

# 物联网服务中间件: 挑战与研究进展

陈海明<sup>1)</sup>, 石海龙<sup>1)</sup>, 李勤<sup>1),2)</sup>, 崔莉<sup>1)</sup>

<sup>1)</sup> (中国科学院计算技术研究所 北京 100190)

<sup>2)</sup> (中国科学院大学 北京 100049)

**摘要** 目前提出的物联网应用系统构建方法主要基于面向服务的体系结构(SOA), 主要原理是将物端资源与云端资源提供的感知、执行与数据处理等能力分别抽象为实体服务和云服务。物联网服务中间件是在实体服务和云服务之上用于支撑物联网任务编程、发现满足任务需求的服务集以及实现各类服务之间互联与互操作的基础结构软件。由于实体服务的异构性、大规模性、动态可用性、情境感知性、时空相关性、资源受限性和执行冲突性, 目前构建物联网服务中间件面临挑战。本文首先给出了物联网服务中间件的功能模型, 并以此为参考, 结合实体服务的特性, 具体指出了设计与实现物联网服务中间件各功能模块所面临的挑战; 接着系统总结了物联网服务中间件各功能模块的实现方法; 然后根据服务描述、服务注册管理、服务发现, 以及服务组合与任务编程接口的研究进展, 对目前已经提出的物联网服务中间件的功能属性进行了分类比较; 最后, 展望了物联网服务中间件的发展趋势, 并指出了未来还需进一步改进和完善的功能。

**关键词:** 物联网; 中间件; 实体服务; 服务描述; 服务发现; 服务组合; 编程接口

**中图法分类号:** TP301

## 论文引用格式:

陈海明, 石海龙, 李勤, 崔莉, 物联网服务中间件: 挑战与研究进展, 2016, Vol.39, 在线出版号 No.117

CHEN Hai-Ming, SHI Hai-Long, LI Meng, CUI Li, Service Middleware for Internet of Things: Challenges and Approaches, 2016, Vol.39, Online Publishing No.117

## Service Middleware for Internet of Things: Challenges and Approaches

CHEN Hai-Ming<sup>1)</sup>, SHI Hai-Long<sup>1)</sup>, LI Meng<sup>1),2)</sup>, CUI Li<sup>1)</sup>

<sup>1)</sup> (Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

<sup>2)</sup> (University of Chinese Academy of Sciences, Beijing 100049)

**Abstract** Most of currently proposed methods to construct Internet of Things (IoT) software systems are based on Service Oriented Architecture (SOA), by abstracting both physical objects and cloud platforms as resources to provide services. Services provided by physical objects and cloud platforms are referred to as entity services and cloud services respectively. Service middleware for IoT, which lies over entity service and cloud service, provides infrastructure software for IoT task programming, discovering required services, supporting interconnection and interoperation among diverse services. However, developers are confronted with many challenges to build service

本课题得到国家国际科技合作专项项目(2013DFA10690)和国家自然科学基金项目(61672498, 61502461, 61100180)的资助。陈海明, 男, 1981 年生, 博士, 助理研究员, 中国计算机学会(CCF)会员, 主要研究方向为无线网络和网络化计算系统, 包括传感网和物联网, E-mail: chenhai ming@ict.ac.cn。石海龙, 男, 1986 年生, 博士, 助理研究员, 主要研究方向为传感网和物联网。李勤, 男, 1989 年生, 博士研究生, 主要研究方向为物联网资源的发现与检索。崔莉, 女, 1962 年生, 博士, 研究员, 博士生导师, 中国计算机学会(CCF)高级会员, 主要研究方向为传感器技术、无线传感器网络及物联网, E-mail: lcui@ict.ac.cn。

middleware for IoT, because of the unique characteristics of entity services, namely heterogeneity, large scale, dynamic availability, context awareness, spatial-temporal correlation, resource limitation, and actuating conflict. In this paper, we first present the functional model of service middleware for IoT, and outline challenges to design and implement service middleware for IoT when taking the characteristics of entity services into consideration. Confronted with these challenges, we present a comprehensive review of implementation methods of each module of service middleware for IoT. Then we make a comparison study on the functional property of the currently proposed service middlewares for IoT by analyzing their different mechanisms adopted to realize service description, service registration, service discovery, service composition and task programming interface. Finally, we look into future developing trend of middleware for IoT, and point out future work to be done to improve its functionality.

**Key words** Internet of Things; Service Middleware; Entity Services; Service Description; Service Discovery; Service Composition; Programming Interface

## 1 引言

随着传感技术、近距离通信技术和嵌入式计算技术的发展,物理空间中的物品不仅可以被赋予主动标识自己身份的能力,还可以获取自身属性和环境信息,并能执行一定的操作。目前将这种具有信息感知、处理、存储、通信和执行能力的物品称为智能物品(Smart Object)<sup>[1]</sup>,并将由海量的智能物品通过互联网连接组成的高度动态的、完全分布式的,用于实现物品的识别、定位、跟踪,以及物理世界的实时监控和智能管理的网络化系统称为物联网系统(Internet of Things)<sup>[2]</sup>。

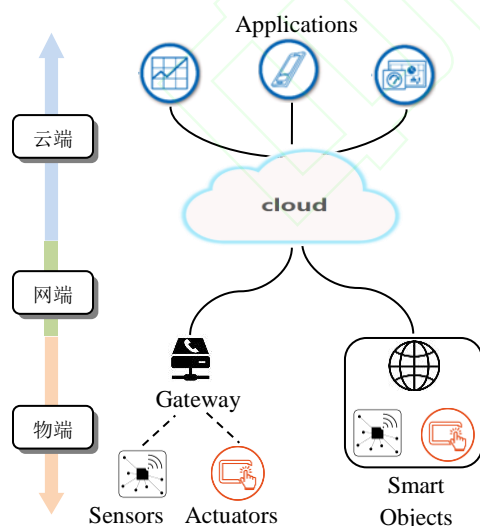


图 1 物联网系统的体系结构<sup>[3]</sup>

物联网系统的体系结构如图 1 所示,由“物端-网端-云端”三大部分组成:(1)物端(或称为海

端<sup>[4]</sup>,或前端<sup>[5]</sup>):由分布在互联网边缘的、属于不同用户、具有不同感知和执行能力的智能物品、智能网关和局域智能系统等物端资源构成;(2)网端:是将物端资源与云端资源互联起来的网络;(3)云端(或称为后端<sup>[5]</sup>):为多用户多应用提供计算、存储和信息处理等各种服务的云端资源构成。由于网端一般基于现有的互联网(Internet),因此物联网系统也被看作是建立在互联网之上的叠加网络(Overlay Network),或者说物联网是互联网技术的应用拓展<sup>[6]</sup>。

对于物联网系统的开发者来讲,实现物理空间中的智能物品、信息空间中的计算资源与社会空间中的用户之间的互联和互操作是构建“物端”和“云端”协同工作的物联网系统的基础。目前提出的物联网应用系统构建方法主要基于面向服务的体系结构(SOA)<sup>[7]</sup>,即将智能物品看作是物理信息的提供者和消费者,将其提供的功能和资源与现有互联网环境中的计算、存储和处理资源都统一抽象为服务<sup>[8][9]</sup>,并将它们作为系统的软件构件按照应用需求集成起来,组成“物端”和“云端”松耦合协同工作的网络化计算系统<sup>[10]</sup>。为便于论述,本文将云端资源提供的数据融合、数据分析、数据可视化等各种功能,称为云服务(Cloud Service),将物端资源提供的服务称为实体服务(Entity Service)。实体服务是物联网系统提供物理信息并与物理环境进行交互的基础构件,云服务是物联网处理物理信息的基础构件。实体服务又可以分为两种形式:1)物理实体服务(Physical Entity Service),是在物端资源上直接构建的获取物理信息或控制物端资源状态的软件构件;2)虚拟实体服务(Virtual Entity

Service), 是物端资源直接或通过物联网网关<sup>[11]-[14]</sup>与云端连接, 在云端构建的提供物理信息和控制物端资源状态的软件构件。实体服务具有以下主要特性:

(1) **异构性**: 在服务接口和交互协议上具有差异性, 并且在数据格式、数据语义和服务质量等方面也会有所不同, 如不同的传感器节点提供的感知服务具有不同的采样速率、精确度和空间分辨率等, 因此物联网中的实体服务具有很强的异构性;

(2) **大规模性**: 随着智能物品的数量不断增加, 将使得物联网系统中可以提供服务的资源规模不断扩大, 据无线世界论坛(WWRF)预测, 到2017年将有7万亿无线设备为70亿人服务<sup>[15]</sup>, 因此可以预见未来物联网中实体服务的规模会十分巨大;

(3) **动态可用性**: 在物联网系统中提供服务的智能物品会因移动和休眠等原因而失效, 导致实体服务的有效性在空间和时间上是动态变化的, 且不易被获知, 从而使得用户需要的服务并不是时时在线(always-on)的, 因此实体服务具有一定的动态性;

(4) **情境感知性**: 实体服务提供的感知数据和交互操作可以根据不同的情境进行相应的调整, 比如一个具有红外传感器的感知设备放置在室内时提供人员探测和防盗预警的功能, 放置在室外时提供测距的功能, 因此实体服务不仅具有动态可用性, 而且具有情境感知性;

(5) **时空相关性**: 实体服务提供的数据和操作在时间上和空间上具有一定的相关性, 比如PM2.5感知服务, 某个地点与相邻地点的监测数据具有一定的相关性, 并且某个时间点的监测数据与过去或未来近几个时间点的监测时间具有一定的相关性, 从而使得用户在需要的服务不可用时也能通过一定的相关推断来获得近似的服务;

(6) **资源受限性**: 对于物理实体服务来讲, 由于处理服务请求的智能物品或智能网关的计算和存储能力有限, 因此它们能够并发响应的服务请求数量有限, 并且感知数据的采样频率也有一定的限制, 因此在资源共享的物联网环境中, 对物理实体服务的访问需要进行一定的约束;

(7) **执行冲突性**: 对于提供跟物理世界交互能力的实体服务而言, 来自不同用户的服务请求之间可能会产生冲突, 比如A用户需要在室内温度低于23度时关闭空调, 而B用户需要在室内温度低于20度时关闭空调, 因此需要一定的冲突检测与

解决机制来保证实体服务的执行可靠性。

根据构建物联网系统时是否采用物理实体服务, 可以将物联网系统分为以下两种类型:

(1) 以“物端”为中心的物联网系统, 或前端分布式物联网系统<sup>[5]</sup>: 这类物联网系统大多基于M2M<sup>[16]</sup>、IoT-A<sup>[17]</sup>、PMDA<sup>[18]</sup>等物联网体系结构, 以物理实体服务为基础构件, 系统开发者可以直接调用这些服务, 也可以将物端资源提供的服务与云端资源提供的数据分析服务、可视化服务等云服务根据应用需求组合起来, 如WOT(Web of Things)<sup>[19]</sup>、SenseWeb<sup>[20]</sup>、SWE(Sensor Web Enablement)<sup>[21]</sup>和SSW(Semantic Sensor Web)<sup>[22]</sup>。

(2) 以“云端”为中心的物联网系统, 或后端集中式物联网系统<sup>[5]</sup>: 这类物联网系统大多采用与CEB(Cloud Edge Beneath)<sup>[23]</sup>类似的物联网体系结构, 以虚拟实体服务为基础构件, 系统开发者在云端将虚拟实体服务与云服务组合起来实现物联网系统的构建, 如Pachube<sup>[24]</sup>、Aneka<sup>[25]</sup>、Atlas<sup>[26]</sup>和AirCloud<sup>[27]</sup>等。目前被广泛关注的移动群智感知系统<sup>[28]</sup>(也被称为参与式感知系统<sup>[29]</sup>或以人为中心的感知系统<sup>[30]</sup>)也属于这类物联网系统, 它们以智能移动终端为前端感知设备, 以机会方式(Opportunistically)采集用户所处环境的物理信息, 并将采集到的信息传输到云端, 通过与已有的或新建的相应数据分析处理服务相组合建立满足用户需求的感知系统。

不论是哪类物联网系统, 以上所述的实体服务的这些特性使得高效的管理、发现与组合实体服务与云服务, 建立面向多用户资源共享的物联网应用软件, 实现物端与云端协同工作的任务执行, 变得更加复杂。根据分布式系统和互联网软件的发展经验, 需要在物联网云端和物端设备的操作系统之上构建中间件(Middleware)<sup>[31]</sup>, 即一组管理计算资源、实现资源之间互联和互操作的支撑服务(Enabling Service), 为上层应用软件提供运行与开发的环境, 帮助用户灵活、高效地开发和集成复杂的软件构件。由于中间件是由一组支撑服务构成的基础软件, 因此也被称为服务中间件。目前国内外研究人员已经从物联网软件体系结构<sup>[32]</sup>的设计出发, 将定义在实体服务和云服务之上的用于支撑物联网应用系统构建的软件模块独立出来, 提出了相应的物联网服务中间件, 如Atlas Cloud/Edge/Sensor Platform Middleware<sup>[26]</sup>、WInternet<sup>[33]</sup>、Physicalnet<sup>[34]</sup>、3CoFramework<sup>[35]</sup>、3Tier<sup>[36]</sup>、SOCRADES<sup>[37]</sup>、



SENSEI<sup>[38]</sup>、Physical Mashup<sup>[39]</sup> 和 InterDataNet<sup>[40]</sup> 等。

目前已有研究人员对国际上在物联网中间件方面的工作进展进行了综述<sup>[41][42]</sup>，其中 Bandyopadhyay 等人<sup>[41]</sup>侧重于物联网中间件的功能概括，包括支持互操作、设备发现与管理、数据管理、安全与隐私保护，以及情境检测；Razzaque 等人<sup>[42]</sup>将物联网中间件分为以下几类：基于事件、面向服务、基于智能体、基于元组空间、应用特定的中间件，以及基于虚拟机或数据库的中间件，并从中间件实现的功能和非功能需求（包括可扩展性、安全性、可用性、可靠性、实时性和私密性），以及为构建应用提供的支持（包括编程抽象、互操作、情境感知、自主交互、自适应调整、分布式和轻量级）方面，对多个物联网中间件（包括针对传统的多跳自组织无线传感网设计的中间件，比如 SensorBus<sup>[43]</sup>、Sensorware<sup>[44]</sup>、SensorWap<sup>[45]</sup>、Hydra<sup>[46]</sup> 和 GSN<sup>[47]</sup>等）进行了分析。本文将重点以构建图 1 所示的物联网系统为目标，从功能模型的角度，提出设计与实现物联网服务中间件各功能模块所面临的挑战，并对其研究进展进行总结分析。

下面，首先给出物联网服务中间件的功能模型，并以此为参考，结合实体服务的特性，提出设计与实现物联网服务中间件中各功能模块需要面临的挑战；接着对目前已经提出的物联网服务中间件各模块的实现方法进行总结；然后对目前提出的各种物联网服务中间件进行分析比较，最后对目前物联网服务中间件的进展进行总结并展望未来物联网服务中间件的发展趋势。

## 2 物联网服务中间件的功能模型

基于物联网软件体系结构的设计<sup>[32]</sup>，将位于实体服务和云服务之上，用于支撑物联网任务编程、发现满足任务需求的服务集以及实现各类服务之间互联与互操作的基础结构软件定义为物联网服务中间件<sup>[41][42]</sup>。图 2 给出了面向服务的物联网服务中间件的功能模型，它以实体服务和云服务为基础，由服务注册管理、服务发现、服务组合和编程接口这四个功能模块构成，为用户提供统一的高层服务抽象、服务管理、服务发现和组合等功能，以及构建物联网应用系统的编程接口。下面对各功能模块进行简要介绍。

(1) 物理实体服务、虚拟实体服务和云服务：

物理实体服务和虚拟实体服务是指智能物品、智能网关和局域智能系统等物端资源提供的感知物理信息并与物理环境进行交互的功能模块；云服务是指云端资源提供的数据融合、数据分析、数据可视化等各种功能的软件模块。

(2) 服务注册管理：该模块可以采用集中式的服务注册中心（Registry）或目录服务（Catalogue Service）的形式来管理物联网中的各类服务，也可以将各类服务的注册信息分布式地部署在物联网的各计算平台中。集中式的服务注册中心也被称为服务仓库<sup>[34]</sup>、协调器<sup>[35]</sup>、服务注册中心<sup>[36]</sup>或资源目录<sup>[38]</sup>，负责管理物联网中各类服务的注册信息，以便服务的消费者能够准确快速地获得所需的服务。

