

面向服务的物联网软件体系结构与模型检测

陈海明, 崔莉

(中国科学院计算技术研究所, 北京 100190)

摘 要 物联网软件体系结构用于定义物联网应用系统的构件模型和交互拓扑, 是构建支持水平互联、异构集成、资源共享和动态维护的物联网应用系统的基础。面向服务的软件体系结构(SOA)是目前构建各类物联网应用系统最常用的风格, 已经提出了多种面向服务以“物端”为中心的物联网软件体系结构参考模型。本文对这些参考模型进行了比较分析, 指出了它们基于的软件构件类型、采用的设计原则及具有的结构属性不同, 说明了这些参考模型适用于指导不同规模和不同应用需求的物联网应用系统的构建, 并且分析了基于软件体系结构的物联网应用系统的形式化建模与模型检测方法。最后, 指出了未来为建立基于软件体系结构的物联网应用系统开发方法还需要开展的研究工作。

关键词: 物联网; 面向服务; 软件体系结构; 参考模型; 模型检测

中图法分类号: TP301 DOI 号:

Design and Model Checking of Service Oriented Software Architecture for Internet of Things: A Survey

CHEN Hai-Ming, CUI Li

(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

Abstract Software architecture of Internet of Things (IoT) is to define component model and interconnection topology of IoT application systems with the aim to construct IoT application systems featured by horizontally interconnectability, heterogeneously integratability, resource usability and dynamically maintainability. Service Oriented Architecture (SOA) is currently the most popular architecture style used in guiding development of IoT application systems. So far, there have been several architecture reference models based on SOA proposed for thing-centric IoT applications. In this paper, we make a comparative analysis on these reference models in terms of component type, design principle and structural characteristics. The results of comparison explain why these reference models are suitable for guiding design of IoT application systems with different scales and requirements. We also analyze ways to establish formal models and model check of IoT application systems based on these architecture reference models. Finally, we figure out some future work to be done to establish a software-architecture-based development method of IoT application systems.

Key words Internet of Things; Service Oriented Architecture; Software Architecture; Reference Model; Model Checking

本课题得到国家国际科技合作专项项目(2013DFA10690)、国家自然科学基金青年科学基金项目(61100180)、中国科学院战略性先导科技专项项目(XDA06010403)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与网络化嵌入式计算系统, E-mail: chenhai ming@ict.ac.cn。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。

1 引言

随着传感技术、通信技术与嵌入式计算技术的发展,物理空间中的越来越多的物品具有了感知执行(简称为感执)、信息处理、数据存储和通信等能力,使得它们不仅能够标识自身相关的属性信息,而且还能采集所处环境的物理信息,有些还能执行一定的命令对物理环境进行操作,这样的物品被称为智能物品(Smart Object)^[1]。智能物品能够基于现有的通信基础设施(一般指互联网)互联起来,形成动态的松耦合系统,被称为物联网(IoT, Internet of Things^[2])。物联网不仅可以使物理空间中的物品之间进行信息交换,还可以使它们与信息空间中的计算资源、社会空间中的用户进行融合计算^[3],进而实现用户对物品的精确识别、定位与跟踪,以及对物理世界的实时监控和智能管理。因此,物联网系统也被认为是建立在互联网之上,融合了物理空间、信息空间和社会空间(简称为三元空间)的叠加网络(Overlay Network),是互联网技术的应用拓展^[4]。

需要指出的是,目前对于三元空间融合计算方面的研究,除了本文关注的物联网以外,还有其它类似的概念,比如信息物理融合系统(CPS, Cyber-Physical System)^[5]、信息物理社会系统(CPSS, Cyber-Physical-Social System)^[6]、社会物联网(SIoT, Social Internet of Things)^[7]。这些概念所指的系统虽然在设计目标和研究内容上各有侧重,比如CPS的设计目标侧重于对物理空间中低概率事件做出实时响应和控制,CPSS的设计目标侧重于将人类社会的信息与物理世界和网络空间中的信息融合起来进行处理,SIoT的设计目标是对物品建立类似于人类社会的社交网络,但是,从系统组成和系统功能目标而言,IoT与CPS、CPSS、SIoT是一样的,都是将三元空间中的物理和信息资源通过网路连接起来进行融合处理,最终达到人机物的协同工作,以提高人类社会生产和生活的信息化与智能化水平。因此,本文将不严格区分这些概念,都将其理解为物联网系统。

物联网系统的组成架构如图1所示,其中,智能物品、智能设备及局域智能系统与物理世界直接进行交互,构成了物联网系统的“物端”(也被称为“前端”^[8]);提供计算、存储和其它资源的云计算平台对来自物端的感知数据进行融合处理和智

能分析,及做出执行控制,构成了物联网系统的“云端”(也被称为“后端”^[8]);连接物端和云端的通信基础设施构成了物联网系统的“网端”。

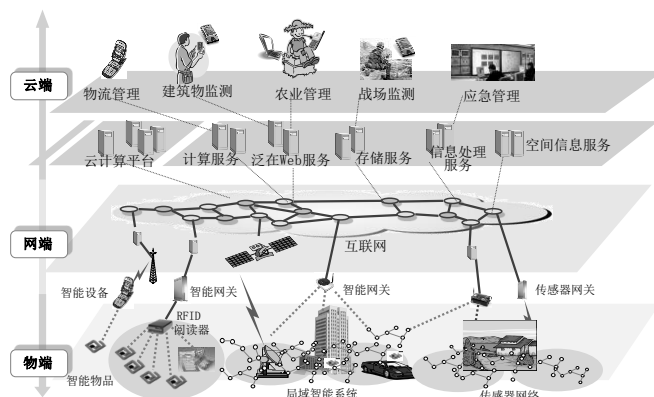


图1 物联网系统的组成架构^①

从图1可以看出,物联网系统的组成具有以下显著特点:(1)物端的智能物品、智能设备和局域智能系统一般由各行业的用户专门建立,采用不同的通信协议和数据格式,相互孤立;(2)云端的计算、存储和信息处理平台与物端感知、执行设备在性能和功能上具有很大的差异性。

