel`，创建模型校验工厂并分别获取和注册上述类型三种模型校验器。然后分别实现三种校验器进行模型校验。

单模型校验器校验该模型是否合法定义和构建，基于 `BPMNModel` 模型校验器对模型元素标签进行校验以及对用户模型元素的属性进行校验。3.2.4 中所介绍了模型映射分别具有多个属性，在单模型校验器重需要逐一进行存在性和合法性校验。以活动模型为例，这里我们截取了对单个活动模型中的 `class` 委托类属性的校验过程的关键代码如 4-1 所示。其他模型的元素标签和属性和活动模型校验过程示例校验类似，该模型校验器不仅校验了模型元素的元素信息，还需要校验元素的属性信息，最后并将检验结果加入校验结果错误信息集。

代码 4-1 单模型校验

```
protected void executeValidation(BpmnModel bpmnModel, Process process,
List<ValidationError> errors) { //校验活动模型委托类属性
    List<ServiceTask> serviceTasks =
process.findFlowElementsOfType(ServiceTask.class);
    for(ServiceTask serviceTask:serviceTasks){
        String delegter = serviceTask.getImplementation();
        if(delegter == null)
        {
            ValidationError validationError = new ValidationError();
            validationError.setProblem(serviceTask.getId()+"委托类未设置");
            errors.add(validationError);
        }else if(!delegter.equals("XXX.JavaDelegate")){ //统一委托类名
            ValidationError = new ValidationError();
            validationError.setProblem(serviceTask.getId()+"委托类设置不规范");
            errors.add(validationError);
        }
    }
}
```

模型间校验器用来校验相邻关联模型之间的语义关联关系，最直接的校验规则就是关联模型的先后顺序关系。一般而言，具有关联关系的模型之间是没有强顺序性约束的，但鉴于人机物特征和面向对象覆盖普通用户，具有了一定的规则约束，比如在我们设计的工具中自定义了复合控制结构 `Waiting` 模型，以及为了

覆盖普通用户而取消了用户菜单配置属性参数的辅助操作方式,我们需要对部分流程模型之间的关联关系进行顺序性的约束。**Waiting** 控制结构模型描述了轮询等待语义的自定义控制结构,它是 **Branch** 模型映射对应的 **exclusive** 元素和多个 **Flow** 模型映射的 **sequence** 元素组合表示,约束了 **Waiting** 模型的前一个关联模型必须是查询活动模型,这样二者的条件表达式就可以统一以某个约定的布尔表达式判定范式来描述,对用户隐式的映射存在。以 **Waiting** 控制结构模型为例的模型间检验实现如代码 4-2 所示。

**代码 4-2 模型间校验**

```
protected void executeValidation(BpmnModel bpmnModel, Process process,
List<ValidationError> errors) { //Waiting 相关联的模型间校验
    List<ExclusiveGateway> waitingGateway =
        process.findFlowElementsOfType(ExclusiveGateway.class);
    for(ExclusiveGateway exclusiveGateway:waitingGateway){
        ValidationError validationError = new ValidationError();
        int index = 0, FlowElement pre = null, FlowElement next = null;
        for(FlowElement
flowElement:process.findParent(exclusiveGateway).getFlowElements()){
            if(index == 0) {
                pre = next = flowElement; index++;
            }else{
                pre = next; next = flowElement; index++;
            }
            if(index != 0 && next.getId().contains("Waiting")
&& !pre.getId().contains("Query")){
                validationError.setProblem(exclusiveGateway.getId()+"
Waiting 模型前未关联查询活动模型, 不合规范!");
                errors.add(validationError);
            }
        }
    }
}
```

整体流程模型的校验则是对用户流程个性化定制设计的流程模型 **UserProcess** 进行校验,旨在对用户流程进行完整性校验。用户设计的流程必须包括一个活动开始状态和一个活动结束状态,并且流程完整性的定义是自开始到结束状态的路径可达性,假若其中一条路径不可达则说明该流程定义不完整,不能被直接加载到运行时环境调度执行,其校验实现核心代码如代码 4-3 所示。

**代码 4-3 流程模型校验**

```
protected void executeValidation(BpmnModel bpmnModel, Process process,
List<ValidationError> errors) {
    ...//开始、结束状态前置校验处理; FlowElement 和 Flow 模型分别初始化处理
    Map<String,String> map = new HashMap<>(); //Map 表示邻接关联关系, 初始化 map
```

---

```

    for (FlowElement flowElement: process.getFlowElements()) {
        // 该方法获取到的 FlowElement 带有前后关联模型信息, map 更新构建流程模型图结构。
    }
    List<String> error = new LinkedList<>();
    for (FlowElement flowElement: node) { // 遍历所有组成流程的 FlowElement 模型
        String id = flowElement.getId();
        while (!id.contains("end") && map.containsKey(id)) { // End 模型为结束标志
            id = map.get(id);
        }
        if (error.size() != 0) {
            for (String s: error) {
                ValidationError validationError = new ValidationError();
                validationError1.setProblem("模型" + s + "不可达, 请认真审核!");
                errors.add(validationError1);
            }
        }
    }
}

```

---

整体流程模型校验实现部分首先需要对开始、结束状态进行前置校验处理; 然后需要对 FlowElement 和 Flow 模型分别进行初始化处理, 获取全部流程模型。接着采用了 HashMap 结构表示模型的邻接矩阵来描述模型之间的关联关系, 建立从 Start 状态模型到 End 状态模型之间的图结构关联关系。接下来我们自 Start 状态模型出发, 一直到 End 正序回溯遍历所有路径, 并在不可达路径的最尾端的模型位置将流程不完整信息加入错误信息集。

至此, 从模型到模型间再到全模型组成的流程模型校验过程全部结束。最后, 我们需要根据错误信息集来判断是否从用户流程设计进入系统资源调度执行状态。我们通过判断错误信息集的集合数量是否为 0 即可判定用户模型校验是否通过。若错误信息集的集合数量是否为 0 即用户模型校验成功通过则将进入运行时环境执行, 否则将错误信息集内的信息反馈给用户显示在工作区, 并提示和指导用户进行修改和重定义。

## 4.4 运行时环境构建

运行时环境是支持人机物融合应用流程执行核心组件, 基于执行引擎支持人机物资源调度执行来完成用户活动。它的输入是用户流程设计工具输出的用户流程模型, 在运行时环境中依赖人机物资源仓库的关联方法和动态服务绑定策略来调度相应的人机物资源。具体来说, 运行时环境面向的对象是人机物实例服务, 进一步通过人机物资源语义仓库由服务来直接关联人机物资源, 调度资源的执行。运行时环境负责两个方面的功能: 一、人机物流程活动的自动化执行; 二、人机物资源的自适应调度执行。人机物流程活动自动化执行负责将用户设计的流程活