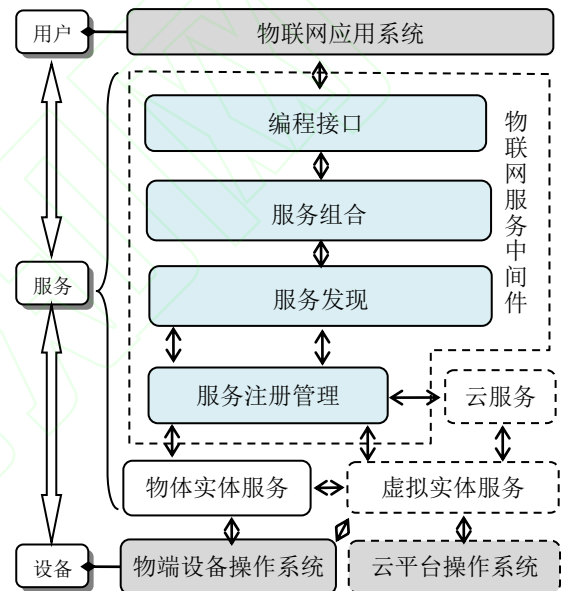


图 2 面向服务的物联网服务中间件的功能模型

(3) 服务发现：按照用户提供的构建物联网应用所需服务的描述，该模块根据物联网服务注册管理机制的不同，采用集中式的匹配查找或分布式的概率发现机制来找到满足应用需求的实体服务和云服务，从而为用户直接或通过服务组合获得所需的物联网服务提供必要的支持。

(4) 服务组合：也被称为连接器<sup>[35]</sup>、服务组合模块<sup>[36]</sup>、任务调度模块<sup>[20]</sup>或资源汇聚模块<sup>[38]</sup>。该模块从编程接口模块获得应用需求，并按照需求处理执行计划通过服务发现模块找到需要的实体服务和云服务，选择多个服务将它们编排起来形成可运行的物联网系统，以完成指定的应用任务。

(5) 编程接口：也被称为编程抽象模型管理模块<sup>[34]</sup>、感执任务创建模块<sup>[38]</sup>、业务流程处理模块

[17]或需求规划模块[18]。该模块给系统开发者提供定义应用需求规划的功能，并负责任务需求的解析，发起服务发现的过程，确定业务处理流程供服务组合模块建立可运行的系统。根据该模块提供的编程抽象的层次不同，还会包括需求分解、服务映射和工作流映射等子模块。

### 3 设计与实现物联网服务中间件的挑战

基于以上对物联网服务中间件的功能模型的分析可以看出，实体服务和云服务的构建是设计与实现物联网服务中间件的基础，而云服务以 PaaS、IaaS 或 SaaS 的形式由云端计算平台提供，它们可以是已经部署在公共云平台中的服务，如 Amazon EC2、Google AppEngine 和 Microsoft Azure 上的计算和存储服务，也可以是部署在各单位的私有云平台中的服务。由于云服务的构建理论与方法已经经过了多年的研究，形成了比较完整的技术体系[48]，本文不再进行论述。本文的论述主要围绕着物联网中的实体服务（尤其是物理实体服务）和物联网服务中间件各功能模块（包括服务注册管理、服务发现机制、服务组合和编程接口）的设计与实现而展开。本节将结合实体服务的特性，指出目前设计与实现物联网服务中间件各功能模块将面临的挑战。

表 1 给出了设计与实现物联网服务中间件需要考虑的实体服务的特性，以及相应的对各中间件模块的特别功能要求。这些特别的功能要求给设计与实现物联网服务中间件带来了以下挑战：

（1）具有情境检测、访问控制和冲突解析能力的实体服务提供。实体服务除了与云服务一样需要提供易于访问的服务接口外，还需要提供情境检测的能力，以根据情境的变化提供相应的服务，同时还需要考虑物端资源的受限性和执行操作的冲突性，提供访问控制和冲突解析的功能；

（2）以统一的形式描述各类服务，并能够描述实体服务的提供约束以及情境与服务提供之间的关系。实体服务的异构性是导致物联网环境中各服务之间的不易互联与互操作的主要原因，因此物联网服务中间件首先需要提供统一的服务描述，并且还要考虑实体服务的资源受限性和情境感知性，提供关于服务约束和服务情境的描述；

（3）支持大规模实体服务的注册管理，且能

够根据实体服务所处的环境和状态的变化，对服务注册信息进行更新维护。可扩展性是物联网环境中对各类服务进行注册管理需要考虑的重要因素，同时还需要考虑实体服务的动态可用性和情境感知性，提供高效的注册信息维护与更新功能；

（4）支持大规模、动态可用实体服务的高效快速发现，并能够在服务发现过程进行时空相关推理和轻量级服务匹配，保证物联网环境中服务发现的响应率和可扩展性。考虑未来物联网环境中各种实体服务和云服务的涌现，并考虑实体服务的动态可用性，需要采用一定的机制来减小搜索空间，来保证服务发现的响应速度；同时还需要考虑实体服务的时空相关性，需要基于时空推理的搜索方法，

表 1 设计与实现物联网服务中间件面临的挑战

| 功能模块   | 需要考虑的实体服务的特性 | 对各物联网服务中间件模块的特别功能要求 |
|--------|--------------|---------------------|
| 实体服务构建 | 情境感知性        | 情境检测                |
|        | 资源受限性        | 访问控制                |
|        | 执行冲突性        | 冲突解析                |
| 服务描述   | 异构性          | 统一的服务描述             |
|        | 情境感知性        | 情境感知的服务描述           |
|        | 资源受限性        | 服务提供的约束描述           |
| 服务注册管理 | 大规模性         | 可扩展服务注册管理           |
|        | 动态可用性        | 注册信息动态维护            |
|        | 情境感知性        | 注册信息更新维护            |
| 服务发现   | 大规模性         | 搜索空间约简              |
|        | 动态可用性        | 可用性评估               |
|        | 时空相关性        | 时空相关推理              |
|        | 资源受限性        | 轻量级服务匹配             |
| 服务组合   | 资源受限性        | 动态选择                |
|        | 执行冲突性        | 冲突检测                |
| 编程接口   | 动态可用性        | 任务需求动态规划            |
|        | 情境感知性        | 任务需求动态规划            |

来提高服务发现的响应率；此外还要考虑物理实体服务的资源有限性，需要采用轻量级服务匹配的方法，来提高服务发现的可扩展性；

（5）支持多用户服务共享，具有物端资源执行冲突检测能力的服务组合。多用户共享是面向服务的中间件具有的一般特性，但是在物联网环境中，考虑到实体服务的资源受限性和执行冲突性，需要在组合构建满足任务需求的物联网应用系统时，动态选择满足服务质量要求的实体服务，并且在组合服务时检测不同应用的执行规则，以避免发生冲突；

(6) 支持任务需求动态规划的编程接口。考虑到实体服务的动态可用性和情境感知性, 用户通过编程接口定义的任务执行需要绑定的实体服务具有一定的动态性, 因此物联网服务中间件提供的编程接口需要具有动态规划任务需求的能力。

## 4 设计与实现物联网服务中间件的方法

以上结合实体服务的特性, 指出了设计与实现物联网服务中间件各功能模块将面临的挑战。本节将对目前已经提出的物联网服务中间件各模块的实现方法进行分析。

### 4.1 实体服务构建

在物端设备或云平台上建立可互联的实体服务是构建面向服务的物联网应用系统的基础。物联网服务中间件所面向的实体服务包括虚拟实体服务和物理实体服务两种类型。虚拟实体服务一般是将来自物端资源的数据和控制接口封装成统一的云服务形式提供给系统开发者。因此, 虚拟实体服务的构建方法与云服务没有本质差别, 为节省篇幅, 本文略去这方面工作的论述。在物端的感知和执行设备或网关<sup>[49]</sup>上建立的物理实体服务不仅需要考虑资源受限性, 提供本地数据管理和代码管理的功能, 还需要提供情境检测、访问控制和操作冲突解析等功能。因此, 物端设备上支持物理实体服务构建的系统平台软件也被看作是物联网物端感知和执行设备或网关上的中间件<sup>[50]</sup>。

实现物理实体服务的基础是物端的感知和执行设备或网关上建立与互联网中的设备一样的数据通信协议栈。最早由瑞典计算机科学研究所的 Dunkels 在传感器节点上实现了嵌入式 IPv4 协议 uIP<sup>[51]</sup>, 近年来在 IPSO<sup>1</sup>等组织的推动下, IETF 制订了在低功耗低速率的个域网协议 (IEEE 802.15.4) 之上实现 IPv6 协议的适配层协议 6LoWPAN<sup>[52]</sup>, 使资源受限的感知和执行设备能够通过 IPv6 协议接入互联网成为网络空间中可以被直接寻址的设备。基于 uIP 或 IPv6/6LoWPAN 协议, 软件开发者可以在感知和执行设备上建立可以被直接寻址和访问的服务, 以使它们作为独立的软件模块向上层应用系统提供采集物理信息和执行控制动作等服务。

物理实体服务的软件架构是保证所建立的物联网应用系统具有良好的服务互联性、大粒度的软件可重用性和良好的系统可维护性的关键。目前在感知和执行设备或网关上实现物理实体服务所采用的软件架构可以分为四种: 分布式对象、智能体、SOAP 风格的 Web 服务和 REST 风格的服务。表 2 对这四种类型的实体服务的具体实现架构进行了比较。可见, 在物端资源上实现分布式对象的各层机制与现有的规范 (如 CORBA、DCOM、RMI) 一致, 即用 IDL (Interface Definition Language) 描述分布式对象所提供的服务, 对象之间交换的数据遵循 CDR (Common Data Representation) 格式, 对象之间的交互采用 IIOP (Internet Inter-Orb Protocol)、RPC (Remote Procedure Call) 或 RMI (Remote Method Innovation) 协议, 这些交互协议都是基于可靠的数据传输协议 TCP。

智能体作为一种特殊的分布式对象或 Web 服务, 在物端节点或网关上进行实现的架构可以参照 CORBA、SOAP 风格的 Web 服务和 REST 风格的 Web 服务。但是, 为了便于以不同形式实现的智能体之间的交互, FIPA (The Foundation for Intelligent Physical Agents) 为智能体各层实现机制制定了规范。智能体按照“黄页” (DF, Directory Facilitator) 定义的 Agent 注册格式描述服务, 智能体之间的交互以 ACL (Agent Communication Language) 为数据表示格式, 以 MTP (Message Transport Protocol) 为数据交换协议。

对比两种风格的 Web 服务的各层实现机制可以看出, 在物端节点或网关上可以基于不同的应用层协议实现 SOAP 风格的 Web 服务, 但是必须采用统一的 WSDL 对提供的服务进行描述; 对于 REST 风格的 Web 服务来讲, 由于其采用应用层协议定义的操作原语 (GET/PUT/POST/DELETE) 作为服务的标准接口, 因此这个服务描述并不是必须的。为了便于服务的访问和组织, 未来也可以按照 Atom 发布协议 (Atom Publisher Protocol) 的格式将感知和执行设备提供的服务进行发布。

下面对这四种实体服务的软件架构和各层实现机制进行具体分析。

#### 4.1.1 基于分布式对象架构实现的实体服务

分布式对象是在分布异构环境下建立应用系统的框架和模式, 目前这种模式主要被用于开发 Internet 应用软件的设计开发。Vicaire 等人<sup>[34]</sup>将这

<sup>1</sup> IPSO. <http://www.ipsso-alliance.org>, 2010.



种模式引入到物联网应用系统的设计开发，提出了 Physicalnet 这一实现物理实体服务的架构（如图 3 所示）。可见，Physicalnet 的实体服务由节点实现，通过一个细粒度访问控制和冲突解析机制来保护资源的共享，并支持在线权限分配。传感和执行节点上实现的服务由网关进行收集后以远程方法调用（RMI）形式向外发布。

北京大学的研究人员提出了 ThingWare<sup>[53]</sup>智能物品中间件也是基于分布式对象的软件架构，包括五个模块：1）智能物品交互（Smart Things Interactions）模块，主要实现智能物品通信协议（STCP, Smart Things Communication Protocol）以及智能物品的邻居发现；2）智能物品管理（Smart Things Management）模块，记录发现的智能物品的属性、标识信息和网络接口等；3）服务管理（Services Management）模块，管理智能物品本地的服务，同时维护发现的邻居智能物品所提供的服务以方便应用程序查询；4）智能物品本地服务接口（Local API），给出智能物品提供的服务；5）智能物品服务管理接口（Things API），提供物品查询、智能物品服务查询和调用，以及智能物品本地服务注册与注销的服务。

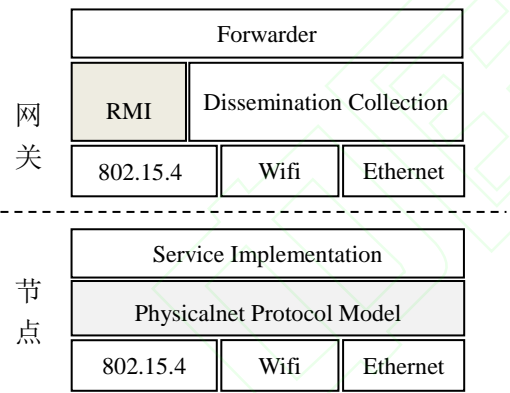


图 3 Physicalnet 物理实体服务的实现架构

考虑到单个物端资源上建立多个实体服务的情形，以及实体服务的动态可用性和情境感知性，目前还有研究人员提出了在此类物理实体服务的架构基础上建立服务引擎，如 Nain<sup>[54]</sup>等人以 OSGi 运行时平台为基础建立了一个具有反射性（Reflective）的服务引擎（CBSE），它为物端资源的多个服务间连接关系提供外部管理接口，满足由物联网的事物动态变化引起的自适应监控的需求，达到松耦合和动态配置的目的。Wu 等人<sup>[55]</sup>也以 OSGi 平台为基础，通过移动智能体脚本标记语言

（MASML）文档的传输实现服务的动态加载，并按需执行感知任务和返回结果。

4.1.2 基于智能体架构实现的实体服务

以分布式对象形式实现的物理实体服务一般不具有情境感知的能力，即这些服务无法根据所处环境的变化自主进行物理信息的采集或控制动作的执行。考虑到实体服务的情境感知性，目前提出了一些基于智能体的实体服务设计架构与实现方法，如 Agent-oriented Smart Object (ASO)<sup>[56]</sup>和 Smart Products<sup>[59]</sup>。

ASO 的实现架构如图 4 所示，与 Physicalnet 中给出的实体服务的架构类似，ASO 也包括分别在节点和网关上实现的软件模块。节点主要基于 INF（Internal Node Framework）实现多请求调度、动态群组管理和节点内信号处理的功能；网关主要基于 ISOF（Internal Smart Object Framework）实现请求调度和群组管理的功能，并在此功能之上实现面向智能体的中间件，提供情境管理、知识仓库适配和知识推理的功能。节点与网关之间采用 ISOP（Intra Smart Object Protocol，如 IEEE 802.15.4）进行通信，网关之间采用 ESOP（External Smart Object Protocol，如 TCP/IP）进行信息交换。在文献[56]中，作者基于 JADE 和 MAPS（Mobile Agent for SUN SPOT）<sup>[57]</sup>开发框架实现了 ASO。节点上的多请求调度（Multi Request Scheduling）模块用于优化感知节点的能量消耗，并保证用户请求的处理延迟最低<sup>[58]</sup>。

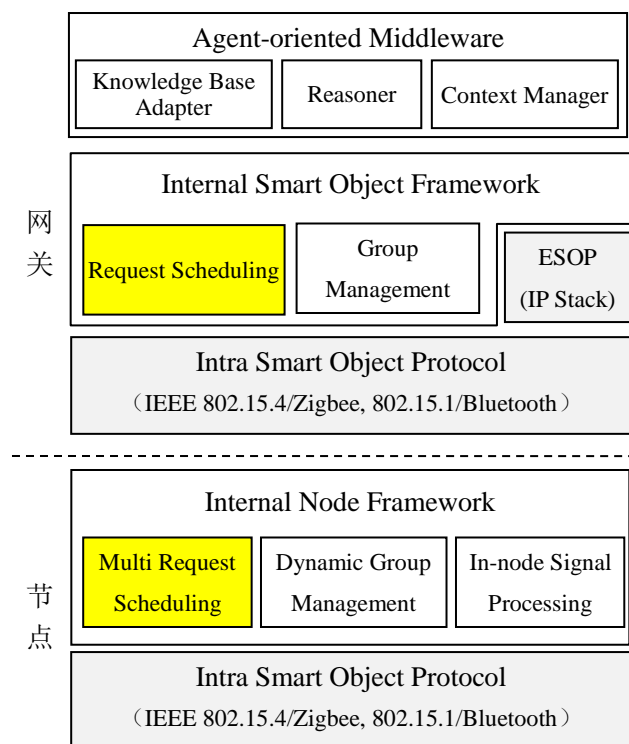


图 4 Agent-oriented Smart Object 的实现架构

Smart Products<sup>[59]</sup>是为构建具有异构信息源的智能产品定义的实现架构，它由两个模块组成：先验知识模块（Proactive Knowledge Module）和推理模块（Reasoner Module）。先验知识模块中的数据分为：元数据模型（Meta Model）、时间模型（Time Model）、用户模型（User Model）、情境模型（Context Model）和领域模型（Domain Model）；推理模块基于先验知识模块，采用 RDFS<sup>2</sup>或 OWL<sup>3</sup>进行本体推理，实现基于用户需求、领域知识和所处环境的智能实体服务。