基于这样的系统组成部件设计与实现各种物联网应用系统(如智慧交通、智慧城市、智慧环保等)^[9]将需要首先考虑物端的智能物品、智能设备和局域智能系统之间以及与云端计算、存储和信息处理平台之间的水平互联问题;其次考虑这些部件之间的异构集成问题;此外为降低物联网应用系统的建设成本,提高物联网应用系统组成部件的利用率,还需要考虑资源共享问题,即云端计算平台和物端感执设备上可以建立多个不同功能的软件模块,并且每个软件模块可以被多个不同的应用共享使用;为适应多应用共享的场景,还需要考虑部件功能的动态维护问题,即云端计算平台或物端感执设备上的软件模块的功能可以根据用户需求进行更新或根据应用场景自动进行调整。

从软件开发角度来看,上述总结的设计与实现各种物联网应用系统的问题可以归结为:如何根据应用需求将“物端”和“云端”资源提供的能力抽象为统一类型的软件构件,并定义这些构件之间的交互拓扑,建立实现物端与云端的互联与互操作的统一的物联网软件体系结构,用于指导各种支持水平互联、异构集成、资源共享和动态维护的物联

^① ITU Internet Report 2005: The Internet of Things, 7th Edition, <http://www.itu.int/pub/S-POL-IR.IT-2005/e>, 2005.11.18.

网应用系统的构建，同时提高物联网软件构件的可复用性和可维护性，并降低物联网应用系统的开发与维护成本。

可见，物联网软件体系结构用于定义物联网应用系统的构件模型和交互拓扑，是构建支持水平互联、异构集成、资源共享和动态维护的物联网应用系统的基础。采用不同的构件类型，定义不同的交互拓扑会导致不同的物联网系统属性，包括功能和非功能属性（如水平性、自适应性、互操作性等），从而形成不同风格的物联网软件体系结构。SOA（Service Oriented Architecture）是为实现不同应用系统间互联互通，解决信息孤岛等问题提出的一种灵活的，并独立于具体实现平台的软件体系结构风格^②。SOA 中的服务是指具有自描述性（Self-describing），支持快速、低成本分布式应用程序集成的软件模块，一般由服务提供者作为开放组件部署在可以通过网络访问的平台上。基于 SOA 建立的软件系统具有以下属性^[10]：（1）通过服务屏蔽资源的差异性；（2）同一资源可以提供不同的服务；（3）每个服务可被多个不同的应用重用；（4）各个服务可以根据应用需求或应用场景的变化分别更新或自动调节其功能。因此，目前提出的物联网应用系统的设计方法主要基于 SOA，即将现有的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、智能设备及局域智能系统与云计算平台提供的信息采集、处理、存储和分析等能力的都统一抽象为服务，将它们作为软件构件按照应用需求交互集成起来，实现物端与云端的互联与互操作，从而构成松耦合、分布式协同工作的物联网应用系统^[11]。

为便于区分，本文将“云端”中的资源提供的服务称为云服务（Cloud Service）；将“物端”智能物品提供的服务称为实体服务（Entity Service），包括物理实体服务（Physical Entity Service）和虚拟实体服务（Virtual Entity Service），其中实体服务是指在单个物端资源上实现的软件模块，如某点的温度感知模块；虚拟实体服务是指由多个物理实体服务与云服务组合后得到的服务，如感知某个区域的平均温度，它由区域内多个温度感知节点提供的物理实体服务与计算平均值的云服务组合而成的软件模块。

下面首先介绍两类基于面向服务的软件体系

结构构建的物联网应用系统，分别是以“云端”为中心的物联网应用系统和以“物端”为中心的物联网应用系统，同时给出了本文把以“物端”为中心的物联网应用系统作为研究对象的理由；接着，从构件类型、设计原则和结构属性三个方面，对目前已经提出的多种面向服务以“物端”为中心的物联网软件体系结构参考模型进行比较分析；然后，分析基于软件体系结构的物联网应用系统形式化建模和模型检测的方法；最后，对未来基于软件体系结构的物联网应用系统设计与实现方法的研究进行展望。

2 面向服务的物联网应用系统

基于图 1 所示的物联网系统的架构，根据物联网应用系统中物理实体服务实现的不同功能，可以将目前采用面向服务的方法建立的物联网系统分为以下两类：（1）以“云端”为中心的物联网应用系统；（2）以“物端”为中心的物联网应用系统。

2.1 以“云端”为中心的物联网系统

在以“云端”为中心的物联网应用系统中，物理实体服务实现基本的物理信息采集、本地的信息处理和向云端传输信息的功能，但不对外提供可直接访问的服务接口，至于物理信息的海量存储、关系处理和知识挖掘等功能都在云端实现。早期的典型系统如 EPCglobal[®]（早期的版本为美国麻省理工学院 AutoID 实验室发布的 networked RFID 系统^③）和日本东京大学 Sakamura-Koshizuka 实验室提出的 uID 系统^[12]。这两个系统也是物联网在发展初期的代表，它们都是基于自动识别技术并采用相似的系统架构建立，即由终端设备（或称为阅读器）采集物品的标识信息，并通过网络将标识信息传输到系统的后端；后端提供物品信息服务，即根据物品标识查找对应的信息服务，对物品信息进行相应的处理。类似的系统还有 EPC Network^[13]和 Smart Object System^[14]。随着传感技术的发展，终端设备不仅可以采集物品标识信息，还可以采集其它物品属性信息和环境物理量信息，并将信息直接或借助物联网网关^{[15]-[17]}传输到云端进行统一存储和处理，并以

^② <http://www.ibm.com/developerworks/cn/architecture/ar-soastyle/>

^③ The EPCglobal Architecture Framework, EPCglobal Final Version 1.3 [EB/OL], http://www.gs1.org/gsmp/kc/epcglobal/architecture/architecture_1_3-framework-20090319.pdf, 2009

^④ <http://http.autoidlabs.org/single-view/dir/article/6/93/page.html>

服务形式提供给用户和软件开发者^[18], 软件开发者基于这些位于云端的服务构建物联网应用系统。例如, WInternet^[19]将感知数据和数据感知等方法封装成 Internet 之上的服务(称为 Netlet), 通过 Netlet 之间的互联实现物联网应用系统的构建; 群智感知(也被称为参与式感知、机会感知)^[20]也是属于这类物联网应用系统, 它将用户携带的具有传感器的智能移动终端作为前端感知设备, 这些设备有意识或无意识地采集数据, 并将数据进行一定的本地处理后传输到云端进行深度分析, 以实现城市环境、交通状况、群体活动等社会感知应用系统。