动自动化的调用相应服务来完成场景流程需求,人机物资源自适应执行负责在运行时动态地绑定一个最优服务来调度人机物资源完成流程活动。两者面向的范围不同,自动化执行针对的是整个流程活动的自动化,自适应针对的是流程中某个活动由某个人机物资源自适应调度执行完成。

#### 4.4.1 人机物应用流程自动化执行

流程活动自动化的执行依赖于 Activiti 工作流执行引擎,我们对其进行了扩展来满足人机物融合场景下的服务自动化。本文对该执行引擎的扩展主要集中在以下三个方面:一、统一了用户活动委托类,该委托类启动服务自适应执行过程;二、流程节点的异步执行实现,具体体现在人机物服务的是否支持回调的挂起执行需求上,该部分内容也会包含在接下来对整个流程活动自动化执行具体实现的描述内;三、自定义 Waiting 停等结构控制流程节点的解析执行。

首先,运行时环境输入经校验通过的用户流程定义映射后的 BPMN 的 xml 描述,我们需要对 xml 描述进行流程的部署和执行,这里采用了流部署的方式方便将用户流程定义工具中直接获取 xml 字符串描述直接进行部署和执行,该过程就是实例化一个工作流实例,开始启动执行流程活动,核心代码代码 4-4 所示。

代码 4-4 执行引擎部署应用流程

```
ProcessEngineServices.repositoryService.createDeployment(). //流部署方式
    addString(processKey+".bpmn", bpmn).deploy();

new Thread() {
    public void run() {
        ProcessInstance pi = ProcessEngineServices.runtimeService.
            startProcessInstanceByKey(processKey, variables);

        String processInstanceId = pi.getId();

        ProcessManagement.continueProcess(processInstanceId);
    }
}.start();
```

设计上我们将活动模型对应的节点调用类统一委托给了同一个执行类,即上一节单模型校验中的 class 属性值为"xxx.JavaDelegate"的委托类。该委托类作为流程活动自适应调度执行的统一入口,负责用户活动的具体执行过程,该部分将在下一节内容描述。

在 3.2.4 一节中介绍了模型的相关属性,其中活动模型包含了 exclusive、async 属性,用来实现流程活动的异步。实现上我们的设计是所有的活动模型默认都将这两个属性设置为 true,代表的含义是该活动模型及之前的所有活动模型作为一个完整的事务将被挂起,只有获得执行信号时才继续执行,这样实现了流程活动异步回调的等待功能。活动模型的挂起和执行实现的关键代码如 4-5 所示。

代码 4-5 节点挂起和继续执行



---

```
...//从历史节点信息获取最近节点,判断应用流程是否已经结束,若未结束将继续向下执行
Job job = ProcessEngineServices.managementService.createJobQuery().
    processInstanceId(processInstanceId).singleResult();
ProcessEngineServices.managementService.executeJob(job.getId());
...//流程若结束返回结束时间
```

---

通过构造 Job 执行器从历史执行模型中获取最近的流程模型,判断流程活动是否结束,如果未结束则继续推进执行。除了开始状态模型直接调用调用该模块实现直接执行,其余活动模型都需要根据其动态绑定的人机物服务的类型(是否支持回调)来决定是否直接调用,若支持异步回调,则停下来等待直到获得异步完成信号调用由该部分封装实现的 rest 接口执行继续执行下一个模型;否则,直接调用该模块代码,不需要等待回调信号直接完成整个活动的执行,继续继续执行下一个关联模型。受限于文章篇幅,我们不具体展开其调用过程。

#### 4.4.2 人机物应用流程自适应执行

人机物资源自适应过程是由执行引擎委托类统一代理,调用的具体人机物服务是服务自适应模块完成服务的动态绑定返回的一个最优服务,基于人机物资源仓库关联人机物资源实例,调度人机物资源完成用户活动。人机物应用自适应依据当前空间下的实际环境基于服务动态绑定策略来匹配最优的人机物服务,完成特定的用户活动。本文设计的服务自适应的动态绑定策略采用了三个维度来对进行服务最优匹配,即人机物实例资源可用性、人机物实例资源的空间位置关系和人机物实例资源使用次数代表的用户习惯。上述三个维度下所自适应匹配到的唯一服务就是最终返回的最优人机物服务,每个维度匹配到唯一实例服务即可作为最优服务直接返回,并提交给执行引擎活动委托类调用执行,最终即可调度相关联的人机物资源完成该活动。人机物应用流程自适应执行是在人机物应用流程执行时进行的资源服务的动态绑定,进而调度提供该服务的人机物资源完成用户活动,满足了运行时实时环境下的应用自适应执行的需求。

人机物资源自适应是人机物服务自动化执行每个活动进行人机物服务调度人机物资源执行过程中进行服务动态绑定的一个子过程问题求解方案,执行引擎直接关联的是活动模型对应的统一委托类的人机物服务调用,从委托类延伸到具体人机物服务则是由服务自适应算法求解。本文的服务动态绑定算法描述从上述三个维度分别进行匹配与绑定,下面相应也将从三个维度的自适应匹配过程来描述该资源自适应调度算法。

首先,执行引擎解析执行活动模型描述的活动,委托类作为调用监听类获取模型属性 id,委托类请求服务自适应处理器,处理器基于人机物资源仓库的关联方法将模型属性 id 直接关联人机物服务能力特征模型,通过该一对一的映射关

系获取唯一的服务能力特征模型。服务能力特征模型是由若干人机物品资源实例具有的相同服务抽象而成，即一个服务特征模型关联了若干实例服务模型，进而通过该关联方法可以获取得到所有的人机物服务实例候选集。最后，通过实例服务可以关联的人机物实例资源模型获取实例资源 id，调用相应的属性状态查询判断可用性，初步获取到可用的人机物实例服务候选集，实现过程如下公式所示。公式(4.4) (4.5) (4.6)分别描述了用户活动模型集、服务能力特征模型集和服务实例模型集，(4.7)最终获取得到的 $Candidate_{avl}$ 是人机物资源可用状态下的人机物实例服务候选集合。

$$Model_{engine} = \langle task_1, task_2, task_3 \dots task_n \rangle \quad (4.4)$$

$$Model_{ablity} = \langle ablity_1, ablity_2, ablity_3 \dots ablity_m \rangle \quad (4.5)$$

$$Entity_{Service} = \langle service_1, service_2, service_3 \dots service_k \rangle \quad (4.6)$$

$$Candidate_{avl} = \left\{ \begin{array}{l} s \in Model_{engine} \cap Model_{ablity} \cap Entity_{Service} \\ |s \text{ is available} \end{array} \right\} \quad (4.7)$$