#### 4.1.3 基于 SOAP 风格的 Web 服务架构实现的实体服务

随着万维网（World Wide Web）技术的发展，互联网中的资源一般基于 HTTP 协议提供各种 Web 服务。为了便于以后将感知和执行设备上建立的服务与互联网上的这些服务进行互联与集成，SenseWeb<sup>[20]</sup>的设计者基于 HTTP 协议在感知网关上实现了 SOAP 风格的 Web 服务。由于 HTTP 协议建立在 TCP 协议之上，而 TCP 采用复杂的流量控

制和重传机制实现可靠传输，实际上在资源受限的物联网感知设备上建立 Web 服务具有一定的挑战。为此，IETF 成立了 CoRE（Constrained RESTful Environment）<sup>[60]</sup>工作组，为在资源受限的设备上建立 Web 服务制订了与 HTTP 对等的轻量级应用层协议 CoAP（Constrained Application Protocol）<sup>[61]</sup>和 EBHTTP（Embedded Binary HTTP）<sup>[62]</sup>。这两种协议都建立在 UDP/IPv6/6LowPAN 协议栈之上，并已经分别在 Contiki 和 TinyOS 操作系统中实现<sup>[63]</sup>。

Moritz 等人<sup>[64]</sup>提出了一种在 CoAP 协议之上建立 SOAP 风格的 Web 服务的方法，并采用 EXI 数据封装格式，进一步提高数据传输效率。本文将这种 Web 服务简称为 EXI/SOAP-over-CoAP 服务的话，将 SenseWeb 实现的 Web 服务简称为 XML/SOAP-over-HTTP，表示它建立在 HTTP 协议之上，并采用 XML 数据封装格式。为实现互联网中基于 HTTP 协议的 Web 服务与嵌入式设备上基于 CoAP 协议的 Web 服务之间的透明连接，Colitti 等人建立了一种 HTTP-CoAP 代理服务器<sup>[65]</sup>，用于实现 HTTP 协议与 CoAP 协议之间的相互转换。为进一步降低数据传输的协议开销，由德国 WS4D 项目组提出的 DPWS<sup>[66]</sup>架构直接将 SOAP 风格的 Web 服务建立在 UDP 协议之上。IBM 提出的面向服务的设备架构（SODA）<sup>[67]</sup>包括三个主要模块，即设备适配模块，面向各种私有和工业设备的接口标准；总线适配模块，将设备提供的服务映射为企业使用的指定的 SOA 绑定机制；设备服务注册模块，提供发现和访问 SODA 服务的功能。

以上这些基于不同协议在感知和执行设备上实现 SOAP 风格的 Web 服务的架构可以用图 5 来统一表示。由于采用 SOAP 风格建立的 Web 服务还需要开发者使用 WSDL 对其进行具体描述，因此在 Web 服务之上还需要建立相应的发布、管理和查找机制，用 WS-\*表示。需要说明的是，虽然目前还没有在感知和执行设备中基于 EBHTTP 协议建立 SOAP 风格的 Web 服务，但是这样的实现架构是可行的，因此在图 5 中给出了这样的情形示意。

<sup>2</sup> RDFS. <http://www.w3.org/TR/rdf-schema/>, 2004

<sup>3</sup> OWL. <http://www.w3.org/TR/owl-ref/>, 2004



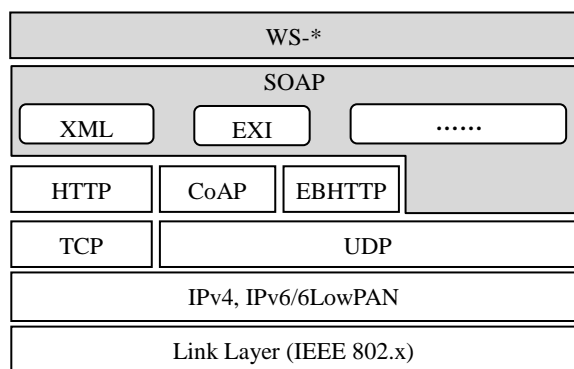


图 5 WS-based Web 风格的实体服务实现架构

#### 4.1.4 REST 风格的 Web 服务架构实现的实体服务

目前互联网中更多的是 REST 风格的 Web 服务，它们使用 HTTP 协议定义的四种操作原语（GET/PUT/POST/DELETE）作为服务接口，使用 URI 标识服务，使用 RSS 或 ATOM 来发布和联合组织服务。为此，最早由 Drykiewicz 等人<sup>[68]</sup>在 MicaZ 节点上基于 uIP 和 HTTP 协议实现了 REST 风格的 Web 服务 pREST；后来 Schor 等人<sup>[69]</sup>提出了一种以即插即用（Plug-and-Play）的方式在节点上实现 REST 风格的 Web 服务的方法 ZeroConfig。pREST 和 ZeroConfig 都是基于 HTTP 协议在传感器节点上实现的 Web 服务，而 Guinard 等人<sup>[39]</sup>考虑到传感器节点的资源受限性，将各传感器节点提供的的数据由智能网关进行缓存和格式转换后以 REST 风格的 Web 服务形式提供给系统开发者和用户；Luckenbach 等人<sup>[70]</sup>基于类似的考虑，提出由家庭网关将不同传感网、自动化家居与消费电子设备，以及通过 Bluetooth SDP、UPnP<sup>4</sup>、JINI<sup>5</sup>、DLNA<sup>6</sup>和 Infrared<sup>7</sup>等服务发现协议<sup>[71]</sup>管理的局域智能系统提供的服务统一为 REST 风格的 Web 服务（TinyREST）后接入互联网。

由于 HTTP 协议以 TCP 协议为数据传输控制协议，虽然能够保证数据传输的可靠性，但是会引起较大的协议控制开销，Duquenooy 等人<sup>[72]</sup>以事件驱动的方式优化了 HTTP 的性能，并通过编译时页面预处理，以在资源受限的节点上实现 PULL/PUSH 形式的 Web 服务；Yazar 等人<sup>[73]</sup>为提高节点提供的 Web 服务性能，对节点的 MAC 协议和 HTTP 协议

同时进行了改进。考虑到目前已经由 IETF CoRE 工作组基于 UDP 协议制订了轻量级应用层协议 CoAP 和 EBHTTP，Castellani<sup>[74]</sup>等人提出了一个在节点上实现 REST 风格的 Web 服务的架构 BWS（Binary Web Service）；Mainetti<sup>[75]</sup>等人实现了基于 CoAP 协议的 REST 风格的物理实体服务，并给出了一种通过代理服务器（Proxy Server）实现与基于 HTTP 协议的 Web 服务之间进行互联的方法；Dawson-Haggerty<sup>[76]</sup>等人基于 EBHTTP 协议建立了另一个在节点上实现 REST 风格的 Web 服务的架构 sMAP。此外，Hou<sup>[77]</sup>等人考虑到用户请求操作物理实体服务的群组性，即用户一般不是请求对单一的某个物理实体服务进行操作，而是对一组具有时空相关性的服务进行操作，对 CoAP 协议进行了改进，新增了 COMBINE 和 BRANCH 两个操作原语，同时增加了数据缓存机制以及时响应节点休眠时的服务请求，改进后的协议称为 SeaHTTP。基于 SeaHTTP，本研究组也在传感器节点上实现了 REST 风格的 Web 服务。

以上这些采用不同的应用层协议在节点或网关上建立 REST 风格的 Web 服务的架构可以用图 6 来统一表示。可见，在物联网前端实现 REST 风格的 Web 服务的架构比实现 SOAP 风格的 Web 服务的架构减少了一层，从而降低了物理实体服务与其他服务之间进行互联与组合的复杂度。

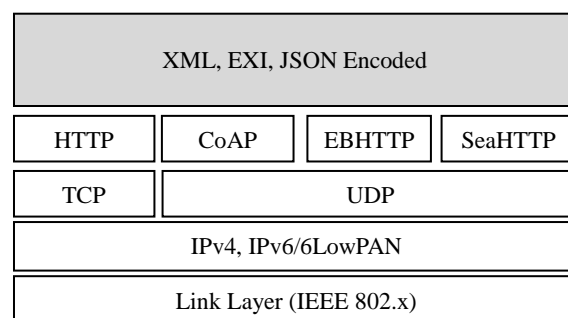


图 6 RESTful Web 风格的实体服务实现架构

通过以上对四种实体服务的软件架构和各层实现机制的比较分析，对照表 1 给出的构建实体服务所面临的挑战，不难发现基于智能体架构实现的实体服务考虑了情境感知性和资源受限性带来的挑战，设计了情境检测和访问控制的功能模块，而其它三种实体服务都没有考虑这两个特性，因此也不具有情境检测和访问控制的功能。但是，它们都没有考虑执行冲突性带来的挑战，因此都没有设计

<sup>4</sup> UPnP. <http://www.upnp.org/>, 2006.

<sup>5</sup> JINI. <http://www.jini.org/>, 2011.

<sup>6</sup> DINA. <http://www.dlna.org/>, 2016.

<sup>7</sup> IRDA. <http://www.irda.org/>, 2011.

冲突解析的功能。

表 2 物理实体服务的实现架构比较

| 实现架构层次 | 分布式对象  | 智能体  | SOAP 风格的<br>Web 服务   | REST 风格的<br>Web 服务  |   |
|--------|--|--|--|---|---|
| 服务描述   | IDL  | DF Agent<br>Description                                  | WSDL   | Atom Publisher Protocol   |   |
| 数据表示格式 | CDR  | ACL  | XML<br>EXI   | XML<br>EXI<br>JSON  |   |
| 服务交互协议 | IIOP<br>RPC<br>RMI                                       | MTP  | SOAP-over-HTTP<br>SOAP-over-CoAP<br>SOAP-over-EBHTTP                     | HTTP<br>CoAP<br>EBHTTP<br>SeaHTTP   |   |
| 数据传输协议 | TCP  | TCP  | TCP<br>UDP   | TCP<br>UDP  |   |
| 实例     | Physicalnet <sup>[34]</sup><br>ThingWare <sup>[53]</sup> | ASO <sup>[56]</sup><br>Smart<br>Products <sup>[59]</sup> | SenseWeb <sup>[20]</sup><br>DPWS <sup>[66]</sup><br>SODA <sup>[67]</sup> | smartGateway <sup>[39]</sup><br>pREST <sup>[68]</sup><br>ZeroConfig <sup>[69]</sup> | TinyREST <sup>[70]</sup><br>BWS <sup>[74]</sup><br>sMAP <sup>[76]</sup> |

#### 4.2 服务描述

从表 2 可以看出, 基于这四种架构实现的实体服务之间的最大区别在于它们提供的服务接口的形式不同。具体来讲, 分布式对象是以远程方法或自定义的应用程序接口的形式提供, 智能体是以自定义的交互接口的形式提供, SOAP 风格是以自定义的 Web 服务接口的形式提供, REST 风格的 Web 服务是以标准化的 HTTP 操作原语的形式提供。因此, 面对不同形式的自定义服务接口, 并且考虑有些实体服务(如基于智能体架构实现的实体服务)具有情境感知的服务提供能力, 目前提出了一些物联网中实体服务与云服务一致的服务描述方法。

服务描述的元素一般包括以下几个部分: (1) 服务的输入和输出; (2) 前件和效果(或后件); (3) 非功能属性, 比如 QoS、服务成本、服务的位置或者服务请求者的偏好、时空信息等; (4) 其他描述信息, 比如服务名称、服务类型、操作名称和数量、服务的文本描述和 WSDL 描述等。

从表 2 可以看出, 目前对实体服务的接口说明仅限于语法级别(syntactic)的描述, 主要说明服务的地址、服务提供的方法, 以及访问服务的参数格式和操作协议。由于不同类型的实体服务采用具有不同语法的描述语言, 因此不易于服务之间的互联与互操作。针对该问题, 目前已经提出了更高抽象层次的服务描述方法, 即基于语义(semantic)

的描述方法。由于本体(ontology)能够通过一些结构化的词汇来精确表达特定领域相关的概念(concept)或类(class), 实现数据和服务信息的精确描述和交换, 因此本体成为语义描述的基础。

目前最常用的本体描述语言是 OWL 和 OWL-S。以 OWL-S<sup>8</sup>为例, 采用“Profile-Process- Grounding”服务描述模型, 包括三个部分: 1) profile, 定义服务的名字、描述、输入/输出语义; 2) process, 定义输入和输出的参数; 3) grounding, 定义用户与服务交互时的细节, 包括入口点和参数等。

Avancha 等人<sup>[78]</sup>最早将本体用于描述传感网中节点系统的属性, 如处理器性能、存储容量、电量、无线通信模块、感知模块。Michita 等人<sup>[79]</sup>将语义网中类(class)和实例(instance)的概念应用于传感网, 提出了语义传感网(SSN, Semantic Sensor Network)的概念。Eid 等人<sup>[80]</sup>基于 IEEE 1451 标准提出了感知数据的本体描述方法, 包括(1)收集智能传感器相关的概念词汇表; (2)建立最初的层次化概念分类; (3)确定概念之间的关系和约束条件; (4)归类测试和一致性检查。现在, W3C 组织成立了专门的 SSN 孵化工作组(SSN-XG)来建立 SSN 本体<sup>[81]</sup>, 不仅描述智能传感器提供的感知数据服务, 还描述它们提供的数据库操作和参数配置等相关其它服务<sup>[82]-[84]</sup>。

<sup>8</sup> OWL-S: Semantic Markup for Web Services, <http://www.daml.org/services/owl-s/1.0/>, 2004

Mokhtar 等人<sup>[85]</sup>针对普适计算环境的服务特性，提出了一种基于本体的服务描述语言（EASY-L）。EASY-L 描述的服务模型包含能力（Capacity）和属性（Property）两个模块。能力模块用于表达设备提供的服务或用于请求的服务，即服务的功能性属性，具体包括输入（Input）、输出（Output）和类型（Category）三个部分；属性模块用于表达服务的非功能性属性，包括服务的情境信息（Context），如物理环境信息、用户信息、平台的软硬件信息和服务质量信息（QoS Information），如服务代价、服务性能、服务可靠性、安全性等。图 7 给出了 EASY-L 描述的服务模型，其中类型、输入、输出、情境信息和服务质量信息这几个模块都以本体的形式进行定义和描述。

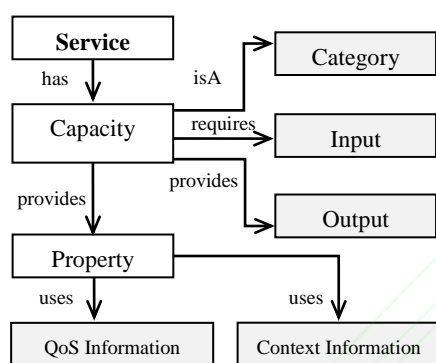


图 7 一种基于本体的服务描述模型

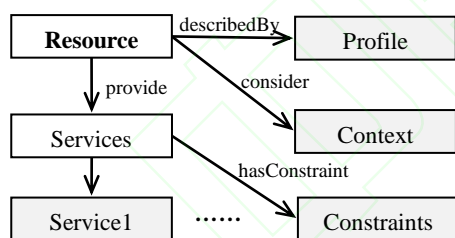


图 8 一种基于本体的物端资源和物理实体服务描述模型

Huang 等人<sup>[86]</sup>指出了物端资源与云端资源之间的区别：（1）物端资源提供的服务与所处的环境有关，即处于不同环境或来自不同地方的请求，提供的实体服务不同；（2）每个物端资源上可以建立多个服务，但是物端资源的计算能力和存储能力有限。基于这样的区别，他们建立了一种具有情境和资源意识的物端资源模型（如图 8 所示），并且基于扩展的 OWL-S 语言对物端资源及其提供的服务进行描述，其中资源属性（Profile）、资源所处情境（Context）、服务（Service）、服务提供约束（Constraints）这几个模块都以本体的形式进行定