对于软件开发者来说, 构建这类系统的主要挑战在于如何建立一个支持海量物品信息管理和数据处理的后端云服务平台。目前已经建立的云平台主要有 Xively[®](最初的名字为 Pachub^[21])和 Aneka^[22]。Xively 提供了支持物联网应用系统开发的公共云平台服务, 主要包括数据存取服务和设备管理服务; Aneka^[22]是一个将 Google、Amazon 和 Azure 等公共云平台和私有云平台结合起来建立的混合式云平台, 它为开发以云端为中心的物联网应用系统提供了数据分析、可视化、计算和存储服务。为提高这些云服务的智能性, Lin 等人^[23]将来自物理世界的物理数据以本体和语义服务的形式提供给物联网软件开发者; SenaaA^[24]也采用本体和语义服务的形式将来自不同类型的传感数据转化为云端具有一定领域知识表达能力的数据服务, 并与云端的感知服务连接起来, 从而对物理对象的动态事件进行处理、推理和反馈, 并采用基于语义的访问策略保证整个物联网应用系统的安全; Elci 等人^[25]将系统中的每个物理实体抽象为自治语义智能体(ASA, Autonomous Semantic Agent), 并在云端以语义 Web 服务形式实现 ASA, 其中一个 ASA 为 Facilitator 智能体, 用于管理系统内智能体间的互联拓扑等。

2.2 以“物端”为中心的物联网应用系统

近年来在 IPSO[®]等组织的推动下, 使资源受限的智能物品和感知节点能够通过 IP 协议, 如 uIP^[26]和 IPv6/6LowPAN^[27], 接入互联网成为网络空间中可以被直接寻址的设备, 其中 uIP 是 IPv4 协议在感知节点上的实现版本, 6LowPAN 是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, 用于对 IPv6 报文进行拆分和组

合、报头压缩、地址映射等。基于 uIP 或 IPv6/6LowPAN 协议, 软件开发者可以在每个感知节点或网关节点上建立可以被直接访问的物理实体服务, 其实现形式一般为可以被远程调用的分布式对象、具有与环境自主交互能力的智能体、SOAP 风格的 Web 服务和 REST 风格的 Web 服务。这些物理实体服务可以被用户或软件开发者直接调用以获取物理感知信息或操作物理环境, 也可以与后端的数据分析服务、可视化服务等云服务根据应用需求组合起来, 逐级建立虚拟实体服务, 最终建立物联网应用系统。

从以上对两类物联网应用系统的介绍可以看出, 当开发者采用面向服务的软件体系结构设计以“云端”为中心的物联网应用系统时, 将实体资源抽象为物理信息源, 相应地将物理实体服务的功能定义为采集、处理和传输物理信息, 并在云端建立存储、分析和可视化物理信息的服务。构建这类物联网应用系统的主要工作是将云端的这些服务按照应用需求组合起来, 不涉及对物理实体服务的操作。因此, 面向服务以“云端”为中心的物联网应用系统的软件体系结构与现有的云计算系统一样。但是, 在以“物端”为中心的物联网应用系统中, 物理实体提供实现物理信息采集、本地信息处理和局部信息分析等功能的服务接口, 并能够与提供信息处理、信息可视化等功能的云服务相组合。考虑到物理实体服务具有一些与云服务不同的特性(见第 2.1 节), 以及这些特性将给云服务与实体服务之间的互联与互操作带来一定的影响, 软件开发者在采用 SOA 构建这类物联网应用系统时, 需要重新设计物联网应用系统的软件体系结构参考模型(或称为软件架构)。因此, 本文把以“物端”为中心的物联网应用系统作为研究对象, 对面向服务的物联网应用系统的设计与实现方法进行深入分析。为便于论述, 下文中提到的物联网系统默指为面向服务以“物端”为中心的物联网应用系统。

3 面向服务的物联网软件体系结构设计

软件开发者采用面向服务的软件体系结构设计物联网系统, 将物端和云端资源提供的获取和处理等功能都统一抽象为服务, 可以屏蔽位于物理空间中的实体资源和位于信息空间中的计算资源的异构性, 但是无法屏蔽不同物理实体资源提供的服务的异构性, 以及物理实体服务本身具有的一

[®] https://xively.com/whats_xively/

[®] <http://www.ipso-alliance.org>

些特有性质，比如与物理世界的交互性、资源受限性、动态性和不完整性。物理实体服务的这些特性给面向服务的物联网系统设计与实现带来了新的挑战。下面对物联网实体服务具有的特性进行分析。

3.1 物联网实体服务的特性

(1) 异构性：在传统的互联网环境中，云服务分为基础设施（Infrastructure as a Service, IaaS）、平台（Platform as a Service, PaaS）和软件（Software as a Service, SaaS）三种^[28]，每种服务都遵循统一的规则设计和实现；而对于物联网系统来讲，包含的智能物品来自不同区域、不同领域、不同时期建立的物理感知和执行系统，因此这些智能物品提供的服务语义、数据格式及服务质量（如不同的采样速率、精确度和空间分辨率等）会有所不同，这使得选择多个合适的服务来协同完成一个任务变得比较复杂；

(2) 大规模性：在传统的互联网环境中，计算、存储和信息处理等服务大多由专业的服务提供商来开发和部署，因此服务的规模有限。而在物联网系统中，具有感知或执行能力的智能物品都可以作为服务提供者。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，这将给服务命名、查找与管理带来新的挑战，而且大规模服务之间的交互也将给每个智能物品带来计算时间、存储容量、能量消耗的极大增加；

(3) 与物理世界的交互性：在传统的互联网环境中，提供的服务一般具有数据或业务流程处理能力，以及服务与服务之间或服务与用户之间的交互能力，但是这些服务没有与物理世界的交互能力；而在物联网系统中，智能物品除了可以提供基础的数据采集、处理和传输服务，还可以产生一定的动作与物理世界进行交互。由于不同的需求对同一执行器可能会产生相互冲突的执行动作请求，因此调用物联网中与物理世界交互的服务会受到更多的约束；