接着，根据上一输出候选集进行空间位置关系的自适应匹配过程。基于资源仓库最终关联方法，将人机物服务实例模型和空间人机物资源关联模型进行关联，遍历上一阶段可用的人机物服务实例候选集，获取到提供服务的人机物实例资源所在空间，以 Room 为单位。我们借助 UWB 技术采用三角定位算法对用户和实例资源进行空间下的初始化位置定位，获取了人和物资源的位置。遍历所有可用的人机物实例服务候选集，借助资源语义仓库的空间人机物资源关系模型关联方法判定有无和用户所在空间在一个 Room 区域下的人机物实例资源，默认所有和用户所在同一空间下的资源和用户距离最近且近似相等，将其在候选集中的人机物实例服务加入空间距离维度下的候选集；如果没有，通过将对应实例服务的人机物资源和用户进行欧式距离计算获取最近的人机物实例资源，并将其在上一阶段候选集中关联的人机物实例服务加入候选集。

最后，本系统用户的每次服务资源的调用都将反馈到系统，即将涉及的所有服务资源调用次数进行累加计数，并作为仓库中服务资源的属性持久化记录。本文以这种方式来近似估计用户的偏好和使用习惯。该阶段会遍历上一阶段输出的候选集服务，统计其最大用户使用次数来作为衡量用户使用习惯的标准，最终返回用户最常使用的实例服务作为最优服务。

经过以上三个维度的服务最优匹配过程，用户个性化定制的流程活动最终直接绑定上了一个最优的人机物服务，然后交由委托类调用执行，即人机物场景流程自动化中单个活动的服务动态绑定过程完成。最后基于人机物资源仓库中人机物实例资源模型和人机物实例服务模型的关联关系，该人机物实例服务将直接关联到某个人机物实例资源，执行引擎将调度该资源来自动化完成用户活动，实现了系统运行时环境下基于实时环境的用户流程自动化执行和人机物资源自适应调度过程。

## 4.5 本章小结

本章详细描述了人机物融合应用工具的各组件模块构建的方法和算法具体实现。首先介绍了人机物融合应用工具的系统架构，接着从人机资源语义仓库、用户流程设计工具和运行时环境三个核心组件的设计和实现展开详细的分析和介绍。本章主要是对系统实现上的技术方法和核心算法进行了描述，各模块部分的可视化结果将在下一章实验部分展示。在下一章内容安排上，我们将基于该工具进行场景实例的多方面实验进行验证和分析。



## 第五章 场景实例实验与分析

本章的主要内容是介绍和分析本文设计实现的系统工具实验结果，第一部分介绍实验工具和实验过程，然后先从面向用户的角度介绍和分析工具使用的性能结果，再基于案例分析和验证系统工具的各模块实验结果。

### 5.1 实验工具

本节将主要介绍本文设计实现的系统工具和实验环境。本实验基于该系统工具，在实验所覆盖的一个两个房间的实验环境下进行人机物融合应用场景实验，实验对象除了系统工具，还包括实验环境下的所有关联设备和人。

如图 5-1 展示了用户流程设计工具的使用界面，它是一个 Web 网页的用户活动流程定义工具，通过该工具用户可以个性化的通过拖拽的形式组合场景流程活动模型完成场景活动的描述和定义，然后点击菜单栏上的模型检查菜单来对用户的流程定义进行校验，不通过则需要按照提示修改，通过校验则流程活动将在经由系统运行时环境调用相应的资源完成活动。

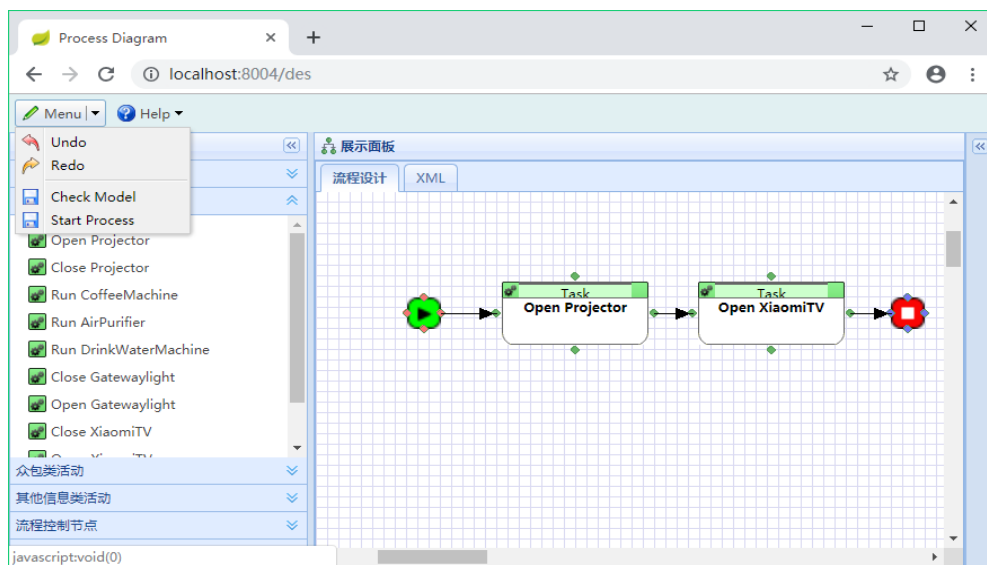


图 5-1 系统的用户活动流程定义工具

### 5.2 性能评估

本节从用户的角度对一系列实验进行系统工具的效率和易用性两个方面的评估和分析。工具的效率和易用性分别对应回答下面研究问题：

- 效率：用户通过该工具完成某流程活动需要花耗的时间可以确保比用户直接负责完成所有的流程活动耗费的时间更短，效率更高吗？

- 易用性：用户支持使用该系统工具来满足现实生活下的场景活动吗？用户对该工具的使用满意度怎么样？

### 5.2.1 效率

本文设计实现的系统对于用户来说只有流程定义工具是可见的，所以用户完成一个自定义活动的所要耗费的时间在于从用户打开活动定义工具和校验通过、发布执行的花耗的过程时间。用户直接负责相同的所有活动则需要从用户开始行动到完成最后一个活动的所有花耗时间。

为了回答效率性能所描述的问题，我们邀请了十名包含技术人员和非技术人员的同学分别采用本文工具来完成自定义的一项场景活动和直接负责该场景活动的所有活动的两个对比试验。这项场景活动顺序包含以下六个活动：启动投影仪，启动空气净化器，启动咖啡机，启动热水器，打开灯（网关灯代替）和接送咖啡。本实验忽略了咖啡机煮咖啡的等待时间，使用该工具用户和协作者异步协作，不存在该时间花费。而直接负责该活动因为是串行活动，会增加时间消耗。，两种方式的活动数量和活动完成的平均耗费时间关系如下图 5-2 所示。