义和描述。

Fortino 等人<sup>[87]</sup>用 JSON 语言定义了元数据模型，包括四种数据类型：（1）类型（Type），用于描述智能物品的标识或标签；（2）设备（Device），用于描述智能物品的物理属性，比如存储容量，运算速度和操作系统等；（3）服务（Service），用于描述智能物品提供的服务，包括服务索引、服务名称、服务类型、返回类型、服务说明和操作，通过服务名称和服务说明中的关键字来发现服务；（4）位置（Location），用于描述智能物品的绝对位置或相对位置。这四种数据类型相当于描述智能物品功能性和非功能性属性的本体，因此基于元数据模型的服务描述相当于基于本体的服务描述。

Wang 等人<sup>[88]</sup>提出了描述物联网服务的本体需要具有的四个性：（1）轻量级；（2）完备性；（3）兼容性；（4）原子性，并对 OWL-S 本体进行改进，提出了“Profile-Model-Grounding”服务描述模型。该模型与以上所述的实体服务描述模型类似，但在以下几个方面进行了扩展：（1）本体中增加了对信息质量（QoI, Quality of Information）、设计和部署阶段用于测试物端资源提供的功能性和非功能性属性的服务测试（Service Test）的描述；（2）允许模型与外部本体、知识库或者现有的链接数据连接起来，并通过分布式存储提高物联网服务访问的可扩展。此外，Linux 基金会也在支持一个名为 IoTivity 的项目<sup>9</sup>，它在 uPnP 社区的基础上提出了 IoT 的描述规范，并为各种物端资源接入物联网提供统一的 API，以保证物端资源提供的服务之间无缝互联。

可见，目前提出的实体服务描述方法都较好地考虑了实体服务的异构性和情境感知性，为物联网中服务的统一注册和管理提供了很好的基础。

#### 4.3 服务注册管理

在物联网环境中，为使用户能够准确快速地获得所需的实体服务和云服务，需要对服务的描述信息进行统一的组织与管理。表 3 总结了目前物联网环境中所采用的实体服务与云服务注册管理机制。最简单的方法是采用集中式注册中心的方式进行管理，比如 Fortino 等人<sup>[87]</sup>采用元数据模型对以智能体形式建立的实体服务进行描述，并基于 JADE（JAVA Agent Development Framework）开发框架，提出了一种集中式的服务索引机制；Botts<sup>[21]</sup>和 Sheth<sup>[22]</sup>分别采用 SensorML<sup>[89]</sup>和 OWL 语言对以

<sup>9</sup> IoTivity. <https://www.iotivity.org/>, 2010.



SOAP 风格实现的实体 Web 服务进行描述,并通过目录服务 (Catalogue Service) 来管理实体服务。

由于注册中心的唯一性,这种集中式服务注册管理方法不具有良好的可扩展性,而且一旦注册中心失效,整个物联网系统将无法工作,因此考虑到实体服务的大规模性,有研究人员将 P2P 网络中管理文件的方法用于组织管理物联网中的服务<sup>[90]</sup>。一种方法是让物联网中的资源在本地通告各自提供的服务,当用户需要访问某个服务时,以洪泛 (Flooding) 或广播方式向物联网中的所有资源发送服务请求,如 Yap 等人基于该方法建立的一个 RFID 物品搜索系统 MAX<sup>[91]</sup>。可见该方法维护服务注册信息的代价较低,但是发现服务的网络通信开销较高。为降低服务发现的网络开销,另一种方法,如 Liu 等人提出的将 Push 和 Pull 结合起来的 service discovery model “comb-needle”<sup>[92]</sup>,是让物联网中的每个资源向周围邻居节点广播服务注册信息 (<服务描述, 服务位置>), 邻居节点根据收到的服务注册信

息广播包后,建立以“服务描述”信息为“目标地址”的路由表。这样当用户需要访问某个服务时,将基于服务描述逻辑推理 (Description Logics Reasoner) 把服务请求发送给物联网中提供所需服务的物端或云端资源,降低了服务发现的网络开销。

以上这两种分布式服务注册管理方法虽然解决了注册中心的唯一性带来的物联网应用系统的可靠性问题,但是这两种方法都需要采用广播包来实现服务的注册管理或服务的发现。由于实体服务具有动态可用性和情境感知性,采用这样的分布式注册管理机制会带来极大的更新维护开销,因此仍然会对物联网应用系统的可扩展性和可维护性造成一定的影响。为此, Evdokimov 等人<sup>[93]</sup>采用分布式哈希表提出了一种结构化分布式方法 DHT-P2P 来管理物联网中的服务,简要来讲,该方法采用一致性哈希 (Consistent Hashing) 确定物理实体服务和云服务的注册信息的存放位置,注册信息可以用一个二元组来表示,即<服务描述, 服务

表 3 物联网实体服务与云服务的注册管理机制

| 服务注册管理 |      |                                 | 服务发现           |
|--------|------|---------------------------------|----------------|
| 集中式    |      | 唯一注册中心                          | 基于目录的服务查找      |
| 分布式    | 非结构化 | 不广播服务注册信息                       | 基于洪泛服务请求       |
|        |      | 广播服务注册信息并建立以“服务描述”信息为“目标地址”的路由表 | 基于路由服务请求       |
|        | 结构化  | 分布式哈希表                          | 基于一致性哈希的服务查找   |
| 混合式    |      | 多个注册中心                          | 基于多个目录的分布式服务发现 |

位置>, 其中服务描述是进行一致性哈希操作的键值, 服务位置是说明提供该服务的物理资源或云端资源的 IP 地址和端口地址。由于分布式结构在可靠性方面的优点, 加上一致性哈希在负载均衡和维护代价方面的优点, 基于 DHT 的分布式结构化服务注册管理方法比以上方法具有更高的可靠性、可扩展性和可维护性。

另外, 还有一种混合式管理服务注册信息的方法, 就是在系统中设立多个服务注册中心, 服务提供者将它们的描述信息发布到对应的注册中心, 注册中心之间以非结构化或结构化分布式结构进行组织管理<sup>[94]</sup>。当某个注册中心收到服务发现请求时, 首先查找本地服务目录是否有匹配的服务, 若存在则直接返回满足要求的服务, 否则通过分布式服务发现的方法, 将服务查询请求转发给其它注册中心。这样既可以避免每个物端和云端资源成为服务管理者, 从而降低整个物联网应用系统的实现难度,

又能保持系统的可靠性、可扩展性和可维护性。

#### 4.4 服务发现

服务发现与服务描述与服务注册管理密切相关。具体来讲:

(1) 根据服务描述方法的不同, 可以将服务发现机制分为基于语法匹配 (Syntactic Matching)、基于语义匹配 (Semantic Matching) 和混合匹配三种类型。语法匹配又包括字符串匹配和相似度匹配, 语义匹配又包括情境感知的语义匹配和非情境感知的语义匹配。基于 3.2 节中所述的基于本体的服务描述方法 (如图 7 和图 8 所示), 目前提出了几种基于情境感知语义匹配的服务发现机制<sup>[85][95]</sup>。在文献<sup>[85]</sup>中作者定义了精确匹配、包含匹配和弱匹配三种服务发现结果, 基于 EASY-L 提出了服务匹配方法 (EASY-M), 并将 EASY-L 和 EASY-M 用于扩展现有的服务发现机制, 使服务发现机制更加具

有智能性；Yang 等人<sup>[95]</sup>将用户的服务请求和实体服务都抽象为包含属性和操作的本体，提出了基于本体的服务发现机制，包括以下步骤：1）请求分解：将用户的服务请求本体分解为层次化的若干子本体；2）子查询管理：基于子本体创建子查询；3）感知服务发现：基于子查询获得相应的感知服务；4）有效性评价管理：评价查询结果与用户请求的匹配效果。

（2）根据服务注册管理机制的不同，可以将服务发现机制分为基于目录的服务查找、基于洪泛服务请求的服务发现、基于路由服务请求的服务发现，以及基于一致性哈希的服务查找（见表 3）。如 Bluetooth SDP、JINI 和 UPnP 是基于集中式服务注册和目录查找的服务发现机制，mDSN<sup>[96]</sup>和 GSD<sup>[97]</sup>不需要集中式服务注册目录，而是采用局域广播和带约束地洪泛服务请求的方法发现服务，SLP 综合了以上这两种服务发现机制的原理，既可以基于集中式服务注册目录，也可以采用洪泛服务请求的方法来发现服务。这些服务发现机制目前都是基于语法匹配的方法为用户精确地查找所需的服务。参照文献[95]，若这些服务发现机制采用基于语义匹配的方法，可以在服务请求传递过程中通过描述逻辑推理获得本体之间的关系，如归类关系(subsumption)，使本体中的概念之间构成分类化的层次结构，从而提高服务发现的质量。

针对实体服务的异构性、大规模性、情境感知性和资源受限性，魏强等人<sup>[98]</sup>总结了目前提出的物联网环境中实现可扩展、高效快速的服务发现方法的几种解决思路。比如，Snoogle<sup>[99]</sup>采用分层(Object Sensor/IP/KeyIP)的动态索引方案，并使用 Bloom Filter 对服务描述进行压缩存储，服务发现时使用一种用于信息检索与数据挖掘的常用加权技术 TF-IDF(Term Frequency-Inverse Document Frequency)进行搜索；Zhang 等人<sup>[100]</sup>建议采用基于上下文的搜索机制来减小搜索空间，从而减小搜索响应时间，并节省资源受限设备的能量；Teixeira 等人<sup>[101]</sup>建议采用概率发现的方法来寻找满足请求的近似服务集，以加速服务发现和减小资源消耗；还有其它物联网环境中降低服务搜索空间的办法，比如预先根据服务的功能、物理位置和移动模式的相似性等特征通过聚类算法进行服务分类，或者根据服务请求上下文（如用户偏好、地理位置、可获取时间等）对匹配的服务数量进行快速过滤，以缩小搜索空间，例如在文献[102]中，作者先通过采用

集合概率主题模型 LDA (Latent Dirichlet Allocation) 缩小搜索服务的范围，之后再使用基于逻辑的方式进行匹配；在文献[103]中，作者和非参数贝叶斯主题模型 HDP (Hierarchical Dirichlet Process) 有效地提取物联网服务文本描述中的隐含主题，用于对服务进行自动分类，并通过主题相似度匹配快速减小搜索空间，提高物联网服务发现的实时响应率，节约资源受限设备的能量；同时，他们采用逻辑签名 (Signature) 匹配方法支持异构服务 (WSDL-based 和 RESTful 两种物联网服务) 的发现。Elahi 等人<sup>[104]</sup>使用预测技术以捕获实体服务周期性的行为，将物端资源上可能的状态存储在页面上，查询时先匹配静态信息，之后根据模型预测实体服务的动态状态，计算得到一个排名最高的列表之后，再与请求真实值一一比对，以减小搜索空间、提高搜索响应效率。本研究组在考虑实体服务的动态可用性，提出了一种高效的实体服务可获取性评估与预测方法，以提高物联网服务的发现效率<sup>[105]</sup>。

需要指出的是，以上所述的物理实体服务的注册管理与服务发现所采用的服务描述一般是由物端资源的所有者在部署时主动完成的，且服务描述不含有资源的状态信息。此外，德国吕贝克大学的科研人员还提出了一种无需注册、基于资源状态信息模式匹配、自动生成服务描述的方法<sup>[106]</sup>，以及相应的基于资源状态信息的服务搜索机制<sup>[107]</sup>。还有，美国弗吉尼亚大学的研究人员基于传感器类型和数值特性主动学习的方法也实现了服务描述的自动生成<sup>[108]</sup>，以及相应的基于 Back-Chaining 推理方法的物联网事件服务搜索平台<sup>[109]</sup>。虽然这些方法对实体服务的情境感知性考虑得更为周密，但是在大规模和资源性受限的物联网环境中，该方法的可扩展性和服务发现响应率和速度都还有待进一步验证。

从以上对服务发现机制的分析中可以看出，服务发现可能返回多个与用户请求匹配的结果，因此需要从中选择最符合需求的一个或几个服务。对于服务选择，目前一般采用人工选择或基于用户请求的自动选择两种方法<sup>[110][111]</sup>。自动选择方法主要考虑实体服务的动态可用性、情境感知性和资源受限性，基于服务质量信息 (QoS Information, 如图 7 中所示)，通过构建 QoS 评估模型，动态地评价实体服务的服务距离、响应延迟、可获取性和可靠性等，快速定位满足服务质量要求的实体服务。

不管采用哪种方法最终确定了所需的服务，都

需要将这些服务组合起来构建可运行的物联网应用系统,以完成指定的应用任务。

#### 4.5 服务组合与编程接口

服务组合 (Service Composition) 又被称为服务集成 (Service Integration) 和任务编排 (Task Orchestration),它是根据需求处理执行计划将发现的服务组合起来以完成指定的应用任务,其中需求处理执行计划是由系统开发者或用户通过以下形式的编程接口定义的:

(1) 以高层次抽象的编程语言 (PL, Programming Language) 或可视化编程方式定义处理树 (Processing Tree) 或规则树 (Rule Tree),说明任务执行需要的服务和任务之间的调用关系;

(2) 以规划领域定义语言 (PDDL, Planning Domain Definition Language) 或可视化建模方式说明业务流 (Workflow),通过映射机制确定需要的服务和相应的业务流模型。

因此,根据编程接口的形式不同,可以将目前物联网环境中定义服务组合关系的形式分为以下两种:

(1) 以处理树或规则树的形式定义服务组合关系,如 Kovatsch 等人<sup>[112]</sup>基于现有的 Web Mashup 方法提出了一种实体服务组合方法。该方法采用 Clickscript 编辑器 (Mashup Editor) 定义物联网任务相关的实体服务和计算模块之间的调用关系,并在一个专门的服务器 (Mashup Server) 上实现服务的组合和任务的执行。

(2) 以业务流模型的形式定义服务组合关系,如 Huang 等人<sup>[113]</sup>提出了一种基于图规划 (Graph Plan) 的服务组合方法,该方法考虑了实体服务的情境感知性和资源受限性,又考虑了实体服务的动态可用性,采用 PDDL 给出任务的需求、初始状态和目标状态,不仅初始状态和目标动态的,而且规划领域也是根据任务需求动态生成的。

可见,采用第一种编程接口定义的服务组合关系比较精确,但不能很好地适应物联网应用任务的动态变化;第二种定义服务组合关系的方法具有一定的智能性,在任务需求易变的物联网环境中,无需重新编程就能自动重组服务。比如,基于图规划定义的动态需求,Huang 等人<sup>[113]</sup>提出了一种基于智能规划 (AI Planning) 的动态自适应服务组合方法,根据形式化逻辑模型找出相关的实体服务,自动确定业务流模型并将找到的相关实体服务组合起来以完成指定的物联网应用任务。湖南大学的研

究人员<sup>[114]</sup>也考虑到实体服务的动态性,提出了一种语义驱动的服务组合算法,并基于传统的多维 QoS 模式,设计了一种快速的服务路径优化选择算法,以及失效服务重定向机制。

另外,依据服务发现和服务组合的过程是在任务执行时还是处理树或业务流模型解析时进行,可以将物联网环境中的服务组合分为运行时组合和运行前组合两种类型。

(1) 运行时组合:根据用户通过编程接口创建的业务流模型,组合服务执行引擎借助服务模版,为业务流程中的每一个环节动态选择和运行时绑定实体服务或云服务,并根据业务流程中的数据流设置服务之间的参数传递和参数映射,从而形成满足任务需要的组合服务。如 SOCRADES<sup>[37]</sup>中的服务组合采用的就是这种运行时组合的方式。

(2) 运行前组合:每次用户提交物联网应用任务,编程接口模块就即时解析处理树或业务流模型,即时调用服务发现模块确定完成任务需要的实体服务和云服务,完成服务组装,以满足物联网应用任务的需求。如 Physicalnet<sup>[34]</sup>中包含的服务解析模块和编程抽象模型管理模块实现的就是这种运行前组合的方式。

## 5 现有物联网服务中间件的分类比较

基于以上对目前设计与实现物联网服务中间件各模块所提出的方法总结,本节将对现有的物联网服务中间件进行分类比较。根据所面向的实体服务的类型不同,本文将物联网服务中间件分为两类,如图 9 所示,一类是面向物理实体服务的物联网服务中间件,它们将由物端资源直接提供的实体服务作为物联网系统的主要组成模块,为构建以“物端”为中心的物联网系统提供支持;另一类是面向虚拟实体服务的物联网服务中间件,它们将物端资源提供的感知数据和控制接口封装成云端设备上的实体服务,物联网服务中间件各功能模块都以这些实