(4) 资源受限性：在传统的互联网环境中，服务一般被部署在高性能计算平台中，因此在构建应用系统时一般不需要考虑所采用的服务是否已经被其他应用共享，是否还有足够的资源来支持新发起的服务请求；而在物联网系统中，智能物品的计算能力、存储容量和能量供给都十分有限，因此将智能物品提供的服务和互联网中的服务组合起来构建物联网系统时需要考虑这些资源的受限性，

以保证系统的可用性；

(5) 动态性：在传统的互联网环境中，服务一般被部署在固定的计算平台上，而且服务具有一定的稳定性，即服务在其生命期内始终有效，除非遇到平台故障等特殊情况；而在物联网系统中，提供服务的智能物品会因移动和休眠等原因而失效，从而导致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，因此提高了在物联网系统中发现和管理服务的难度；

(6) 不完整性：在传统的互联网环境中，开发人员一般会在构建系统之前检查所需的服务是否存在，如果不存在的话，会提前部署，以保证服务的完整性；而对物联网系统而言，在动态的环境中无法准确判断每一个所需的服务是否存在，因此会出现想要获取的服务不存在的现象，比如构建的物联网系统用于监测某个区域的平均温度，需要区域内各个地点和时间的温度感知服务，以及计算平均值的云服务，这时无法保证所有地点和时间的温度感知服务都存在，因此需要采用估计或即时加载的方法来解决服务不完整带来的问题。

从以上对实体服务的特殊性分析可以看出，基于面向服务的软件设计与实现方法，将实体服务与云服务组合起来构建物联网系统，会面临很大的挑战，不仅要考虑实体服务的异构、大规模性，还要考虑实体服务与物理世界的交互性以及资源的受限性给实体服务共享带来的约束，并且还要考虑实体服务的动态性和不完整性给系统运行时带来的服务缺失问题。因此，在物联网系统的设计阶段，软件开发需要建立符合这些特性的物联网软件体系结构参考模型，以精确定义物联网系统的组成结构和逻辑功能。下面对目前已经建立的面向服务的物联网软件体系结构参考模型进行比较分析。

3.2 面向服务的物联网软件体系结构参考模型

软件体系结构的参考模型是从不同视角（Perspective）对软件系统的组成进行抽象，用具有精确语义的标记符号或形式语言对软件系统中的构件、连接件以及期望属性和行为进行精确规格说明（Specification）的图形化或形式化模型^[29]。最早由 Kruchten 提出的“4+1”模型^[30]指出了从五个不同的视角建立软件体系结构参考模型的视图（View），即功能视图（即逻辑视图）、开发视图、过程视图、物理视图和场景视图。

近年来，欧盟和美国等地区 and 国家的科研人员基于面向服务的设计方法，提出了多种构建物联网

系统的软件体系结构参考模型。我们根据这些物联网软件体系结构参考模型基于的基础软件构件的类型不同,将它们分为以下四类来进行分析:(1)基于可被远程调用的分布式对象的物联网软件体系结构参考模型;(2)基于具有自主环境交互能力的智能体(Agent)的物联网软件体系结构参考模型;(3)基于 SOAP 风格的 Web 服务的物联网软件体系结构参考模型;(4)基于 REST 风格的 Web 服务的物联网软件体系结构参考模型。为易于理解,本小节所比较的各种物联网软件体系结构参考模型是以功能视图为抽象视角采用图形化规格说明方法建立的。

3.2.1 基于分布式对象的物联网软件体系结构参考模型

Physicalnet^[31]是由美国佛吉尼亚大学的研究人员提出的一种基于轻量级分布式对象的物联网体系结构,它的参考模型如图 2 所示。底层的两个构件分别代表由感知设备直接提供的服务和由网关节点采集转发的服务,这两个构件在软件部署时位于物联网系统的物端;之上的构件包括服务解析、服务仓库、访问权限表和需求表,这些构件位于物联网系统的云端,负责管理感知设备和网关提供的服务,并且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;最上层的构件提供编程抽象管理服务,以屏蔽底层细节,便于物联网应用系统的软件开发者编程。

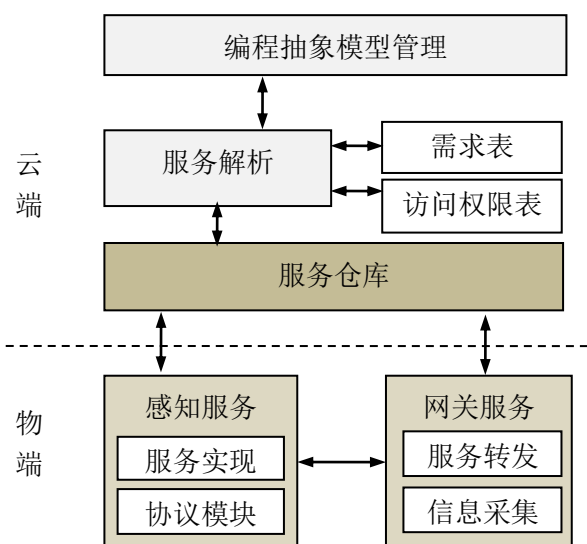


图 2 Physicalnet 体系结构参考模型

3CoFramework^[32]是由美国内布拉斯加林肯大学的研究人员提出的一种基于分层组件(Component)的物联网软件体系结构,它的参考

模型如图 3 所示。它按照组件提供的服务的抽象层次,将构建物联网系统的软件组件分为以下四层:数据层、信息层、知识层和表达层,其中数据层组件位于物端,提供分布式关系数据和空间数据;信息层组件提供专业领域信息,知识层组件提供数据分析和分险估计等服务,表达层组件提供数据显示和用户操作界面等服务,它们都位于云端。以上四层分别位于物端和云端的组件在协调器(Coordinator)的管理下,通过连接件(Connector)连接起来,从而构成一个物联网系统。