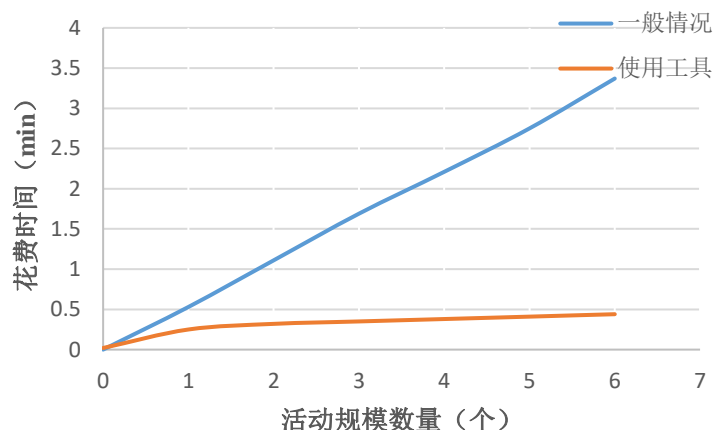


图 5-2 活动规模数量和花费时间的关系

可以看出，当完成的活动数量不断增加的时候，通过本工具（下方的红色线）的耗费时间几乎不会有很大增加，而用户直接完成（上方的蓝色线）所有活动的时间则增加明显。由于使用本工具的耗费时间主要是活动模型组合的流程在设计过程和模型校验过程。发布执行过程过程可忽略不计，活动模型组合过程花耗的是用户拖拽过程拖拽组合时间，活动数量增加对用户时间消耗没有明显增加，而模型校验过程的时间几乎是相同的。而用户直接完成活动则因活动增加不断往返移动移动操作，以及完成活动中因对活动对象操控不熟悉而耽误时间。

从整个结果可以看出，本系统工具在用户使用的效率性能上是很具有优势的，可以显著提高用户的时间效率。

## 5.2.2 易用性

为了回答易用性性能描述的问题，我们对上述十名同学发送了用户工具使用的满意度调查，我们提到该工具的涉及到的“人”资源即实验参与者分为活动发起用户、活动委托用户两种角色，分别从以下几个方面进行了满意度评价：

Q1（活动发起用户）：工具活动设计可视化界面有效，并且能够轻易理解左边导航栏活动建模图元代表的单个活动；

Q2（活动发起用户）：工具的操作简单，效率高，平时生活场景愿意尝试该工具完成活动；

Q3（活动委托用户）：收到众包活动委托请求通知，能够轻松理解活动任务内容并且接受和完成操作易于操作。

Q4（活动委托用户）：平时活动委托请求通知，乐意帮助完成委托活动任务。

所有实验参与者将给出满意度从 1 分到五分代表完全不满意和完全满意的梯度评分，后续我们对参与者进行了评分的反馈意见收集，用来补充数据表达上的内容缺失，满意度结果如下表 5-2 所示。

表 5-2 用户满意度评估

	Q1			Q2			Q3			Q4		
	M	$\bar{x}$	$\sigma^2$	M	$\bar{x}$	$\sigma^2$	M	$\bar{x}$	$\sigma^2$	M	$\bar{x}$	$\sigma^2$
满意度	5	4.7	0.23	4	4.2	0.40	4	4.0	0.22	3	3.4	0.26

从上表参与者对工具使用的满意度评分可以看出，活动发起用户对工具的简单可用性（Q1）评分最高，活动委托者积极参与度的意愿度相对低一点（Q4），反映了该工具的简单可操作性还是比较可靠的，但是活动委托者的参与意愿不够高。通过对参与者的反馈记录统计该工具的操作方面可以工具界面和活动描述改为中文描述，加入活动定义可复用和修改功能等等改进意见。另外，活动委托者的参与度的提升可以设定一系列的激励机制，比如积分兑换或者基于积分的使用权等。总体来看，该工具还是被用户认可，可以认为具有较强的易用性。

## 5.3 案例分析

本文以“智能会议室”场景为例，该场景来源于真实生活场景。我们通过对该场景案例的实验，从系统整体的角度来验证本文工具各模块的有效性和可用性。

### 5.3.1 场景实例背景

本文实验环境下实验室的会议室中购置了一系列智能设备或者电子设备，在

本文场景实例中涉及的智能设备包括投影仪一个，空气净化器一台，饮水机一台，咖啡机一台，由于设备资源有限，这里我们将可以提供烧热水功能的咖啡机模拟成另外一台饮水机。在该实验环境下，我们观察到存在一个比较迫切的场景需求能够阐述本文工具的优势。

该场景实例需求描述如下：会议组织者发起了一次小组会议，召集小组成员在会议室召开会议。为了让小组成员能够在紧张的讨论中得到放松，希望利用会议室和空间内的资源提供相关舒适的会议室环境。平时的活动流程是先打开投影仪，打开空气净化器，会议短暂休息中开饮水机加热水，把热水接到会议室。由于某些智能设备的使用需要使用经验才能熟练使用，比如咖啡机、投影仪，经常会出现叫人完成这些活动，并且在会议中需要自己或者安排与会者去取是一件糟糕的事，与我们目的提供一个舒适的环境背道而驰。

基于本文设计的工具，我们可以为上述场景需求提供解决方法。会议组织者只需要使用本文工具拖拽图元模型定义上述场景活动流程，校验成功后发布流程活动后即可继续准备开会。所有活动的执行将由系统自动调用相关的资源来完成，会议组织者和参会者完全不需要关心，其场景活动时序图如 5-3 所示。系统运行时环境自动化完成的活动执行顺序为打开投影仪，投影仪打开后打开空气净化器，开饮水机烧水，发送众包服务请求非与会人员将开水送至会议室。