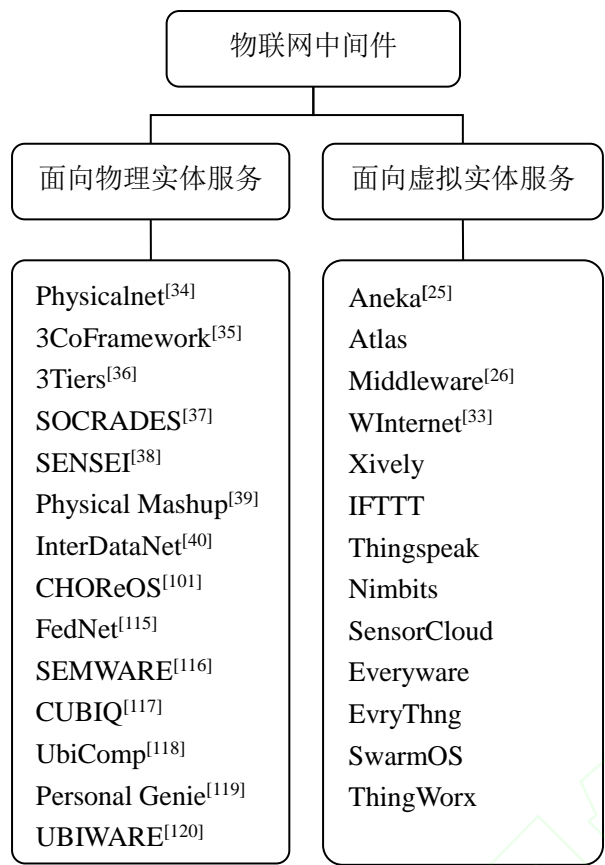


图 9 物联网服务中间件分类

体服务为对象，进行管理与操作，为构建以“云端”为中心的物联网系统提供良好的支撑软件平台。不管是哪种类型的物联网服务中间件，目前大都是将云服务支撑平台作为实现和应用中间件的平台，这样可以避免物端嵌入式平台的资源受限性，能够支持大规模集中式、分布式或混合式的服务管理、应用开发与运行。

5.1 面向物理实体服务的物联网服务中间件

该类物联网服务中间件面向的物理服务是建立在感知/执行设备或网关上的物理实体服务，它们所面临的挑战基本上就是表 1 所示。根据以上分析得出的物联网服务中间件各模块的设计与实现方法，表 4 对目前各国研究机构已经建立的面向物理实体服务的物联网服务中间件进行了比较。

首先，从表 4 可见这些物联网服务中间件所面向的物理实体服务采用不同的软件架构实现，Physicalnet、3CoFramework、3Tiers、CHOReOS、FedNet、UbiComp 和 Personal Genie 这 7 个都是基于分布式对象的软件架构实现物理实体服务；SOCRADES、SEMWARE 和 CUBIQ 采用 SOAP 风

表 4 面向物理实体服务的物联网服务中间件比较

| 中间件                             | 物理实体<br>服务提供者 | 实体服务<br>软件架构           | 服务<br>描述 | 服务<br>注册管理 | 服务发现          | 服务<br>组合 | 编程接口          |
|---------------------------------|---------------|------------------------|----------|------------|---------------|----------|---------------|
| Physicalnet <sup>[34]</sup>     | 感知设备 /<br>网关  | 分布式<br>对象              | 语法       | 集中式        | 基于目录的<br>服务查找 | 运行前      | 处理树<br>(PL)   |
| 3CoFramework <sup>[35]</sup>    | 感知设备 /<br>网关  | 分布式<br>对象              | 语法       | 集中式        | 基于目录的<br>服务查找 | 运行前      | 处理树<br>(PL)   |
| 3Tiers <sup>[36]</sup>          | 感知设备 /<br>网关  | 分布式<br>对象              | 语法       | 集中式        | 基于目录的<br>服务查找 | 运行时      | 业务流<br>(PDDL) |
| SOCRADES <sup>[37]</sup>        | 感知设备          | SOAP 风<br>格的 Web<br>服务 | 语法       | 集中式        | 基于目录的<br>服务查找 | 运行时      | 业务流（可<br>视化）  |
| SENSEI <sup>[38]</sup>          | 感知 / 执行<br>设备 | REST 风<br>格的 Web<br>服务 | 语法       | 集中式        | 基于目录的<br>服务查找 | 运行时      | 业务流<br>(PDDL) |
| Physical Mashup <sup>[39]</sup> | 网关            | REST 风<br>格的 Web<br>服务 | 语法       | 集中式        | 基于目录的<br>服务查找 | 运行前      | 处理树（可<br>视化）  |
| InterDataNet <sup>[40]</sup>    | 感知 / 执行<br>设备 | REST 风<br>格的 Web       | 语法       | 集中式        | 基于目录的<br>服务查找 | 运行前      | 处理树（可<br>视化）  |

|                                 |         | 服务              |    |     |                |     |            |
|---------------------------------|---------|-----------------|----|-----|----------------|-----|------------|
| CHOReOS <sup>[101]</sup>        | 感知/执行设备 | 分布式对象           | 语义 | 集中式 | 基于目录的服务查找      | 运行时 | 业务流 (PDDL) |
| FedNet <sup>[115]</sup>         | 感知/执行设备 | 分布式对象           | 语法 | 集中式 | 基于目录的服务查找      | 运行前 | 处理树 (PL)   |
| SEMWARE <sup>[116]</sup>        | 感知/执行设备 | SOAP 风格的 Web 服务 | 语义 | 集中式 | 基于目录的服务查找      | 运行时 | 业务流 (PDDL) |
| CUBIQ <sup>[117]</sup>          | 感知/执行设备 | SOAP 风格的 Web 服务 | 语义 | 分布式 | 基于一致性哈希的服务查找   | 运行时 | 业务流 (PDDL) |
| UbiComp <sup>[118]</sup>        | 感知/执行设备 | 分布式对象           | 语义 | 分布式 | 基于洪泛服务请求       | 运行前 | 处理树 (PL)   |
| Personal Genie <sup>[119]</sup> | 感知/执行设备 | 分布式对象           | 语法 | 分布式 | 基于洪泛服务请求       | 运行时 | 业务流 (PDDL) |
| UBIWARE <sup>[120]</sup>        | 感知/执行设备 | 智能体             | 语义 | 混合式 | 基于多个目录的分布式服务发现 | 运行时 | 业务流 (PDDL) |

注：最后一列所列的编程接口包括用户定义应用任务的形式（处理树或业务流模型）和方式（PL 或 PDDL 或可视化）

格的 Web 服务构建物理实体服务，而 SENSEI、Physical Mashup、InterDataNet、采用 REST 风格的 Web 服务构建物理实体服务；还有 UBIWARE 是基于智能体的软件架构实现物理实体服务。第二，从服务描述的方法来看，除了 CHOReOS、SEMWARE、CUBIQ、UbiComp 和 UBIWARE 这 5 个采用基于语义的描述方法外，其它中间件对实体服务的描述仅限于语法级别的描述；第三，从服务注册管理和服务发现的实现方法来看，除了 CUBIQ、UbiComp、Personal Genie 和 UBIWARE 这 4 个采用分布式或混合式的服务注册管理和对应的服务发现方法外，其它都采用集中式的服务注册和基于目录的服务查找方法；第四，从服务组合和编程接口的形式来看，Physicalnet、3CoFramework、Physical Mashup、InterDataNet、FedNet 和 UbiComp 以处理树的形式定义应用的任务需求，并在运行前进行服务组合，其它 8 个中间件以业务流模型的形式定义任务，并在运行时进行服务组合。因此，总体来看 UBIWARE 最符合表 1 列出的物联网服务中间件的功能要求。下面对这些中间件的具体实现方法进行简要分析。

(1) Physicalnet<sup>[34]</sup>是由美国弗吉尼亚大学的研究人员提出的一种物联网软件体系结构，其中包含了一种面向物理实体服务的物联网服务中间件，它所面向的物理实体服务基于分布式对象架构实现，

包含服务仓库、访问控制表、需求表、服务解析模块和编程抽象模型管理模块，其中服务仓库、访问控制表和需求表是管理感知服务和网关服务，实现多个应用程序在同一资源上或跨网络和管理域并发运行的支撑模块；服务解析模块和编程抽象模型管理模块是实现物联网系统的编程接口模块。Physicalnet 提供的编程口基于 Bundle 构建策略，因此属于处理树的形式。

(2) 3CoFramework<sup>[35]</sup>是美国内布拉斯加林肯大学的研究人员提出的一种物联网软件体系结构，它将物联网中的服务分为四层，即数据层、信息层、知识层和表达层，其中数据层服务是指物联网中的物理实体服务，其它三层服务为虚拟实体服务和云服务，这些服务组件 (Component) 通过 NRM (Node Resource Manager) 向唯一的协调器 (Coordinator) 注册非功能信息，并在协调器的管理下通过连接件 (Connector) 连接起来，支持物联网系统的构建。因此，3CoFramework 定义的协调器和连接件被看作是它所定义的物联网服务中间件的功能模块，它所面向的服务包含物理实体服务。3CoFramework 基于 COBRA 编程模型，通过表达层服务组件为用户提供基于处理树的编程接口，支持运行前组合。

(3) 3Tiers<sup>[36]</sup>是韩国崇实大学的研究人员提出的一种以物端为中心的物联网软件体系结构，它定

义了物理组件及物理组件监测模块和配置模块、云服务组件及云服务组件监测模块和接口适配模块、服务注册中心、服务查找模块、服务组合模块，以及服务失效管理模块，其中物理组件和云服务组件分别是位于物联网前端的物理实体服务和云端的云服务，除了物理组件与云服务组件，其它模块构成了 3Tiers 的中间件。它采用注册中心实现了这两类服务的集中式管理，并实现了以下功能：1) 监视物理实体服务和云服务；2) 服务动态查找和服务组合；3) 服务接口适配与变异；4) 服务失效自主管理。因此，3Tiers 支持运行时服务组合和基于业务流模型的编程。

(4) SOCRADES<sup>[37]</sup> 是由德国 SAP 公司研究院和瑞士苏黎世联邦理工大学的研究人员提出的一种将支持 Web 服务的设备（如 DPWS）与企业应用平台（如 ERP）集成起来的中间件，主要实现设备管理与监测、服务发现、服务生命周期管理、跨层服务目录和安全支持等功能。该中间件面向的设备服务和企业应用平台服务都是以 SOAP 风格的 Web 服务实现的，因此采用 WSDL 语言描述这些服务，采用集中式注册中心的服务管理和基于目录的服务查找，服务组合与提供的编程接口都是基于业务逻辑模型。

(5) SENSEI<sup>[38]</sup> 是欧盟 FP7 项目提出的一种将物理世界中各类感知和执行设备集成到数字世界的物联网软件体系结构，它包括无线传感与执行网络（WS&AN）资源、资源目录、资源查找模块、资源创建模块、资源汇聚模块、资源查询解析模块和感执任务创建模块，其中 WS&AN 资源表示物联网的实体服务，资源目录和资源发现、创建、汇聚模块构成物联网服务中间件，用于管理实体服务并为构建物联网系统提供支撑服务，资源查询解析模块和感执任务创建模块是提供给系统开发者的编程接口。由于 SENSEI 中间件含有资源创建和汇聚模块，因此它可以支持实体服务的运行时组合，并通过资源查询解析模块支持基于业务流模型的任务需求规划编程。

(6) Physical Mashup<sup>[39]</sup> 是由瑞士苏黎世联邦理工大学的研究人员提出的一种面向 REST 风格的物理实体服务的物联网软件体系结构，它所定义的用于管理建立在智能网关上的轻量级 Web 服务和云端 Web 服务的事件中心和物理聚合模块就构成了物联网服务中间件，并提供了图形化的用户编程接口 Clickscript，支持以处理树的形式定义用户的

任务需求，并在任务执行之前完成服务的聚合。

(7) InterDataNet<sup>[40]</sup> 是由意大利佛罗伦萨大学的研究人员提出的一种面向 REST 风格的实体服务的物联网服务中间件，与图 6 所示的基于 REST 风格的 Web 服务实现的实体服务不同的是，它所面向的实体服务除了最上层是表达建立在感知/执行设备上的实体服务的接口外，还包括管理有关该资源的历史服务信息和存储数据的接口。该中间件采用 Apache Solr 搜索服务来管理物联网中的实体服务。与 Physical Mashup 不同的是，该中间件支持实体服务之间直接互联，不需要通过一个集中的聚合模块，因此它包括一个服务适配模块，用于在服务组合时将来自不同实体服务的请求和数据进行转化。此外，它提供了一个图形化的编程环境 IDN-Studio，采用处理树的形式来定义用户的任务。

(8) CHOReOS<sup>[101]</sup> 是由法国 INRIA 研究所的研究人员提出的一种语义物联网的中间件，实现了节点选择、服务组合、数据估计和更新元数据等功能。它以集中式注册中心的方式管理服务，并提出了概率性查找（Probabilistic Lookup）服务的概念，服务组合所基于的模版被称为数据流（Dataflow），它与业务流（Workflow）的概念是一样的，因此该中间件支持基于业务流模型的编程方式和运行时服务组合。

(9) FedNet<sup>[115]</sup> 是由英国兰卡斯特大学的研究人员提出的另一种面向分布式对象的物联网服务中间件，它使物联网中的资源以 Profile 形式定义提供的服务，每个资源可以定义多个 Profile，该中间件负责管理这些 Profile，并将需要的 Profile 互联起来实现指定的物联网任务。它所采用的管理 Profile 的方法基于集中式注册中心，提供的编程接口是基于文档的任务定义模式，该模式与定义处理树具有一定的相似性。

(10) SEMWARE<sup>[116]</sup> 是由富士通美国研究院的研究人员提出的一个物联网任务计算环境（Task Computing Environment）中的中间件，它采用 OWL-S 语言来描述局域智能系统提供的服务，包含一个语义服务描述模版库（知识仓库）来作为语义发现的基础，实现基于语义的分解任务、映射服务和组合运行等功能。SEMWARE 采用集中式的服务注册管理方法，以及基于目录的服务查找机制，并且提供一个基于语义的任务模版（即业务流模型）定义方法，来确定服务之间进行组合的关系。

(11) CUBIQ<sup>[117]</sup> 是日本 NEC 公司提出的一种



旨在将现有异构的泛在服务集成起来,提供统一的数据访问、处理与融合的中间件。该中间件面向的物端资源提供透明地数据访问、移动管理、数据迁移、复制、一致性、容错性与永久性保证等服务,以及基于情景的资源内业务处理服务和资源间业务处理服务。它的服务描述采用通用服务描述语言(USDL),服务组织与发现采用开放的结构化 P2P 平台 PIAX,服务组合和提供的编程接口采用基于业务流模型的方法。

(12) UbiComp<sup>[118]</sup>是由希腊开放大学的研究人员提出的一种面向分布式对象的物联网服务中间件,主要包括基于语义的服务发现、动态绑定机制和基于 Plug/Synapse 的应用程序编程接口,其中服务源发现采用基于洪泛服务请求的方法,编程接口提供的定义应用程序的方法是基于处理树。该中间件使用 J2ME CLDC (Connected Limited Device Configure)框架进行了实现。

(13) Personal Genie<sup>[119]</sup>是韩国科学技术院的研究人员提出的一个基于智能手机的情境驱动面向任务的自发式服务中间件,它实现了智能物品发现、选择性情境获取、状态推理和任务推荐、任务组合以及组合任务执行等功能,其中物品发现和组合任务执行都没有集中式的服务器,而是采用分布式的方法,而且服务组合时还考虑了实体服务的服务质量约束,因此该中间件能够更好地基于智能物

品构建物联网系统。

(14) UBIWARE<sup>[120]</sup>是由芬兰于韦斯屈莱大学 Agora 中心的研究人员提出的一种基于智能体的物联网服务中间件。它将物联网中的资源以智能体的形式实现,每个智能体由一组可复用的原子行为(RABs, Reusable Atomic Behaviors)构成,并用 RDF 描述智能体的行为,可见它采用的是基于语义的服务描述方法。UBIWARE 采用多个目录(DF, Directory Facilitator)来组织管理智能体,并以 P2P 的方式实现服务的发现,并且提供了基于业务流程的应用任务配置和服务组合的接口。

需要说明的是,以上所述的(11)、(12)和(13)从严格意义上讲,并不能算是物联网服务中间件,而是构建普适计算环境的中间件,因为它们都是为实现局域范围内的智能物品之间的智能互联与互操作提出的。但是,这些中间件可以扩展到基于互联网的计算环境中,因此本文也将它们作为可用于支撑物联网应用系统构建的服务中间件。