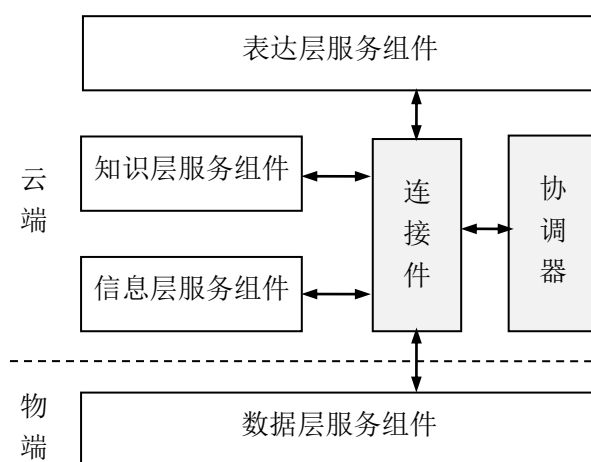


图 3 3CoFramework 体系结构参考模型

3Tiers^[33]将后端的云服务与前端的物理实体服务结合起来,提出了一种由环境层、服务层和控制层组成的三层信息物理融合系统软件体系结构,它的参考模型如图 4 所示。环境层中的物理组件是由感知器和执行器提供的服务;服务层中的云服务组件是由传统的云计算等平台提供的服务;控制层中的组件实现以下功能:(1)监视环境层和服务层提供的服务;(2)组件动态查找和服务组合;(3)组件接口适配与变异;(4)服务失效自主管理。

3.2.2 基于智能体的物联网软件体系结构参考模型

在上述基于分布式对象建立的物联网软件体系结构中,位于物端的服务组件,即 Physicalnet 中的感知服务组件和网关服务组件、3CoFramework 中的数据层服务组件、3Tiers 中的物理组件,提供了采集物理数据和与物理世界进行交互的服务,但是这些服务不构成独立的可执行的物联网系统,需要由用户调用或与云服务结合起来才能完成相关任务的执行。考虑到物联网系统可以由多个局域智能系统或具有自主环境交互能力的智能物品互联组成,而局域智能系统和智能物品与智能体

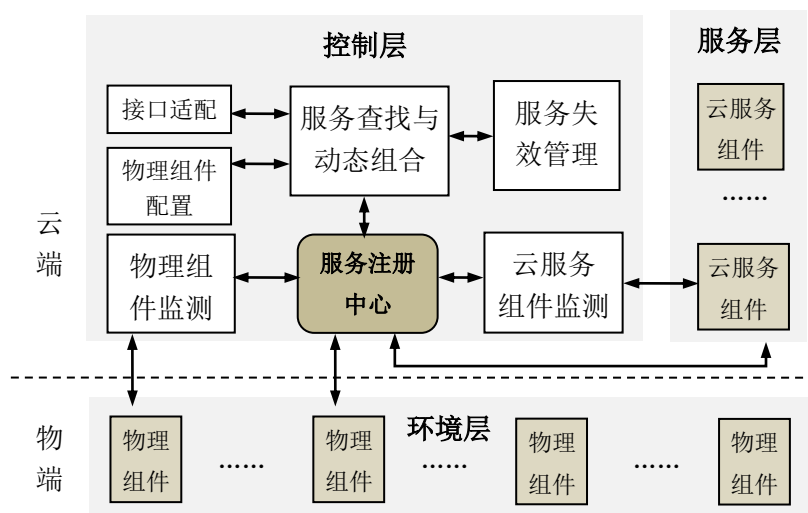


图4 3Tiers 体系结构参考模型

（Agent）具有十分相符的特性，具体表现在以下几点：

（1）自主性（Autonomous）：在没有用户使用或与云端服务互联的情况下，能够独立地执行相关任务；

（2）互动性（Interactive）：能够与其他局域智能系统或智能物品进行交互，以完成自己的任务或协助其它局域智能系统或智能物品完成任务；

（3）反应性（Reactive）：能够根据周围环境的变化及时地做出反应；

（4）主动性（Proactive）：不仅能够被动式地对环境的改变做出反应，还能够根据目标自发性地采取一些预知式行为。

为此，目前已有相关工作提出了几种以智能体为物端构件的物联网软件体系结构，它们将物理实体服务定义为可以执行用户指定任务并且能够通过周围环境的自主交互完成部分任务的软件实体，多个智能体之间可以通过网络进行互联与交互，实现物联网的分布式计算模型。

CSO (Cooperating Smart Object)^[34]是由意大利卡拉布里亚大学的研究人员提出的一种以智能体为物端构件的物联网软件体系结构，它将无线感知与执行网络（WSAN, Wireless Sensor and Actuator Network）抽象为智能体，WSAN 中的节点作为智能体的感知和执行部件，汇聚节点或网关作为智能体的协调者（Coordinator），提供与外界其它智能体、用户和环境进行交互的接口。每个 WSAN 智能体在没有用户干涉的情况下可以自主执行一定的任务，也能够根据用户的任务输入与系统内的其它

WSAN 智能体进行互联与协作，从而构建完整的物联网系统。图5给出了CSO体系结构的参考模型。

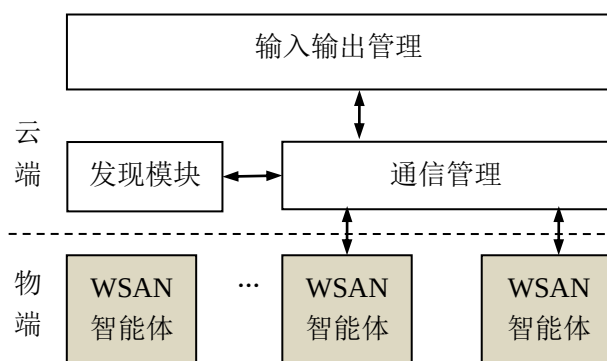


图5 CSO 体系结构参考模型

SmartProducts[®]定义的智能体是对具有异构信息源的智能产品的抽象，主要模块包括先验知识模块（Proactive Knowledge Module）和推理模块（Reasoner Module）。先验知识模块是智能体的数据模块，为推理模块提供各种实现本体推理的数据，包含的数据模型有元数据模型（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）。推理模块是智能体的核心模块，它基于先验知识模块进行本体推理，实现情境感知的自主服务。SmartProducts 以发布/订阅（Publish/Subscribe）的模式提供服务接口。采用与 CSO 类似的体系结构将 SmartProducts 集成起来可以构建基于智能体