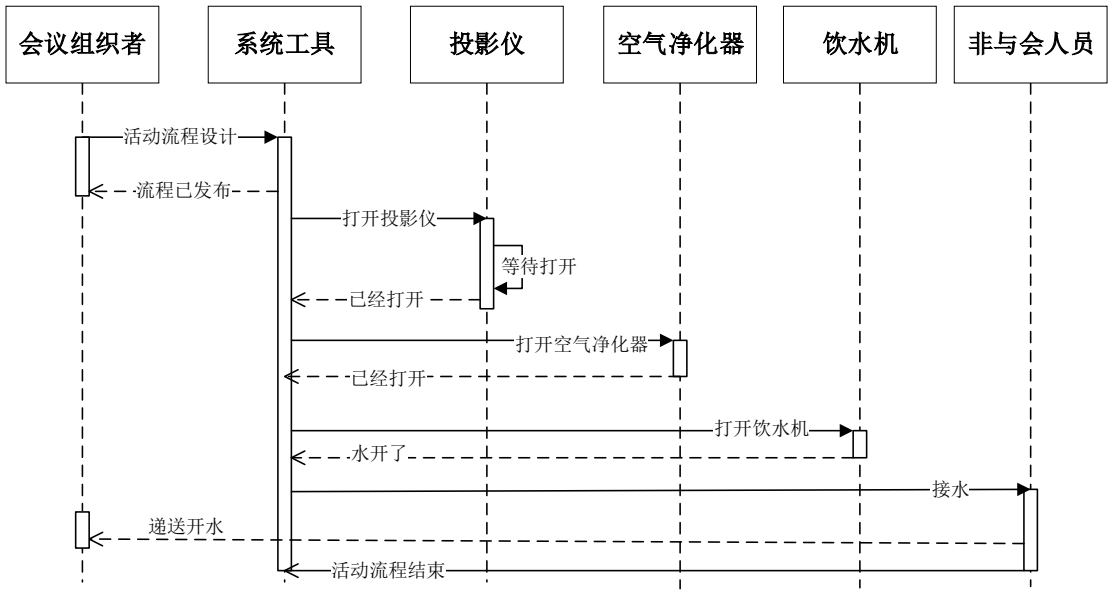


图 5-3 流程活动时序图

5.3.2 结果与分析

基于场景该案例，本节将对系统工具的各模块进行实验结果的验证与分析。

（一）会议组织者流程定义结果与分析

会议组织者通过拖拽相关活动图元模型后完成该场景活动的流程定义。该场景流程活动定义最终结果如下图 5-4 所示。

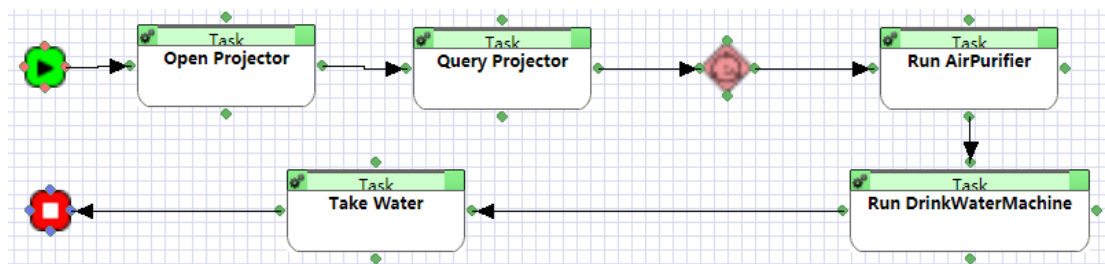


图 5-4 场景活动流程定义

该场景流程活动定义中，首先打开投影仪，由于投影仪打开延迟时间较长，需要轮询判断，使用打开投影仪后添加了状态查询操作，并使用 Waiting 控制节点示意代表轮询等待，投影仪开启后打开空气净化器，启动饮水机热水加热，最后是热水帮送众包请求，“Take Water”为发接开水众包请求，接开水众包完成后才能进入下一个活动即结束，该流程定义执行的活动和场景需求描述一致。

## （二）模型映射结果与分析

场景流程定义结束完成后，用户活动流程将被映射为 BPMN 模型的 xml 描述，可以在用户流程定义工具的 xml 可视化工作区中展示，如图 5-5 所示，由于该 xml 描述较长，我们主要截取了其中的核心部分，即流程模型属性的定义和关联部分的部分结果。在该 xml 描述中，我们可以看到每个用户活动组成模型和关联关系都能够反映在该 xml 描述，并支持运行时环境的解析和自动化执行过程。

```
<process id="process1567752279666" name="process1567752279666">
  <startEvent id="start" name="Start" activiti: initiator="_wf_default_v_applicant">
  </startEvent>
  <endEvent id="end" name="End" activiti: async="true">
  </endEvent>
  <serviceTask id="Open_Projector" name="Open_Projector"
  activiti: class="cn.edu.fudan.selab.hct_activiti.engine.ActivitiJavaDelegate" activiti: exclusive="true" activiti: async="true">
  </serviceTask>
  <serviceTask id="Query_Projector" name="Query_Projector"
  activiti: class="cn.edu.fudan.selab.hct_activiti.engine.ActivitiJavaDelegate" activiti: exclusive="true" activiti: async="true">
  </serviceTask>
  <exclusiveGateway id="Waiting1567769821097" name="Waiting1567769821097">
  </exclusiveGateway>
  <serviceTask id="Run_AirPurifier" name="Run_AirPurifier"
  activiti: class="cn.edu.fudan.selab.hct_activiti.engine.ActivitiJavaDelegate" activiti: exclusive="true" activiti: async="true">
  </serviceTask>
  <serviceTask id="Run_DrinkWaterMachine" name="Run_DrinkWaterMachine"
  activiti: class="cn.edu.fudan.selab.hct_activiti.engine.ActivitiJavaDelegate" activiti: exclusive="true" activiti: async="true">
  </serviceTask>
  <serviceTask id="Take_Water" name="Take_Water" activiti: class="cn.edu.fudan.selab.hct_activiti.engine.ActivitiJavaDelegate"
  activiti: exclusive="true" activiti: async="true">
  </serviceTask>
  <sequenceFlow id="flow1567769871289" name="flow1567769871289" sourceRef="start" targetRef="Open_Projector">
  </sequenceFlow>
  <sequenceFlow id="flow1567769872581" name="flow1567769872581" sourceRef="Open_Projector" targetRef="Query_Projector">
  </sequenceFlow>
  <sequenceFlow id="flow1567769873816" name="flow1567769873816" sourceRef="Query_Projector" targetRef="Waiting1567769821097">
  </sequenceFlow>
  <sequenceFlow id="flow1567769873816_back" name="flow1567769873816_back" sourceRef="Waiting1567769821097"
  targetRef="Query_Projector">
    <conditionExpression xsi:type="tFormalExpression"> ${QueryState == false} </conditionExpression>
  </sequenceFlow>
  <sequenceFlow id="flow1567769875573" name="flow1567769875573" sourceRef="Waiting1567769821097" targetRef="Run_AirPurifier">
    <conditionExpression xsi:type="tFormalExpression"> ${QueryState == true} </conditionExpression>
  </sequenceFlow>
  <sequenceFlow id="flow1567769877095" name="flow1567769877095" sourceRef="Run_AirPurifier" targetRef="Run_DrinkWaterMachine">
  </sequenceFlow>
  <sequenceFlow id="flow1567769878867" name="flow1567769878867" sourceRef="Run_DrinkWaterMachine" targetRef="Take_Water">
  </sequenceFlow>
  <sequenceFlow id="flow1567769880749" name="flow1567769880749" sourceRef="Take_Water" targetRef="end">
  </sequenceFlow>
</process>
```