## 5.2 面向虚拟实体服务的物联网服务中间件

该类物联网服务中间件面向的实体服务是以 PaaS、IaaS 或 SaaS 的形式建立在云端计算平台上的虚拟实体服务,因此它们不需要考虑资源受限性带来的挑战,各模块的设计与实现比上述物联网服

表 5 面向虚拟实体服务的物联网服务中间件比较

| 中间件              | 云平台类型  | 虚拟实体服务类型        | 服务描述 | 服务注册管理 | 服务发现      | 服务组合 | 编程接口        |
|------------------|--------|-----------------|------|--------|-----------|------|-------------|
| Aneka            | PaaS   | REST 风格的 Web 服务 | 语义   | 集中式    | 基于目录的服务查找 | 运行时  | 编程逻辑(PL)    |
| Atlas Middleware | PaaS   | 分布式对象           | 语义   | 集中式    | 基于目录的服务查找 | 运行时  | 规则树(PDDL)   |
| WInternet        | Server | 分布式对象           | 语法   | 集中式    | 基于目录的服务查找 | 运行前  | 编程逻辑(可视化)   |
| Xively           | SaaS   | REST 风格的 Web 服务 | 语义   | 集中式    | 基于目录的服务查找 | 运行前  | 业务流(PL/SDK) |
| IFTTT            | SaaS   | REST 风格的 Web 服务 | 语义   | 集中式    | 基于目录的服务查找 | 运行前  | 规则树(可视化)    |
| Thingspeak       | PaaS   | REST 风格         | 语法   | 无      | 基于 URI    | 运行前  | 规则树         |

|             |      | 的 Web 服务        |    |     | 的访问        |     | (PL/可视化)      |
|-------------|------|-----------------|----|-----|------------|-----|---------------|
| Nimbits     | PaaS | REST 风格的 Web 服务 | 语法 | 无   | 基于 URI 的访问 | 运行前 | 编程逻辑 (PL)     |
| SensorCloud | PaaS | REST 风格的 Web 服务 | 语义 | 无   | 基于 URI 的访问 | 运行前 | 编程逻辑 (PL/可视化) |
| Everyware   | PaaS | REST 风格的 Web 服务 | 语义 | 集中式 | 基于目录的服务查找  | 运行前 | 规则树 (可视化)     |
| EvryThng    | PaaS | REST 风格的 Web 服务 | 语义 | 集中式 | 基于目录的服务查找  | 运行前 | 规则树 (可视化)     |
| SwarmOS     | SaaS | SOAP 风格的 Web 服务 | 语义 | 集中式 | 基于目录的服务查找  | 运行前 | 业务流 (PL/可视化)  |
| ThingWorx   | PaaS | 智能体             | 语义 | 集中式 | 基于目录的服务查找  | 运行前 | 业务流 (可视化)     |

务中间件要简单些。表 5 对目前各国研究机构已经建立的面向虚拟实体服务的物联网服务中间件进行了比较。从表 5 可见, 首先目前这类物联网服务中间件采用的云平台类型大多为 PaaS, 而 3 个主流的物联网中间件 (Xively、IFTTT 和 SwarmOS) 基于 SaaS 类型的云平台, 只有 WInternet 以最基础的服务器形式提供建立虚拟实体服务的平台; 其次, 它们面向的虚拟实体服务大部分都是以 Web 服务的形式构建的, 尤其是以 REST 风格的 Web 服务为主, 包括 Aneka、Xively、IFTTT、Thingspeak、Nimbits、SensorCloud、Everyware、EvryThng; 第三, 它们 (表 5 中所列除 WInternet、Thingspeak 和 Nimbits 以外的中间件) 大都从语义的层次来对服务进行描述, 以提高服务发现和搜索的智能性, 但是由于这些中间件都采用集中式的服务注册管理, 因此所采用的服务发现机制也都是基于目录的服务查找。对于 Thingspeak、Nimbits 和 SensorCloud 来讲, 由于它们缺少服务的注册管理, 所以要求上层应用获知服务的 URI 以对其进行访问; 第四, 它们所提供的编程接口主要以编程语言或可视化方式定义应用任务的业务逻辑或规则树为主, 只有 Xively、SwarmOS 和 ThingWorx 提供的编程接口能够以业务流模型的形式说明应用任务的需求。因此, 总体而言 ThingWorx 最符合表 1 列出的物联网服务中间件的功能要求, 但是由于它以智能体的形式构建虚拟实体服务, 实现起来具有较高的复杂度, 在专

门的行业领域有一定的应用; Xively、IFTTT、Everyware 等以 REST 风格的 Web 服务为软件架构实现虚拟实体服务, 且提供的编程接口具有较高的用户友好性, 在通用性和用户数量上具有一定的优势。下面对这些物联网服务中间件进行简要分析。

(1) Aneka<sup>[25]</sup>是澳大利亚墨尔本大学的研究人员采用混合云的方式而设计的一个云服务平台。在该平台之上, 除了为提供 PaaS 云服务所需的系统虚拟化模块和资源监测模块外, 还为构建以“云端”为中心的物联网系统而设计了一个面向 REST 风格的虚拟实体服务的物联网服务中间件, 包括虚拟实体服务提供 (Provisioning)、服务调度 (Scheduling)、服务计费 (Billing)、运行时 (Runtime) 服务和多种编程架构的编程接口。由于 Aneka 上的虚拟实体服务主要是对物端设备提供的数据进行抽象构建而成的, 所以采用数据库的形式对它们进行统一管理, 并提供查询服务。用户通过编程接口以任务、线程或其它编程架构来操作虚拟实体服务提供的数据, 并可以通过服务组合的方式对它们进行集成。

(2) Atlas Middleware<sup>[26]</sup>是由美国佛罗里达大学的研究人员为构建以“云端”为中心的物联网系统而设计的物联网服务中间件, 由分别位于云端 (Cloud)、网关边界 (Edge) 和感知平台 (Sensor Platform) 上的中间件组成, 其中云端中间件面向虚拟实体服务 (在该文中被称为云传感器和虚拟传

传感器),包括边界连接模块、实用服务(Utility Service)管理模块、服务注册管理模块、服务提供管理模块、服务查找模块、服务组合模块和应用开发模块。该中间件所面向的虚拟实体服务以 E-SODA (将感知数据抽象为事件 Event,对 SODA<sup>[67]</sup>进行扩展后的服务架构)为实现架构,整体采用 OSGi 基础架构,并且支持基于规则(Rule)的编程模式。基于规则树的建立和服务查找模块的支持,该中间件可以实现虚拟实体服务之间的动态集成,并且该中间件还支持服务组合后的系统优化。

(3) WInternet<sup>[33]</sup>是由澳门大学的研究人员提出的一种实现各类感知控制设备互联的物联网架构,它主要由建立在互联网层之上的物联网服务层组成。物联网服务层将来自边界网络的感知数据按照标准的 WInternet 协议封装成软件功能模块(称之为网件 Netlet),通过管道协议(PIPE)在网件的阀口之间建立连接,以此实现感知控制设备的互联,形成满足用户需求的物联网应用系统。可见,WInternet 中间件将网件作为虚拟实体服务,包括网件接口注册管理模块、网件搜索模块和网件连接模块,满足物联网服务中间件的基本功能需求。但是,在编程接口和自动服务组合方面,其功能正在进一步的扩展。

(4) Xively<sup>10</sup>(最初的名字为 Pachub<sup>[24]</sup>)是由美国 LogMeIn 公司在其云平台 Gravity 上建立的物联网云服务平台,它采用 SaaS 云服务模式,即将各类物品的信息接入到云平台中,封装成虚拟实体服务(称之为 Blueprint),并通过云平台上的 Blueprint 数据库、服务管理模块、服务集成模块和应用开发工具包(SDK),来提供通用的物联网应用开发平台。可见,Xively Blueprint 是采用 REST 风格的 Web 服务架构将设备、用户、组织机构和频道(Channel)这四类物理世界中的实体在云平台上建模而成的虚拟实体服务,其它模块就构成了 Xively 的中间件,提供给用户的编程接口就是 SDK,包括开发个人计算平台和智能手机(iOS 和 Android)上物联网应用系统的 SDK。

(5) IFTTT<sup>11</sup>是一个通过设置触发条件(If This Then That)实现不同的网络服务平台之间的信息互联与互操作,如设置这样的一个任务“If ‘watering time stops’ Then ‘send me an email’”,表示当浇水结束时发送一份邮件给用户,这样便在邮件服务平台

与感知服务平台之间建立了一个连接。基于 IFTTT 平台建立物联网系统的前提是在该平台上建立各种感知和执行设备的信息服务,即 REST 风格的 Web 虚拟实体服务,然后获得需要其他网络服务平台的操作权限,通过设置触发条件建立任务,构建简单的物联网系统。需要说明的是,虽然表 5 中列出了 IFTTT 采用基于规则树的可视化编程接口,但是目前支持的规则树只有一层,即只有一个触发条件。当多个用户对某个虚拟实体服务设置触发条件时,尤其是支持执行操作的虚拟实体服务,可能会产生冲突,为此微软研究院的研究人员提出了一种冲突规则检测机制 SIFT<sup>[121]</sup>,以在服务连接时避免这样的执行冲突。

(6) Thingspeak<sup>12</sup>是一个由 MathWorks 公司为物联网应用的构建设计的通用数据服务平台,各种感知设备通过 HTTP 协议接入到该平台,平台上为每个感知设备建立频道(Channel),每个 Channel 最多包含 8 个维度的数据、位置信息(经度、纬度)和状态信息。这些频道以 REST 风格的 Web 服务建立和提供访问操作接口,因此可以将这些 Channel 看作是 Thingspeak 上的虚拟实体服务。面向这些 Channel,它还提供了 Channel 配置接口、Channel 订阅接口、数据分析工具(平均、求和、取整等)和数据图形化显示工具。这些接口和工具都建立在 Thingspeak 中间件的基础上,对应的中间件模块包括 Channel 管理模块、数据分析模块、数据显示模块、触发事件管理模块,其中数据分析模块可以从多个 Channel 输入数据,通过 Matlab 编程语言定义处理操作,实现虚拟实体服务的组合处理。但是,该中间件目前只支持基于 URI 的服务获取,不支持服务发现。

(7) Numbits<sup>13</sup>是一个与 Thingspeak 类似的物联网应用云服务平台,它可以建立在普通的 J2EE 服务器上,也可以建立在 Google App Engine 和 Amazon EC2 这样的云基础设施上。与 Thingspeak 不同的是,Numbits 将物理感知和执行设备的数据封装成虚拟实体服务,称之为 Data Points,每个 Data Points 可以包含多维感知数据,并在此基础上建立其它服务,包括规则触发器、数据处理器、警报器等。通过该平台提供的 API (JAVA 语言),可以将这些服务连接起来,构成满足用户需求的物联网应用系统。管理这些服务并支持这些服务互连的

<sup>10</sup> Xively. <http://developer.xively.com/architecture/>, 2009.

<sup>11</sup> IFTTT. <https://ifttt.com/>, 2011.

<sup>12</sup> Thingspeak. <https://thingspeak.com/>, 2012.

<sup>13</sup> ThingSquare. <http://nimbis.com>, 2014.



Numbits 中间件由以上各类服务的创建和访问管理模块、业务(transaction)管理模块、图表显示(chart)模块等组成。但是,与 Thingspeak 一样的是,该中间件目前只支持基于 URI 的服务获取,不支持服务发现。

(8) SensorCloud<sup>14</sup>是用于感知数据安全存储的物联网云平台。与 Numbits 类似,该平台上将各类感知数据封装成 REST 风格的 Web 服务,通过 URI 可以访问这些服务,并可以与 Google Maps 服务相结合,提供图形化数据显示;还可以通过该平台上的 MathEngine 对数据进行傅立叶变换、滤波、滑动平均等处理,并且支持用户用 Python 语言自定义处理方法;此外,还支持基于条件触发的事件提醒。可见, SensorCloud 为支持各类物联网应用系统的建立所提供的中间件包括数据接入模块、数据管理模块、数据集成模块、数据分析模块、数据显示模块和事件管理模块。需要指出的是 Thingspeak、Numbits 和 SensorCloud 没有实现服务注册管理和服务发现的功能,因此要求用户构建应用任务时获知所需的实体服务的唯一资源标识 (URI)。

(9) Everyware<sup>15</sup>是由 Eurotech 公司设计的物联网开发平台,包括智能感知设备、多功能网关、云基础设施,以及这些硬件上的操作系统、中间件和编程框架,其中智能感知设备和网关上的中间件为 ESF (Everyware Software Framework),云基础设施 (Everyware Cloud) 上的中间件包括设备连接模块、设备管理模块、数据管理模块、安全管理模块、云平台管理模块和应用集成模块,其中设备管理模块采用 MQTT<sup>16</sup>协议对感知设备进行远程管理,数据管理模块采用复杂的统计规则引擎将来自感知设备的数据进行处理和模式识别,并可以触发相应的动作。来自不同感知设备的数据经过不同的规则处理,形成不同的服务,这些服务就是 Everyware Cloud 之上的虚拟实体服务,它们以 REST 风格的 Web 服务形式提供给用户,通过应用集成模块可支持各类物联网应用的创建。云平台管理模块提供了 Web 界面支持可视化的规则创建和应用集成。

(10) EvryThng<sup>17</sup>是面向各类物联网应用建立的一个智能物品云管理平台,该平台的中间件架构与 Everyware 类似,包括物品连接管理模块、实时

数据管理模块、数据分析模块、集成模块和开发工具模块,其中数据分析模块负责处理来自物端各类感知设备的数据并建立虚拟实体服务,集成模块支持将这些服务进行组合构建各类物联网应用,开发工具模块提供管理和操作虚拟实体服务的 Web 界面,以降低应用的开发难度。EvryThng 还包括安全认证模块,以支持 EvryThng 云平台与其它物联网云平台 (如 Nest 云) 之间的互联。

(11) SwarmOS<sup>18</sup>是由美国加州伯克利大学等研究机构在 TerraSwarm 项目中建立的物联网操作系统,用于将物理环境中的智能网络化的传感器和执行器连接起来,并对这些资源和系统中的其它资源 (如计算资源、存储资源、能量和通信频谱资源) 进行管理,以支持分布式并发物联网应用系统的构建。SwarmOS 的架构借鉴了 SOA 软件设计方法的优点,允许用户以服务连接的方式构建应用,并通过应用任务分析和任务分解模块、服务发现模块和服务组合模块来保证构建的系统具有一定的自适应性,以支持应用任务的重配置。

(12) ThingWorx<sup>19</sup>是由美国 PTC 公司基于 Axeda 云平台建立的面向虚拟实体服务的物联网公共支撑平台,除了云平台本身为提供可扩展、安全的云服务而设计的基础设施功能模块外,它为支持以“云端”为中心的物联网系统而设计的功能模块 (即物联网服务中间件),包括物端设备接入模块、设备管理模块、数据管理模块、应用服务模块、应用逻辑构建模块、脚本执行引擎,以及拖放式编程接口 (“drag and drop” Mashup Builder)。它提供基于 SQUEAL 的交互式虚拟实体服务查询,以提高基于业务流的物联网应用的构建效率。

## 6 总结与展望

通过以上对两类物联网服务中间件的比较分析可知,采用面向服务的软件架构 (SOA),以不同类型的实体服务 (物理实体服务和虚拟实体服务) 作为基础模块,构建不同类型的物联网系统 (以“物端”为中心或以“云端”为中心),需要建立的物联网服务中间件所面临的挑战有所不同,但是综合来讲如表 1 所示。针对这些挑战,目前已经建立了一些物联网服务中间件。从表 4 和表 5 所列的各

<sup>14</sup> SensorCloud. <http://sensorcloud.com/>, 2014.

<sup>15</sup> Everyware. <http://www.eurotech.com/en/products/software+services>, 2010.

<sup>16</sup> MQTT. <http://mqtt.org/documentation>, 2010.

<sup>17</sup> EvryThing. <https://evrythng.com/>, 2014.

<sup>18</sup> SwarmOS. <https://www.terraswarm.org/platforms/>, 2013.

<sup>19</sup> Thingworx. <http://www.ptc.com/internet-of-things/technology-platform-thingworx>, 2013.