[®] Final Version of the Conceptual Framework Version 1.0, http://www.smartproducts-project.eu/media/stories/smartproducts/publications/SmartProducts_D2.2.1_Final.pdf, 2011

的物联网系统。

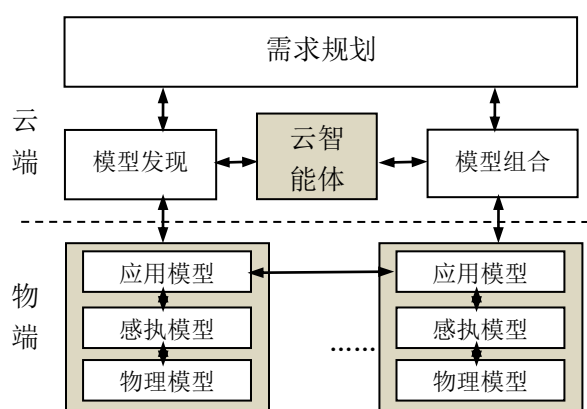


图 6 PMDA 体系结构参考模型

PMDA^[35]也是一种基于智能体的物联网软件体系结构，它的参考模型如图 6 所示。与 CSO 和 SmartProducts 不同的是它所定义的智能体包含三个模块，分别是物理模型（Physical Model）、感知模型（Sensor Execution Model）和应用模型（Application），其中物理模型是智能体的数据来源与动作执行单元，感知模型实现数据处理、知识推理和决策执行的功能，应用模型提供解析应用需求的接口。PMDA 的云端主要包括需求规划、模型发现和模型组合三个构件，其中需求规划提供了应用需求的总体规划描述和解析接口，模型发现用于查找满足应用需求的智能体，模型组合将发现的物端智能体和云端智能体组合起来构建满足需求的物联网系统。

随着万维网（World Wide Web）技术的发展，感知和执行设备可以被嵌入 Web 服务^{[36][37][38]}，通过 HTTP 等协议为用户提供实时的数据服务，并与互联网环境中的现有其它 Web 服务组合起来，构成一个基于 Web 服务的物联网系统，被称为 WoT（Web of Things）^[39]。由于目前实现 Web 服务的架构风格有两种，即 SOAP 和 REST，目前这类物联网系统的软件体系结构可以分为两种，一种是基于 SOAP 风格的 Web 服务的物联网软件体系结构，另一种是基于 REST 风格的 Web 服务的物联网软件体系结构。

3.2.3 基于 SOAP 风格的 Web 服务的物联网软件体系结构参考模型

SenseWeb^[40]是微软研究院的研究人员针对传感器网络的演进式部署模式而设计的一种基于 SOAP 风格的 Web 服务的物联网软件体系结构，它

的参考模型如图 7 所示。SenseWeb 将不同种类、不同接入方式、不同数据公开性和安全性的感知节点通过感知网关（Sense Gateway）或移动代理（Mobile Proxy）抽象为提供统一访问接口（即 Web 服务访问接口，WS-API）的感知服务，通过协调器（Coordinator），包括感知数据库（SenseDB）和任务调度模块（Tasking Module）选择应用所需的感知服务，并通过转换器（Transformer）对感知数据进行处理和显示，最终提供结构化数据供不同的应用使用。可见，SenseWeb 中的感知网关和移动代理作为物联网系统的物端服务构件，协调器、转换器作为物联网系统的云端服务构件。

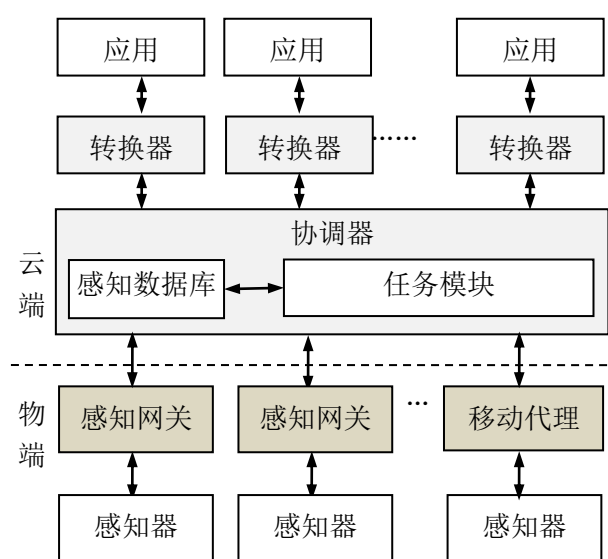


图 7 SenseWeb 体系结构参考模型

SWE(Sensor Web Enablement)^[41]是开放地理空间信息联盟（OGC）为建立地理空间网（Geospatial Web）而设计的一种基于 Web 服务的软件体系结构，它的参考模型与 SenseWeb 类似，将物端的传感器和传感器网络抽象为提供统一访问接口的 Web 服务，并将服务分为四类，即观察服务、警告服务、规划服务和提醒服务。在这些服务模块之上，定义了类似于 SenseWeb 中的协调器和转换器模块，以实现快速发现传感器与感知服务的机制、访问感知服务的标准方法、订阅感知任务和发送警告的机制，以及配置传感器参数的方法。SSW(Semantic Sensor Web)^[42]对 SWE 进行了扩展，将观测数据抽象为具有准确描述语义的本体，并让协调器具有本体推论的能力，以增强物联网系统的应用级表达。

DPWS (Device Profile for Web Service)^[43]是由德国 WS4D 项目组为使资源受限的设备间提供安

全的 Web 服务而提出的软件体系结构，它的参考模型结构与基于 SOAP 风格的 Web 服务基本一致，但在数据表示、服务描述、服务发现、消息传输等方面根据嵌入式设备的资源受限性进行了修改，其中最显著的一个修改是 DPWS 可以直接用 UDP 协议传输消息。