图 5-5 用户流程定义映射成 BPMN 的 xml 文件

### （三）模型校验结果与分析

在进行实例校验结果分析前，我们模拟测试了几种典型的模型校验示例，实验结果如下图 5-6 所示。其中（a）（b）图分别代表了用户模型模型间的校验后错误信息提示和流程模型整体的校验后错误信息提示。（c）图表示了上述描述的活动流程定义的校验结果。值得关注的是，在这项模型校验实验结果中并没有出现单个模型的属性信息校验，这里本系统工具对用户屏蔽了模型映射规则和任何的属性自定义操作，以实现尽可能地覆盖普通用户。所以，对于单个模型的校验主要用来规避用户恶意修改映射后的 xml 描述的带来的安全性问题，一般情况下不会出现单个模型校验的错误信息。

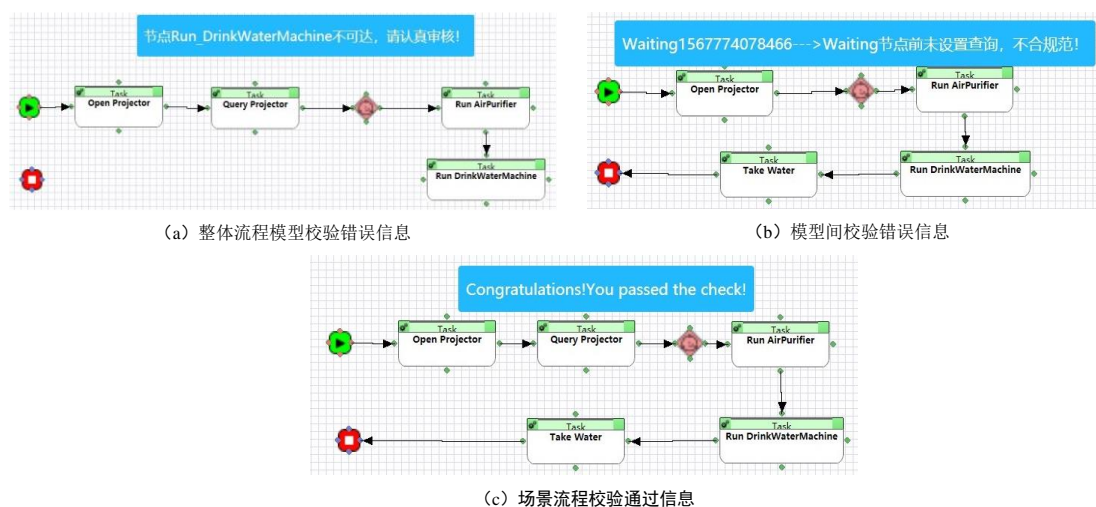


图 5-6 模型校验信息

图（a）描述的是基于整体流程校验检测出在该场景实例下整个流程活动不完整，可视化区域提示了某活动节点存在路径不可达信息，图（b）则描述了基于模型间的校验检测出 Waiting 节点前不存在查询类活动，该模型间关系本文设计实现中自定义 Waiting 节点规定的来实现轮询停等。图（c）则描述的是在实验场景案例中有效流程活动定义通过模型校验器的提示信息。通过一系列的模型校验器校验实验，验证了该模块模型校验的可靠性和准确性。

### （四）运行时环境自动化和自适应的结果与分析

系统运行时环境下的流程活动自动化执行日志结果如下图 5-7 所示，对应上述活动步骤的完成产生了对应日志结果。通过该日志和对实际环境下的各类资源调度情况的验证，该系统可以自动化顺序完成会议组织者所定义的流程活动需求。

```

INFO 9560 --- [nio-8003-exec-1] o.a.engine.impl.ProcessEngineImpl : ProcessEngine default created
INFO 9560 --- [nio-8003-exec-1] org.activiti.engine.ProcessEngines : initialised process engine default
INFO 9560 --- [nio-8003-exec-1] o.a.e.impl.bpmn.deployer.BpmnDeployer : Processing resource process1567781322865.bpmn
INFO 9560 --- [ Thread-19] c.e.f.s.h.ontology.PostMessage : 正在进行:Open_Projector活动
INFO 9560 --- [ Thread-19] c.e.f.s.h.ontology.PostMessage : 正在进行:Query_Projector活动,等待设备就绪
INFO 9560 --- [ Thread-19] c.e.f.s.h.ontology.PostMessage : 正在进行:Run_AirPurifier活动
INFO 9560 --- [ Thread-19] c.e.f.s.h.ontology.PostMessage : 正在进行:Run_DrinkWaterMachine活动
INFO 9560 --- [ Thread-19] c.e.f.s.h.ontology.PostMessage : Take_Water众包活动已发布,请耐心等待接包人送达
INFO 9560 --- [ Thread-19] c.e.f.s.h.ontology.PostMessage : 已送达, 流程活动全部结束!

```

图 5-7 运行时环境自动化执行日志结果



本文实验中包含两台饮水机 T1 和 T2（T2 以咖啡机模拟代替），场景流程发起者位于会议室 Room1 区域，隔壁房间为 Room2 区域。该实验场景中 Room1 和 Room2 紧挨着，同位于一个机构单位所属空间下，在人机物资源仓库的人机物空间实例资源关系模型表示中 T1 和 T2 的父节点分别有同一个祖父节点（单位机构），T1 和 T2 的父节点分别是其祖父节点的两个子节点。该场景实例下利用这两台饮水机，我们进行了人机物资源自适应调度实验，实验结果如下表 5-3 所示。实验中分别对比了动态绑定策略中代表资源可用性、空间位置关系和用户习惯的可用状态、T1 和 T2 位置关系和调用次数三个属性状态和自适应结果的关系。

表 5-3 饮水机调度结果

T1	T2	自适应结果
可用状态、Room2	不可用状态、Room2	T1
不可用状态、Room2	可用状态、Room2	T2
可用状态、Room1	可用状态、Room2	T1
可用状态、Room2 调用次数 176	可用状态、Room2 调用次数 4	T1