物联网服务中间件在服务描述、服务注册管理、服务发现,以及服务组合与任务编程接口的设计与实现方面的异同来看,未来物联网服务中间件会更多考虑跨行业应用的通用支撑服务及同一个运行时平台支持多个应用的功能,从而会向着物联网通用中间件系统的方向发展;同时也会着面向不同行业或不同应用进行扩展,向着物联网专用中间件系统的方向发展。在这种“通用+专用”的发展模式下,未来物联网服务中间件各模块的设计与实现还需开展如下研究工作:

(1) 考虑到物理实体服务实现架构需要具有轻量级特性,以及与云服务及虚拟实体服务的交互接口需要具有统一性,未来将普遍采用 REST 风格的 Web 服务来实现物联网中的各类服务。目前主要是在轻量级应用协议 CoAP 和 EBHTTP 之上构建 REST 风格的物理实体 Web 服务(如表 2 中所示),为各类物联网应用系统的建立提供了可共享的基础软件模块。但是,它们在情境检测、访问控制和冲突解析方面的能力还有待进一步的完善;

(2) 考虑到实体服务的异构性和情境感知性,未来将主要采用基于语义的服务描述方法,以屏蔽采用不同方法建立各类实体服务的语法差异,便于服务的管理、发现与组合。目前已经提出了多种基于本体描述语言的服务描述模型(如图 7 和图 8 所示),它们不仅可以描述实体服务的类型、输入参数和输出参数等功能性属性,还可以描述服务提供的情境、质量和约束等非功能性属性,因此模型上已经相当的完备。但是,服务描述语言(如 OWL、OWL-S、EASY-L<sup>[85]</sup>和 SensorML<sup>[89]</sup>等)还有待于进一步规范 and 统一,以真正实现物联网中各类服务之间的无缝互联;

(3) 考虑到实体服务的大规模性和动态可用性,未来将主要采用结构化分布式或混合式服务注册管理方法,并建立更加高效的基于语义的分布式服务发现方法,以提供整个物联网应用系统的可靠性、可扩展性和可维护性。目前已提出了一些分布式或混合式服务注册方法,以及基于洪泛(或路由)服务请求或基于一致性哈希的服务发现方法(如第 4.4 节所述),但是在大规模、资源受限的物联网环境中,这些方法在可扩展性和服务发现响应率和响应速度方面都还有待于进一步验证和改进;

(4) 考虑到实体服务的动态可用性和物联网应用任务的动态变化性,未来将主要采用基于业务流模型的任务编程接口和运行时服务组合方法,以

提高整个物联网应用系统的智能性与灵活性。目前已经基于图规划或智能规划,采用高级编程语言、规划领域定义语言或可视化编程方式实现了任务需求的动态定义(如表 4 中的 SOCRADES 和 UBIWARE),但是与之相适应的服务模版确定、实体服务动态选择和运行时绑定等算法都还有待于进一步的研究。

## 参考文献

- [1] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy. Smart Objects as Building Blocks for the Internet of Things. *IEEE Internet Computing*, 2010, 14(1):44-51
- [2] A. Al-Fuqaha, M. Guizani, M. Mohammad. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communication Surveys & Tutorials*, 2015, 14(4):2347-2376
- [3] Electronics and Telecommunication Research Institute (ETRI) of the Republic of Korea. Requirements for Support of USN Applications and Services in NGN Environment// ITU NGN-GSI Rapporteur Group Meeting, Geneva, Switzerland, 2007: 11-21
- [4] Ninghui Sun, Zhiwei Xu, Guojie Li. Sea Computing: A New Computing Model of Internet of Things. *Communications of the CCF*, 2010, 6(7): 52-57 (in Chinese)  
(孙凝晖, 徐志伟, 李国杰. 海计算: 物联网的新型计算模型. *中国计算机学会通讯*, 2010, 6(7): 52-57)
- [5] Haiming Chen, Li Cui, Kaibin Xie. A Comparative Study on Architectures and Implementation Methodologies of Internet of Things. *Chinese Journal of Computers*, 2013, 36(1): 168-188 (in Chinese)  
(陈海明, 崔莉, 谢开斌. 物联网体系结构与实现方法的比较研究. *计算机学报*, 2013, 36(1): 168-188)
- [6] Hequan Wu. Review on Internet of Things: Application and Challenges. *Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition)*, 2010, 22(5): 526-531 (in Chinese)  
(邬贺铨. 物联网的应用与挑战综述. *重庆邮电大学学报(自然科学版)*, 2010, 22(5): 526-531)
- [7] M.P. Papazoglou, D. Georgakopoulos. Service Oriented Computing: Introduction. *Communication of the ACM*, 2003, 46(10): 24-28
- [8] P. Banerjee, R. Friedrich, C. Bash, P. Goldsack, et al. Everything as a Service: Powering the New Information Economy. *IEEE Computer*, 2011, 44(3):36-43
- [9] X. Qiao, Y. Zhang, B. Wu, B. Cheng, S. Zhao, H. Ma, and J. Chen. An EDSOA-based Service Provisioning Approach for Internet of Things. *SCIENCE CHINA Information Sciences*, 2013, 43(10):

- 1219-1243 (in Chinese)  
(乔秀全, 章洋, 吴步丹, 等. 事件驱动、面向服务的物联网服务提供方法. 中国科学: 信息科学, 2013, 43(10): 1219-1243)
- [10] D. Miorandi, S. Sicari, F. De Pellegrini, et al. Internet of things: Vision, Applications and Research Challenges. *Ad Hoc Networks*, 2012, 10(7): 1497-1516
- [11] K. Mayer and W. Fritsche. IP-enabled Wireless Sensor Networks and their Integration into the Internet// *Proceedings of the 1st International Conference on Integrated Internet Ad hoc and Sensor Networks (InterSense 2006)*, Nice, France, 2006:1-6
- [12] S. Tompros, N. Mouratidis, M. Draaijer, A. Foglar and H. Hrasnica. Enabling Applicability of Energy Saving Applications on the Appliances of the Home Environment. *IEEE Network*, 2009, 23(6): 8-16
- [13] Q. Zhu, R. Wang, Q. Chen, Y. Liu, W. Qin. IOT Gateway: Bridging Wireless Sensor Networks Into Internet of Things// *Proceedings of the 8th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2010)*, Hong Kong, China, 2010: 347-352
- [14] R. Piyare, S. R. Lee. Towards Internet of Things (IoTs): Integration of Wireless Sensor Network to Cloud Services for Data Collection and Sharing. *International Journal of Computer Networks & Communications*, 2013, 5(5): 59-72
- [15] M. Uusitalo. Global vision for the future wireless world from the WWRF. *IEEE Vehicular Technology Magazine*, 2006, 1(2): 4-8
- [16] Machine-to-Machine communications (M2M): Functional architecture. ETSI, Technical Specification: 102 690 V1.1.1, 2011
- [17] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi. From Today's INTRANet of Things to a Future INTERNet of Things: A Wireless and Mobility Related View. *IEEE Wireless Communications*, 2010, 17(6): 44-51
- [18] Kaibin Xie, Haiming Chen, Li Cui. PMDA: A Physical Model Driven software Architecture for Internet of Things. *Journal of Computer Research and Development*, 2013, 50(6): 1185-1197 (in Chinese)  
(谢开斌, 陈海明, 崔莉. PMDA: 一种物理模型驱动的物联网软件体系结构. 计算机研究与发展, 2013, 50(6): 1185-1197)
- [19] D. Zeng, S. Guo, and Z. Cheng. The Web of Things: A Survey. *Journal of Communication*, 2011, 6(6): 424-438
- [20] A. Kansal, S. Nath, J. Liu, and F. Zhao. SenseWeb: An Infrastructure for Shared Sensing. *IEEE Multimedia*, 2007, 14(4):8-13
- [21] M. Botts, G. Percivall, C. Reed, J. Davidson. OGC Sensor Web Enablement: Overview and High Level Architecture. Wayland, MA, USA: Open Geospatial Consortium Inc., OGC White Paper, 2007
- [22] A. P. Sheth, C. Henson, S.S. Sahoo. Semantic Sensor Web. *IEEE Internet Computing*, 2008, 12(4):78-83
- [23] Y. Xu, S. Helal. An Optimization Framework for Cloud-Sensor Systems// *Proceedings of the 6th International Conference on Cloud Computing Technology and Science (CloudCom 2014)*, Singapore, 2014: 38-45.
- [24] J. Vasseur, A. Dunkels. Interconnecting Smart Objects with IP: The Next Internet. USA: Morgan Kaufmann, 2010
- [25] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A Vision, Architecture Elements, and Future Directions. *Future Generation Computer System*, 2013, 29(7): 1645-1660
- [26] Y. Xu, S. Helal. Scalable Cloud-Sensor Architecture for the Internet of Things. *IEEE Internet of Things Journal*, 2016, 3(3): 285 - 298
- [27] Y. Cheng, X. Li, Z. Li, S. Jiang, Y. Li, J. Jia, and X. Jiang. AirCloud: A cloud-based Air-quality Monitoring System for Everyone// *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys 2014)*, Memphis, USA, 2014: 251-265
- [28] R.K. Ganti, F. Ye, H. Lei. Mobile Crowdsensing: Current State and Future Challenges. *IEEE Communication Magazine*, 2011, 49(11): 32-39
- [29] D. Estrin. Participatory Sensing: Applications and Architecture. *IEEE Internet Computing*, 2010, 14(1): 12-14
- [30] A.T. Campbell, S.B. Eisenman, and N.D. Lane, et al. The Rise of People-Centric Sensing. *IEEE Internet Computing*, 2008, 12(4): 12-21
- [31] B. Aiken, J. Strassner, B. Carpenter, I. Foster, C. Lynch, J. Mambretti, R. Moore, B. Teitelbaum. Network Policy and Services: A Report of a Workshop on Middleware. IETF, Request for Comments: 2768, Feb., 2000
- [32] Haiming Chen, Li Cui. Design and Model Checking of Service Oriented Software Architecture for Internet of Things: A Survey. *Chinese Journal of Computers*, 2016, 39(5):853-871 (in Chinese)  
(陈海明, 崔莉. 面向服务的物联网软件体系结构设计及模型检测. 计算机学报, 2016, 39(5):853-871)
- [33] Jianjia Wu, Wei Zhao. WInternet: From Net of Things to Internet of Things. *Journal of Computer Research and Development*, 2013, 50(6): 1127-1134 (in Chinese)  
(吴建佳, 赵伟. WInternet: 从物网到物联网. 计算机研究与发展, 2013, 50(6): 1127-1134)
- [34] P. A. Vicaire, Z. Xie, E. Hoque, and J. A. Stankovic. Physicalnet: A Middleware for Programming Concurrent, across Administrative Domain Sensor and Actuator Networks// *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009)*, Berkeley, CA, USA, 2009: 317-318



- [35] S. Zhang, S. Goddard. A Software Architecture and Framework for Web-based Distributed Decision Support Systems. *Decision Support Systems*, 2007, 43(2007): 1133-1150
- [36] H. J. La and S. D. Kim. A Service-Based Approach to Designing Cyber Physical Systems// *Proceedings of the 9th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2010)*, Kaminoyama (Yamagata), Japan, 2010: 895-900
- [37] L. Souza, P. Spiess, and D. Guinard, et al. SOCRADES: A Web Service Based Shop Floor Integration Infrastructure// *Proceedings of the 1st Internet of Things Conference (IOT 2008)*, Stockholm, Sweden, 2008:1-10
- [38] F. Carrez, M. Bauer, and Tim Baugé et al. SENSEI Reference Architecture. Deliverable Report D3.2, Jan., 2009
- [39] D. Guinard, V. Trifa, and E. Wilde. A Resource Oriented Architecture for the Web of Things// *Proceedings of the 2nd Internet of Things Conference (IOT 2010)*, Tokyo, Japan, 2010:1-10
- [40] F. Paganelli, S. Turchi, and D. Giuli. A Web of Things Framework for RESTful Applications and Its Experimentation in a Smart City. *IEEE System Journal*, 2014, (1):1-11
- [41] S. Bandyopadhyay, M. Sengupta, S. Maiti and S. Dutta. Role of Middleware for Internet of Things: A Study. *International Journal of Computer Science & Engineering Survey (IJCSES)*, 2011, 2(3):94-108
- [42] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke. Middleware for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 2016, 3(1): 70-95
- [43] A. R. Ribeiro, F. Silva, L. C. Freitas, J. C. Costa, and C. R. Francês. Sensorbus: A Middleware Model for Wireless Sensor Networks// *Proceedings of the 3rd IFIP/ACM Latin American Conference on Networking (LANC 2005)*, Cali, Colombia, 2005: 1-9.
- [44] A. Boulis, C.-C. Han, R. Shea, and M. B. Srivastava. Sensorware: Programming Sensor Networks Beyond Code Update and Querying. *Pervasive and Mobile Computing*, 2007, 3(4): 386-412
- [45] P. L. Evensen and H. Meling. SenseWrap: A Service Oriented Middleware with Sensor Virtualization and Self-configuration// *Proceeding of the 5th International Conference on Intelligent Sensors, Sensor Networks and Information Process (ISSNIP 2009)*, Melbourne, Australia, 2009: 261-265
- [46] M. Eisenhauer, P. Rosengren, and P. Antolin. A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems// *Proceedings of the 6th Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2009) Workshops*, Rome, Italy, 2009: 1-3
- [47] K. Aberer, M. Hauswirth, and A. Salehi. A Middleware for Fast and Flexible Sensor Network Deployment// *Proceeding of the 32nd International Conference on Very Large Data Bases (VLDB 2006)*, Seoul, Korea, 2006: 1199-1202
- [48] R. Buyya, J. Broberg, A. Goscinski. *Cloud Computing: Principles and Paradigms*. New York, USA: Wiley Press, 2011
- [49] H. Park, B. Kim, Y. Ko, and D. Lee InterX: A Service Interoperability Gateway for Heterogeneous Smart Objects// *Proceedings of the 8th IEEE International Workshop on Middleware and System Support for Pervasive Computing Program (PERWARE 2011)*, Seattle, USA, 2011: 233-238
- [50] S. Zheng, Q. Wang, Z. Ding. Research and Design of Web Service-based Matter-centric Middleware in Internet of Things. *Journal of Computer Application*, 2013, 33(7): 2022-2025. (in Chinese)  
(郑树泉, 王倩, 丁志刚. 基于 Web 服务以物为中心的物联网服务中间件的研究与设计. *计算机应用*, 2013, 33(7): 2022-2025)
- [51] A. Dunkels. The uIP Embedded TCP/IP Stack. Sweden: Swedish Institute of Computer Science, Technical Report, 1.0 Reference Manual, Jun., 2006
- [52] IETF Network Working Group. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. IETF, RFC 4919, Aug., 2007
- [53] Junjun Kong, Yao Guo, Xiangqun Chen, and Weizhong Shao. An Approach to Building Systems and Applications of Internet of Things with Smart Things. *Journal of Computer Research and Development*, 2013, 50(6): 1198-1209 (in Chinese)  
(孔俊俊, 郭耀, 陈向群, 邵维忠. 一种基于智能物体的物联网系统及应用开发方法. *计算机研究与发展*, 2013, 50(6): 1198-1209)
- [54] G. Nain, F. Fouquet, and B. Morin. et al. Integrating IoT and IoS with a Component-Based Approach// *Proceedings of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2010)*, Lille, France, 2010: 191-198
- [55] C.-L. Wu, C.-F. Liao, L.-C. Fu. Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 2007, 37(2): 193-205
- [56] G. Fortino, G., Guerrieri, W. Russo. Agent-oriented Smart Objects Development// *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2012)*, Wuhan, China, 2012: 907-912
- [57] R. Lopes, F. Assis, C. Montez. MASPOT: A Mobile Agent System for Sun SPOT// *Proceedings of the 10th International Symposium on Autonomous Decentralized Systems (ISADS 2011)*, Tokyo, Japan, 2011: 25-31