SOCRADES^[44]是以 DPWS 为基础提出的一种将提供 Web 服务的设备与企业应用平台（如 ERP）集成的软件体系结构。该体系结构参考模型中的物端构件即设备层服务，主要是通过 DPWS 提供的服务，云端构件包括设备管理与监测、服务发现、服务生命周期管理、跨层服务目录和安全支持等与设备管理相关的服务模块，还有业务逻辑处理监测、业务连接、虚拟化等与跨应用集成相关的服务模块，这些云端构件组成了系统的中间件服务层，在此之上建立企业应用层，实现设备与企业应用平台集成的应用系统。

3.2.4 基于 REST 风格的 Web 服务的物联网软件体系结构参考模型

基于 SOAP 风格的 Web 服务的软件体系结构允许开发者定义个性化的服务接口，这增加了系统开发的自由度，但也增加了服务描述、服务发现与服务集成的难度。考虑到前端感知设备的资源受限性等因素，近年来物联网中的物理实体服务更多的是采用 REST 风格^[45]来实现。REST 风格的 Web 服务采用 HTTP 协议，基于标准化的操作方法，降低了服务的实现难度，而且结合其他互联网标准技术，如 URI、HTML、XML 等，可以提高服务的互操作能力。目前，已有科研人员在嵌入式感知设备上实现了 REST 风格的 Web 服务，如 TinyREST^[46]和 pREST^[47]，并且 IETF CoRE 工作组正在为资源受限的设备上建立 REST 风格的 Web 服务制定统一的轻量级数据传输协议标准，包括 CoAP^[48]和 EBHTTP^[49]。这些工作使得采用基于 REST 风格的 Web 服务将位于云端和物端资源互联起来构建物联网系统成为目前最广泛采用的方法。

PhysicalMashups^[50]是由瑞士苏黎世联邦理工大学的研究人员提出的一种以 REST 风格的 Web 服务为基础的物联网软件体系结构，它的参考模型如图 8 所示。可见，物端的构件是在智能网关上建立的轻量级 Web 服务，即将各类感知器提供的数据进行缓存和格式转换，以 PULL/PUSH 形式提供的 Web 服务；云端的构件主要包括事件中心（Event Hub）和物理聚合（Physical Mashups）两个模块，

其中事件中心将汇聚由网关 Web 服务触发的事件，并分发给对应的应用，物理聚合模块将智能网关的 Web 服务与云端的 Web 服务聚合起来以快速建立用户自定义的应用。此外，各应用也可以直接访问由智能网关提供的 Web 服务。

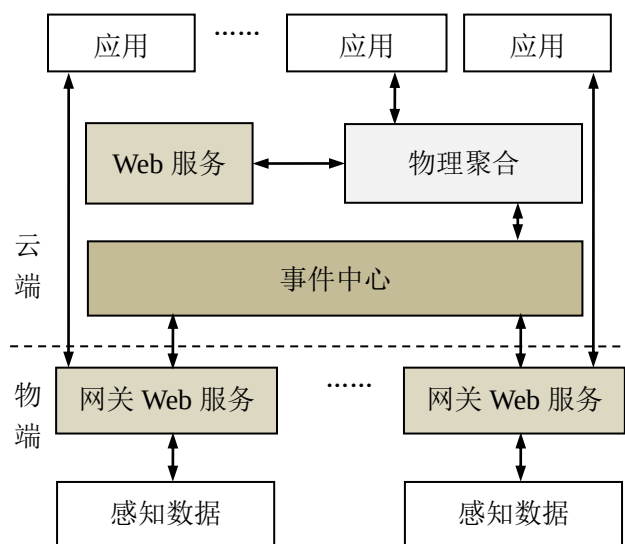


图 8 PhysicalMashups 体系结构参考模型

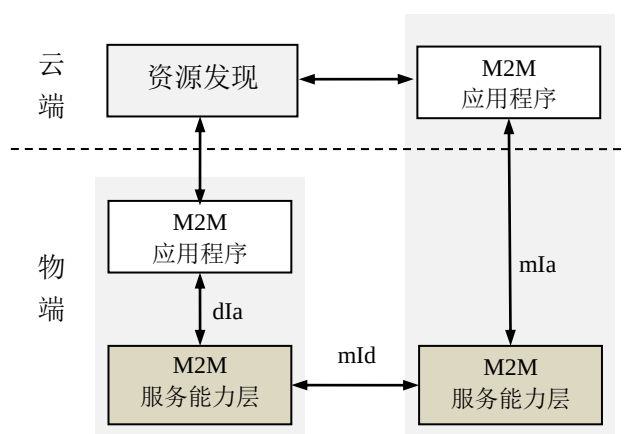


图 9 M2M 体系结构参考模型

M2M^[51]是欧洲电信标准组织（ETSI）正在制定的一个关于非智能终端设备通过移动通信网络与其他智能终端设备（IT, Intelligence Terminal）或系统进行通信的标准体系结构。基于 M2M 建立的物联网系统的软件体系结构参考模型如图 9 所示，它将扩展了 M2M 服务能力层的应用程序作为软件构件，即在具有存储模块的设备、网关和网络域中部署 M2M 服务能力层（SCL, Service Capacity Layer），设备和网关中的应用程序通过 dIa 接口访问 SCL，网络域中的应用程序通过 mIa 接口访问

SCL, 设备/网关与网络域中的 SCL 交互由 mld 接口实现。这些接口的定义基于 REST 风格, 因此我们可以将扩展了 SCL 的应用程序看作是 REST 风格的 Web 服务 (资源), 并且它们通过 URI 来命名与访问, 并基于资源的特殊属性采用字符串模式匹配算法来进行资源发现。

SENSEI^[52]是在欧盟第 7 框架项目支持下建立的将分布在全球的传感器与执行器网络 (WS&AN) 连接起来的物联网软件体系结构, 它的参考模型如图 10 所示, 它使 WS&AN 通过开放的服务访问接口与相应的语义规范来提供统一的获取环境信息以及与物理世界进行交互的服务。SENSEI 中的构件按照它们的角色、功能粒度和抽象层次可分为三层: 通信服务层、资源层与应用层组成。通信服务层中的构件将现有网络基础设施的服务, 如地址解析、流量模型、数据传输模式与移动管理等, 映射为一个统一的接口, 为资源层提供统一的网络通信服务; 资源层中的构件包括真实物理世界的资源模型、资源目录、基于语义的资源查询与解析、资源发现、资源聚合、资源创建和执行管理等模块, 为应用层与物理世界资源之间的交互提供统一的接口; 应用层中的构件为用户及第三方服务提供者提供统一的接口。