从上表中我们可以分析得出，基于现实空间环境，本系统可以有效的调度最优的人机物资源来支持活动的完成，即验证了本系统运行时环境人机物资源自适应调度执行的有效性。

## 5.4 方法的局限性

本文提出的人机物融合应用的工作流设计与执行技术并设计实现的系统工具存在一定的局限性，主要体现在以下几个方面：

（1）本文提出的人机物融合应用的工作流设计与执行技术虽然支持用户流程定义工具动态刷新，从人机物资源仓库映射服务能力特征模型，动态生成可拖拽活动描述图元模型，但代表“物”的设备或者代表“机”的信息服务没有实现动态发现并进行服务抽取和抽象的全过程自动化，即用户场景的设计必须基于服务仓库中以构建存在的模型。

（2）由于资源仓库能力特征模型和活动模型构建采用的“动宾”结构关系指定了同类型设备名称，本文所提出的技术方法仅支持同类型设备间的服务动态绑定过程的自适应，尚未覆盖提供相同能力的不同类型设备间的对应服务的服务动态绑定。

（3）本文所提出的技术方法建立在人机物中物理资源通信可信的基础上，没有考虑空间下网络波动或者掉线的情况，即没有对已接受执行指定的服务进行执行后验检查。

## 5.5 本章小结

本章我们通过本文设计实现的工具进行了一系列实验并对实验结果进行了展示和分析。首先对系统工具及使用进行了介绍，从面向用户的角度对系统工具进行了性能评估实验，并从工具的效率和易用性进行了性能分析。然后基于“智能会议室”的实验案例验证和分析了该系统的可用性和有效性，实验证明本文设计并实现的工具可以高效的完成用户场景活动，用户对工具实验也较为满意。最后基于上述是研究结果的分析，对本文的研究方法存在的局限性进行了描述。



## 第六章 总结与展望

### 6.1 工作总结

人机物融合作为物联网快速发展衍生出的新的研究热点，是一种物联网领域下的社会物理信息系统（CPSS）。与传统的应用构造方式不同，人机物融合应用将分别存在于社会空间、信息空间和物理空间的三元要素进行融合打造成了一个以人为中心的“智能空间”。现有的对人机物融合的研究存在诸多缺陷，研究角度还停留在“以技术为中心”角度上，人机物应用大多是基于“ad-hoc”形式的固化的资源联动模式，无法满足用户即时的场景活动个性化定制的需求。基于这些存在的缺陷和不足，本文以日常生活中的流程类生活场景为研究对象设计并实现了人机物融合应用的工作流设计与执行技术方法。基于该设计方案实现的系统工具，用户只需要通过简单地拖拽方式个性化的定制即时场景活动流程，系统则依据实时空间环境自动化和自适应调度相应人机物资源来支持用户活动的完成。

回顾本文主要内容，第一章主要介绍了本文的研究背景、研究目的和意义以及主要工作和创新。本文的主要研究目的在于提出一种针对流程类的日常生活场景，基于用户活动语义的，从简单可视化的场景活动流程个性化定制设计到人机物资源自动化和自适应调度的技术方法。

第二章主要介绍本文研究的相关工作和技术背景，从人机物融合技术、 workflow 技术和面向用户的服务组合技术三个方面分别对现有的技术研究和工作进行了简单介绍。

第三章主要对人机物场景案例进行了分析和对方案的方法设计进行了描述。首先是一个对人机物场景案例简单介绍，提出了技术方法的设计需求，并对现有技术的挑战和不足进行了描述的方案求解思路过程描述，然后从设计方案的研究框架到核心方法设计的描述，包括人机物语义服务仓库的构建方法、用户流程定义工具的建模方法、用户模型映射和校验方法以及自动化与自适应执行方法的设计思路介绍。

第四章主要详细介绍了基于上述方法设计对系统工具的详细设计与实现过程。首先描述了该系统工具的架构，然后人机物资源仓库构建、用户交互的流程设计工具构建以及运行时环境构建三个核心组件进行详细的设计实现描述。

最后，在第五章中本文分别从用户对系统工具使用的角度和系统可用性的角度进行了性能评估实验和基于案例的系统各模块组件结果验证与分析。实验证明，该工具得到用户的满意和认可，能有效完成人机物资源的自动化与自适应调度。

## 6.2 工作展望

在 5.4 节中本文对研究方法存在的局限性进行了介绍，我们未来的研究和工作的将围绕着上述方法局限性的三个方面进行展开和延伸：

（1）本文的资源语义仓库目前还不支持人机物资源到资源仓库模型全自动化生成和映射，在后续的工作中我们需要添加对人机物资源的动态服务发现，并设计基于一定范式和规则的服务语义抽象方式。

（2）本文所提出的技术方案目前仅支持同类型人机物资源的自适应调度执行，在未来的工作中我们需要进一步设计相同服务功能的不同类型人机物资源的自适应调度研究方法。

（3）本文的研究方案是基于各人机物资源的网络通信可靠的条件，没有考虑依据服务资源调度策略调度某个资源后出现异常的情况，在一点上，后期将进行服务资源的调度后验的异常处理策略研究。

## 参考文献

- [1] Kankanhalli A, Charalabidis Y, Mellouli S. IoT and AI for smart government: A research agenda[J]. 2019.
- [2] de Andrade T P C, Astudillo C A, da Fonseca N L S. Allocation of control resources for machine to-machine and human-to-human communications over LTE/LTE-A networks[J]. IEEE Internet of Things Journal, 2016, 3(3): 366-377.
- [3] Purohit L, Kumar S. Web Services in the Internet of Things and Smart Cities: A Case Study on Classification Techniques[J]. IEEE Consumer Electronics Magazine, 2019, 8(2): 39-43.
- [4] Corcoran P. The Internet of Things: why now, and what's next?[J]. IEEE Consumer Electronics Magazine, 2015, 5(1): 63-68.
- [5] Park D. The Quest for the Quality of Things: Can the Internet of Things deliver a promise of the quality of things?[J]. IEEE Consumer Electronics Magazine, 2016, 5(2): 35-37.
- [6] 戴海兵,袁幽然.物联网体系架构及面临的问题和挑战[J].中国新通信,2019,21(9):63. DOI:10.3969/j.issn.1673-4866.2019.09.054.
- [7] Mohanty S P, Choppali U, Kougianos E. Everything you wanted to know about smart cities: The internet of things is the backbone[J]. IEEE Consumer Electronics Magazine, 2016, 5(3): 60-70.
- [8] Wang F Y. The emergence of intelligent enterprises: From CPS to CPSS[J]. IEEE Intelligent Systems, 2010, 25(4): 85-88.
- [9] Lee T, Jeon S, Kang D, et al. Design and implementation of intelligent HVAC system based on IoT and Bigdata platform[C]//2017 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2017: 398-399.
- [10] Zouai M, Kazar O, Bellot G O, et al. Ambiance Intelligence Approach Using IoT and Multi-Agent System[J]. International Journal of Distributed Systems and Technologies (IJDST), 2019, 10(1): 37-55.
- [11] 李勇. 嵌入式 Web 传感器技术研究 [D]. 四川: 西华大学, 2006. DOI:10.7666/d.y906796.
- [12] Vargas D C Y, Salvador C E P. Smart IoT gateway for heterogeneous devices interoperability[J]. IEEE Latin America Transactions, 2016, 14(8): 3900-3906.
- [13] Francese R, Risi M, Tortora G. Iconic languages: Towards end-user programming of mobile applications[J]. Journal of Visual Languages & Computing, 2017, 38: 1-8.