- [58] H. Shi, D. Li, H. Chen, J. Qiu, Li Cui. EasiCAE: A Runtime Framework for Efficient Sensor Sharing among Concurrent IoT Applications// Proceedings of the 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2014), Hsinchu, Taiwan, China, 2014:328-335
- [59] SmartProducts Deliverable D.2.1.3: Final Version of the Conceptual Framework, Version 1.0, Feb., 2011
- [60] Z. Shelby. Embedded Web Services. IEEE Wireless Communications, 2010, 17(6):52-57
- [61] C. Bormann, A. P. Castellani, and Z. Shelby. CoAP: An Application Protocol for Billions of Tiny Internet Nodes. IEEE Internet Computing, 2012, 16(2): 62-67
- [62] IETF CoRE Working Group. Embedded Binary HTTP (EBHTTP). IETF, Internet-Draft: draft-tolle-core-ebhttp -00, Sep. 2010
- [63] M. Kovatsch, S. Duquennoy, and A. Dunkels. A Low-power CoAP for Contiki// Proceedings of the International Workshop on Internet of Things Technology and Architectures (IoTech), Valencia, Spain, 2011: 855-860.
- [64] G. Moritz, F. Golatowski, and D. Timmermann. A Lightweight SOAP over CoAP Transport Binding for Resource Constraint Networks// Proceedings of the International Workshop on Internet of Things Technology and Architectures (IoTech), Valencia, Spain, 2011: 861-866
- [65] W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota. REST Enabled Wireless Sensor Networks for Seamless Integration with Web Applications// Proceedings of the International Workshop on Internet of Things Technology and Architectures (IoTech), Valencia, Spain, 2011: 867-872
- [66] Devices Profile for Web Services Version 1.1, OASIS WS-DD Technical Committee, Standard, Jul., 2009
- [67] S. de Deugd, R. Carroll, and K.E. Kelly. SODA: Service-Oriented Device Architecture. IEEE Pervasive Computing, 2006, 5(3): 94-96
- [68] W. Drytkiewicz, I. Radusch, S. Arbanowski, and R. Popescu-Zeletin. pREST: A REST-based Protocol for Pervasive Systems// Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004), Lauderdale, Florida, USA, 2004: 340-348
- [69] L. Schor, P. Sommer, and R. Wattenhofer. Towards a Zero-Configuration Wireless Sensor Network Architecture for Smart Buildings// Proceedings of the 1st ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings (BuildSys 2009), Berkeley, USA, 2009:1-7
- [70] T. Luckenbach, P. Guber, S. Arbanowski, A. Kotsopoulos, and K. Kim. TinyREST - A Protocol for Integrating Sensor Networks into the Internet// Proceedings of the Workshop on Real-World Wireless Sensor Network (REALWSN 2005), Stockholm, Sweden, 2005:1-7
- [71] F. Zhu, M. Mutka, and L.M. Ni. Service Discovery in Pervasive Computing Environments. IEEE Pervasive Computing, 2005, (5): 81-90
- [72] S. Duquennoy, G. Guinard, J.-J. Vandewalle. The Web of Things: Interconnecting Devices with High Usability and Performance// Proceedings of the International Conference on Embedded Software and Systems (ICESS 2009), Hangzhou, Zhejiang, China, 2009: 323-330
- [73] D. Yazar and A. Dunkels. Efficient Application Integration in IP-based Sensor Networks// Proceedings of the 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys 2009), Berkeley, USA, 2009: 43-48
- [74] A.P. Castellani, N. Bui, P. Casari, M. Rossi, Z. Shelby, and M. Zorzi. Architecture and Protocols for the Internet of Things: A Case Study// Proceedings of 1st International Workshop on the Web of Things (WoT 2010) in conjunction with IEEE PERCOM 2010, Mannheim, Germany, 2010: 678-683
- [75] L. Mainetti, V. Mihali, and L. Patrono. A Software Architecture Enabling the Web of Things. IEEE Internet of Things Journal, 2015, 2(6): 445-454
- [76] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. sMAP: a Simple Measurement and Actuation Profile for Physical Information// Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys 2010), Zurich, Switzerland, 2010: 197-210
- [77] C. Hou, D. Li, J. Qiu, H. Shi, and L. Cui. Seahttp: A Resource-oriented Protocol to Extend REST Style for Web of Things. Journal of Computer Science and Technology, 2014, 29(2): 205-215
- [78] S. Avancha, C. Patel, and Joshi. Ontology-Driven Adaptive Sensor Network// Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous System: Networking and Services (MobiQuitous 2004), Cambridge, USA, 2004: 194-202
- [79] M. Imai, Y. Hirota, S. Satake and H. Kawashima. Semantic Sensor Network for Physically Grounded Applications// Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision (ICARCV 2006), Singapore, 2006:1-6
- [80] M. Eid, R. Liscano, A. El Saddik. A Novel Ontology for Sensor Network Data// Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSMA 2006), La Coruna, Spain, 2006: 75-79
- [81] The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. Web Semantics: Science, Services and Agents on the World Wide Web, 2012, 17(12): 25-32

- [82] A. I. Maarala, X. Su and J. Riekk. Semantic Data Provisioning and Reasoning for the Internet of Things// Proceedings of the 4th International Conference on the Internet of Things (IOT 2014), Cambridge, USA, 2014: 67-72
- [83] M. Ruta, F. Scioscia, G. Loseto, and et al. Mini-ME 2.0: Powering the Semantic Web of Things// Proceedings of the 3rd International Workshop on OWL Reasoner Evaluation (ORE 2014), Vienna, Austria, 2014: 8-15
- [84] G. Rocher, J. Y. Tigli, S. Lavirotte and R. Daikhi. Run-time Knowledge Model Enrichment in SWoT: A Step Toward Ambient Services Selection Relevancy// Proceedings of the 5th International Conference on the Internet of Things (IOT 2015), Seoul, 2015: 62-69
- [85] S. B. Mokhtar, D. Preuveneers, N. Georgantas, V. Issarny, and Y. Berbers. EASY: Efficient semAntic Service discovery in Pervasive Computing Environments with QoS and Context Support. Elsevier Journal of Systems and Software, 2008, 81(5): 785-808
- [86] J. Huang, F. Bastani, I-L. Yen, J.J. Jeng. Toward a Smart Cyber-Physical Space: A Context-Sensitive Resource-Explicit Service Model// Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC 2009), Seattle, Washington, USA, 2009: 122-127
- [87] G. Fortino, M. Lackovic, W. Russo, and P. Trunfio. A Discovery Service for Smart Objects over an Agent-Based Middleware// Proceedings of the 6th International Conference on Internet and Distributed Computing Systems (IDCS 2013), Hangzhou, China, 2013: 281-293
- [88] W. Wang, S. De, G. Cassar, and K. Moessner. Knowledge Representation in the Internet of Things: Semantic Modelling and its Applications. Automatika-Journal for Control, Measurement, Electronics, Computing and Communications, 2013, 54(4): 388-400
- [89] M. Botts and A. Robin. OpenGIS Sensor Model Language (SensorML) Implementation Specification Version: 1.0.0. Open Geospatial Consortium Inc., OpenGIS Implementation Specification, OGC 07-000, 2007
- [90] Z. Wang, Q. Pan, and T. Xing. Survey on Real-time Search for Entities of Internet of Things. Application Research of Computers, 28(6): 2001-2006, 2011  
(王智, 潘强, 邢涛. 面向物联网的实体实时搜索服务综述. 计算机应用研究, 2011, 28(6): 2001-2006)
- [91] K.-K. Yap, V. Srinivasan, and M. Motani. MAX: Human-Centric Search of the Physical World// Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems (SenSys 2005), San Diego, USA, 2005:166-179
- [92] X. Liu, Q. Huang, and Y. Zhang. Balancing Push and Pull for Efficient Information Discovery in Large-Scale Sensor Networks. IEEE Transactions on Mobile Computing. 2007, 6(3): 241- 251
- [93] S. Evdokimov, B. Fabian, S. Kunz, and N. Schoenemann. Comparison of Discovery Service Architectures for the Internet of Things// Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2010), Newport Beach, USA, 2010: 237-244
- [94] A. J. Jara, P. Martinez-Julia and A. Skarmeta. Light-Weight Multicast DNS and DNS-SD (lmDNS-SD): IPv6-Based Resource and Service Discovery for the Web of Things// Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012), Palermo, Italy, 2012: 731-738
- [95] S. Yang, Y. Xu, and Q. He. Ontology based Service Discovery Method for Internet of Things// Proceedings of the IEEE International Conference on Internet of Things, Cyber Physical and Social Computing (iThing/CPSCoM 2011), Dalian, China, 2011: 43-47
- [96] A. Ostmark, J. Eliasson, P. Lindgren, A. van Halteren, and L. Meppelink. An Infrastructure for Service Oriented Sensor Networks. Journal of Communications, 2006, 1(5):20-29
- [97] D. Chakraborty, A. Joshi, Y. Yesha, T. Finin. Toward Distributed Service Discovery in Pervasive Computing Environments. IEEE Transactions on Mobile Computing, 2006, 5(2):97-112
- [98] Q. Wei, Z. Jin, G. Li and L. Li. Preliminary Study of Service Discovery in Internet of Things: Feasibility and Limitation of SOA. Journal of Frontiers of Computer Science and Technology, 2013, 7(2): 97-113 (in Chinese)  
(魏强, 金芝, 李戈, 李力行. 物联网服务发现初探: 传统 SOA 的可行性和局限性. 计算机科学与探索, 2013, 7(2): 97-113)
- [99] H. Wang, C. C. Tan and Q. Li. Snoogle: A Search Engine for the Physical World// Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM 2008), Phoenix, USA, 2008: 1188-1202
- [100] D. Zhang, L. Yang, H. Huang. Searching in Internet of Things: Vision and Challenges// Proceedings of the 9th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA 2011), Busan, South Korea, 2011: 201-206
- [101] T. Teixeira, S. Hachem, V. Issarny, and N. Georgantas. Service Oriented Middleware for the Internet of Things: A Perspective// Proceedings of the 4th European Conference on Towards a Service-based Internet (ServiceWave 2011), Poznan, Poland, 2011: 220-229
- [102] G. Cassar, P. Barnaghi, W. Wang, K. Moessner. A Hybrid Semantic



- Matchmaker for IoT Services// Proceedings of the IEEE International Conference on Green Computing and Communications (GreenCom 2012), Besancon, France, 2012: 210-216
- [103] Q. Wei, Z. Jin, Y. Xu. Service Discovery for Internet of Things Based on Probabilistic Topic Model. *Journal of Software*, 2014, 25(8):1640-1658 (in Chinese)  
(魏强, 金芝, 许焱. 基于概率主题模型的物联网服务发现. 软件学报, 2014, 25(8):1640-1658)
- [104] B. M. Elahi, K. Romer, B. Ostermaier, M. Fahrmaier and W. Kellerer. Sensor Ranking: A Primitive for Efficient Content-based Sensor Search// Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN 2009), San Francisco, USA, 2009: 217-228
- [105] M. Li, H. Chen, X. Huang, and L. Cui. EasiCrawl: A Sleep-aware Schedule Method for Crawling IoT Sensors// Proceedings of the 21th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2015), Melbourne, Australia, 2015: 148-155
- [106] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, and C. Truong. SPITFIRE: Toward a Semantic Web of Things. *IEEE Communications Magazine*, 2011, 49(11): 40-48
- [107] C. Truong, K. Romer. Content-Based Sensor Search for the Web of Things// Proceedings of the IEEE Global Communications Conference (GLOBECOM2013), Atlanta, USA, 2013: 2654-2660
- [108] D. Hong, H. Wang, K. Whitehouse. Clustering-based Active Learning on Sensor Type Classification in Buildings Categories and Subject Descriptors// Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM 2015), Melbourne, Australia, 2015: 363-372
- [109] K. Whitehouse, F. Zhao, J. Liu. Semantic Streams: A Framework for Composable Semantic Interpretation of Sensor Data// Proceedings of the 3rd European Conference on Wireless Sensor Networks (EWSN 2006), Zurich, Switzerland, 2006: 5-20
- [110] X.-Q. He, Y.-H. Wang. Service Selection Algorithm Based on Dynamic Assessment for Web of Things. *ACTA ELECTRICA SINA*, 2013, 41(1): 117-122 (in Chinese)  
(何秀青, 王映辉. 物联网服务动态评价选择方法研究. 电子学报, 2013, 41(1): 117-122)
- [111] B. Jia, Y. Yang, and Y. Sun. Resource Selection Method for Active Service based on Internet of Things. *Journal of Huazhong University of Sci. & Tech. (Natural Science Edition)*, 2012, 40(SI): 47-50 (in Chinese)  
(贾冰, 杨永健, 孙永雄. 基于物联网的主动服务资源选择方法. 华中科技大学学报 (自然科学版), 2012, 40(增刊 I): 47-50)
- [112] M. Kovatsch, M. Weiss, D. Guinard. Embedding Internet Technology for Home Automation// Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2010), Bilbao, Spain, 2010: 1-8
- [113] J. Huang, F. Bastani, I.-L. Yen, W. Zhang. A Framework for Efficient Service Composition in Cyber-Physical Systems// Proceedings of the 5th IEEE International Symposium on Service Oriented System Engineering (SOSE 2010), Nanjing, China, 2010: 291-298
- [114] J. Luo, F. Zhou, R.-F. Li. Dynamic Service Composition Mechanism Based on OSGi. *Journal of Computer Research and Development*, 2014, 51(2):420-428 (in Chinese)  
(罗娟, 周峰, 李仁发. 基于分布式 OSGi 的动态服务组合算法. 计算机研究与发展, 2014, 51(2):420-428)
- [115] F. Kawsar, T. Nakajima, J.H. Park, and S.-S. Yeo. Design and Implementation of a Framework for Building Distributed Smart Object System. *Springer Journal of Supercomputing*, 2010, 54:4-28
- [116] Z. Song, A. A. Cardenas, and R. Masuoka. Semantic Middleware for the Internet of Things// Proceedings of the Internet of Things 2010 (IoT 2010), Tokyo, Japan, 2010:1-8
- [117] H. Dempo and M. Yoshida. CUBIQ: Cross UBIQUITOUS Platform Architecture// Proceedings of the 10th Annual International Symposium on Applications and the Internet (SAINT 2010), Seoul, Korea, 2010: 1-6
- [118] C. Goumopoulos and A. Kameas. Smart Objects as Components of UbiComp Applications. *International Journal of Multimedia and Ubiquitous Engineering*, 2009, 4(3):1-20
- [119] B. Kim, T. Kim, H.-G. Ko, and et al. Personal Genie: A Distributed Framework for Spontaneous Interaction Support with Smart Objects in a Place// Proceedings of the 7th ACM International Conference on Ubiquitous Information Management and Communication (IMCOM 2013), Kota Kinabalu, Malaysia, 2013:1-7
- [120] A. Katasonov, O. Kaykova, O. Khriyenko, and et al. Smart Semantic Middleware for the Internet of Things// Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2008), Funchal, Portugal, 2008: 169-178
- [121] C. M. Liang, B. F. Karlsson, and N. D. Lane, et al. SIFT: Building an Internet of Safe Things// Proceedings of the 14th ACM International Conference on Information Processing in Sensor Networks (IPSN 2015), Seattle, USA, 2015: 298-309.



**CHEN Hai-Ming**, born in 1981, Ph. D., research assistant professor. His research

interest is on wireless ad hoc networks, sensor networks, and Internet of Things.

**SHI Hai-Long**, born in 1986, Ph. D., research assistant professor. His research interest is on wireless sensor networks and Internet of Things.

**LI Meng**, born in 1989, Ph.D. candidate. His research interest

### Background

So far Service Oriented Architecture (SOA) has been widely used in constructing application systems of Internet of Things (IoT), by abstracting both physical objects and cloud platforms as resources to provide services. Services provided by physical objects and cloud platforms are referred to as entity services and cloud services respectively. To construct IoT application systems, a service middleware, which lies over entity service and cloud service, should be designed and implemented to provide infrastructure software for IoT task programming, discovering required services, supporting interconnection and interoperation among diverse services.

At present, developers are confronted with many challenges to build service middleware for IoT, because of the unique characteristics of entity services, namely heterogeneity, large scale, dynamic availability, context awareness, spatial-temporal correlation, resource limitation, and actuating conflict. Besides, entity services can be located at the thing side or the cloud side, which are referred to as physical entity services and virtual entity services respectively. Taking different types of entity services as basic module, we can develop different types of IoT systems, namely thing-centric (or frontend-centric) IoT and cloud-centric (or backend-centric)

is on the discovery and search of Internet of things.

**CUI Li**, born in 1962, Ph.D., Professor. Her current research is focused on sensor technology, wireless sensor networks and Internet of Things.

IoT respectively. They have to solve almost the same challenges to design and develop service middleware.

To make a deep investigation on current work on service middleware for IoT, in this paper we first present the functional model of service middleware for IoT, and outline challenges to design and implement service middleware for IoT when taking the characteristics of entity services into consideration. Confronted with these challenges, we present a comprehensive review of implementation methods of each module of service middleware for IoT. Then we make a comparison study on the functional property of the currently proposed service middlewares for IoT by analyzing their different mechanisms adopted to implement service description, service registration, service discovery, service composition and task programming interface. Finally, we look into future developing trend of middleware for IoT.

This work is supported by the International S&T Cooperation Program of China (ISTCP) under Grant No. 2013DFA10690, the National Natural Science Foundation of China (NSFC) under Grant Nos. 61672498, 61502461, and 61100180.