- [14]Han S, Wang X, Zhang J J, et al. Parallel Vehicular Networks: A CPSS-Based Approach via Multimodal Big Data in IoV[J]. IEEE Internet of Things Journal, 2018, 6(1): 1079-1089.
- [15]Li T J J, Azaria A, Myers B A. SUGILITE: creating multimodal smartphone automation by demonstration[C]//Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. ACM, 2017: 6038-6049.
- [16]Bae J, Bae H, Kang S H, et al. Automatic control of workflow processes using ECA rules[J]. IEEE transactions on knowledge and data engineering, 2004, 16(8): 1010-1023.
- [17]Liouane Z, Lemlouma T, Roose P, et al. An improved extreme learning machine model for the prediction of human scenarios in smart homes[J]. Applied Intelligence, 2018, 48(8): 2017-2030.
- [18]Reichherzer T, Satterfield S, Belitsos J, et al. An agent-based architecture for sensor data collection and reasoning in smart home environments for independent living[C]//Canadian Conference on Artificial Intelligence. Springer, Cham, 2016: 15-20.
- [19]Cappiello C, Matera M, Picozzi M. End-user development of mobile mashups[C]//International Conference of Design, User Experience, and Usability. Springer, Berlin, Heidelberg, 2013: 641-650.
- [20]Ma Y, Liu X, Yu M, et al. Mash droid: An approach to mobile-oriented dynamic services discovery and composition by in-app search[C]//2015 IEEE International Conference on Web Services. IEEE, 2015: 725-730.
- [21]Rana J, Morshed S, Synnes K. End-user creation of social apps by utilizing web-based social components and visual app composition[C]//Proceedings of the 22nd International Conference on World Wide Web. ACM, 2013: 1205-1214.
- [22]Smidt H, Thornton M, Ghorbani R. Smart application development for IoT asset management using graph database modeling and high-availability web services[C]//Proceedings of the 51st Hawaii International Conference on System Sciences. 2018.
- [23]Howe J. The rise of crowdsourcing[J]. Wired magazine, 2006, 14(6): 1-4.
- [24]Georgakopoulos D, Hornick M, Sheth A. An overview of workflow management: From process modeling to workflow automation infrastructure[J]. Distributed and parallel Databases, 1995, 3(2): 119-153.
- [25]Rademakers T. Activiti in Action: Executable business processes in BPMN 2.0[M]. Manning Publications Co., 2012.

- [26]Laliwala Z, Mansuri I. Activiti 5. x Business Process Management Beginner's Guide[M]. Packt Publishing Ltd, 2014.
- [27]彭伟, 曹雷, 张永亮, 等. 基于 BPMN2. 0 的指挥决策知识自动化关键技术研究[J]. 第六届中国指挥控制大会论文集 (上册), 2018.
- [28]De S, Barnaghi P, Bauer M, et al. Service modelling for the Internet of Things[C]//2011 Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, 2011: 949-955.
- [29]Rao J, Su X.A survey of automated web service composition methods[C]//International Workshop on Semantic Web Services and Web Process Composition. Springer, Berlin, Heidelberg, 2004: 43-54.
- [30]王先平, 齐逸. 面向服务计算中 Web 服务动态绑定模型研究[J]. 计算机工程与设计, 2010 (19): 4165-4167.
- [31]Ankolekar A, Burstein M, Hobbs J R, et al. DAML-S: Web service description for the semantic web[C]//International Semantic Web Conference. Springer, Berlin, Heidelberg, 2002: 348-363.
- [32]Jaeger M C, Rojec-Goldmann G, Muhl G. Qos aggregation for web service composition using workflow patterns[C]//Proceedings. Eighth IEEE International Enterprise Distributed Object Computing Conference, 2004. EDOC 2004. IEEE, 2004: 149-159.
- [33]Martin D, Burstein M, Hobbs J, et al. OWL-S: Semantic markup for web services[J]. W3C member submission, 2004, 22(4).
- [34]Im J, Kim S, Kim D. IoT mashup as a service: cloud-based mashup service for the Internet of things[C]//2013 IEEE International Conference on Services Computing. IEEE, 2013: 462-469.

## 致 谢

随着毕业论文的定稿完成,看似很漫长实际十分短暂的研究生生涯即将画上一个句号。对我来说,从大学本科升入研究生学习和生活的这两三年里,收获的不仅仅是扩宽了知识、技术上的视野,更重要的是问题解决的全局把控能力以及与同学、老师相处的交流沟通和协作能力,这些都成为我日后踏入社会,适应新的角色和身份的巨大帮助。感慨万千,带着许多不舍与留恋衷心感谢给予我帮助的老师 and 同学。

我要向我的导师沈立炜老师表示感谢,非常感谢沈老师在我研究生的学习和生活期间给与的指导和帮助。沈老师是一位平易近人、十分博学且治学严谨的老师,在两年多的学习生活过程中给予了我很多的帮助。从本科毕业论文的完成到研究生入学的学习安排再到研究方向的引导,沈老师一直在我身后提供着指导和关心。研究生以来,沈老师教会我的不仅仅存在知识和技术上,更在于问题解决方法的影响上。这篇论文从选题到完成,沈老师给予了我对宝贵的指导意见,提出了很多不足,使我受益匪浅,再次表达我的感激之情。

另外,还要感谢彭鑫老师、吴毅坚老师、陈碧欢老师以及赵文耘老师在学习和生活上对我的帮助。实验室各位老师的指导和教诲使我在研究生期间得到了很大的成长。

我也要感谢我的实验室同学和好友,感谢各位同学在两年多时间里学习和成长路上的陪伴,大家相互扶持,共同成长。同时也要感谢实验室的各位学长学姐在学习科研和生活上对我的帮助,从入学至今提供了我许多的帮助和照顾,感谢你们让我在学习和生活上有了许多动力。

最后,我要感谢我的亲人对我的支持与鼓励,是你们无微不至的关怀给予了我人生道路上前进的动力。