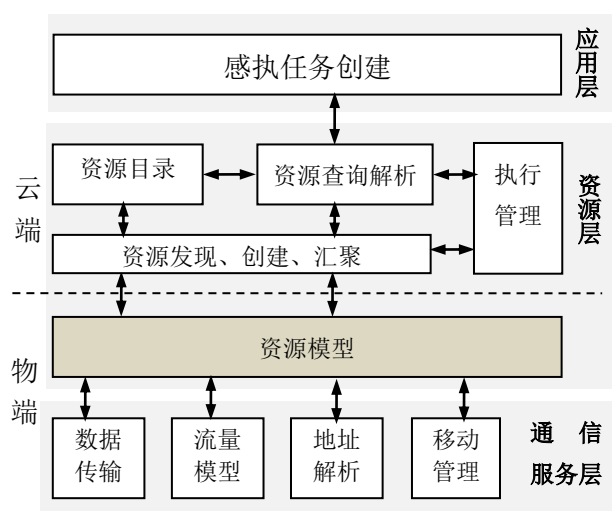


图 10 SENSEI 体系结构参考模型

IoT-A^[53]也是在欧盟第 7 框架项目支持下建立的实现局域物联系统 (Intranet of Things) 之间的水平互联和互操作的物联网软件体系结构, 它的参考模型如图 11 所示。需要指出的是, 在[54]中给出的 IoT-A 体系结构参考模型包括四个视图, 即功能视图、信息视图、部署视图和操作视图, 图 11 给出的是 IoT-A 的功能视图。与 SENSEI 一样, IoT-A

将采用不同感知和通信技术的局域物联系统抽象为提供统一服务的物联网资源模型, 并将构件按照它们的角色、功能粒度和抽象层次分为若干层, 包括设备连接与服务层、资源层、虚拟实体层、流程执行与服务组合层, 以及应用层。与 SENSEI 不一样的是, (1) IoT-A 不是将资源模型作为构建物联网系统的基本组件, 而是在资源模型之上建立虚拟实体服务, 并且通过服务解析、动态映射和服务组合等模块, 为物联网系统的构建提供更加高层的抽象接口; (2) IoT-A 是以业务流程 (Business Process) 的形式规划应用需求, 而 SENSEI 是以指定需求资源和处理树 (Processing Tree) 的方式创建应用。这些不同使得 IoT-A 比 SENSEI 具有更高的灵活性和更广泛的适用性。

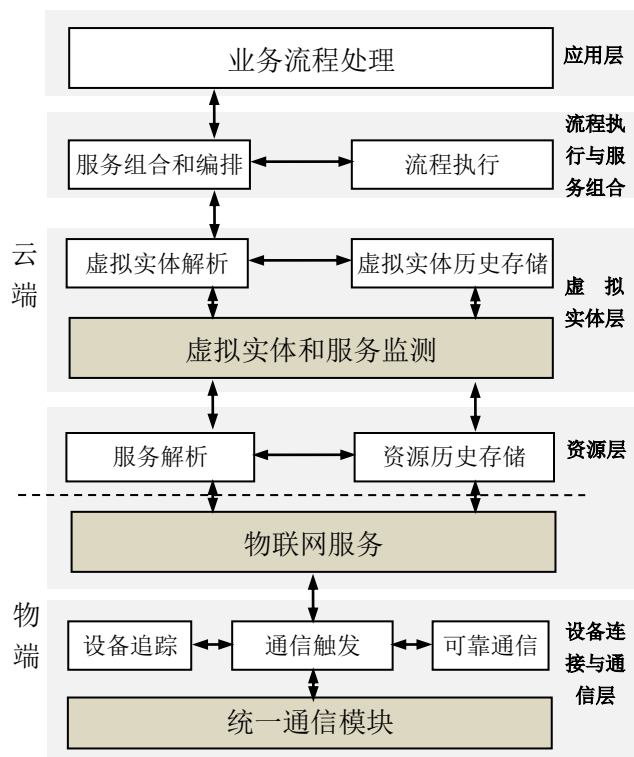


图 11 IoT-A 体系结构参考模型

2.3 面向服务的物联网软件体系结构参考模型比较分析

上述基于四种不同类型的软件构件设计的物联网软件体系结构参考模型为我们未来采用面向服务的架构 (SOA) 构建物联网系统提供了很好的参考。但是, 由于这些参考模型的设计原则不同, 即考虑的物理实体服务的特性和提供者不同, 所以它们具有不同的结构属性。表 1 给出了对这些参考

表 1 面向服务的物联网软件体系结构参考模型的设计原则与结构属性比较

物联网系统 的基础软件 构件类型	面向服务的物联网软 件体系结构参考模型	设计原则		结构属性			
		物理实体服 务的特性	物理实体服 务提供者	协同工作 模式	服务发现 方式	服务组 合方式	应用需求 描述方式
分布式对象	Physicalnet ^[31]	(1) (4)	节点 & 网关	物-物	集中式	静态	编程抽象
	3CoFramework ^[32]	(1) (2) (6)	节点 & 网关	云-物	分布式	动态	业务流抽象
	3Tiers ^[33]	(1) (3) (5) (6)	节点 & 网关	云-物	分布式	动态	/
智能体	CSO ^[34]	(1) (3)	网关	物-物	分布式	静态	业务流抽象
	SmartProducts	(1) (3)	节点	物-物	分布式	静态	业务流抽象
	PMDA ^[35]	(1) (2) (3) (6)	网关	云-物	分布式	动态	业务流抽象
SOAP 风格的 Web 服务	SenseWeb ^[40]	(1) (2)	节点 & 网关	物-物	分布式	静态	编程抽象
	SWE ^[41] /SSW ^[42]	(1) (2)	节点 & 网关	物-物	分布式	静态	业务流抽象
	DPWS ^[43]	(1) (2) (4)	节点	物-物	分布式	静态	业务流抽象
	SOCRADES ^[44]	(1) (2) (4) (6)	节点	云-物	分布式	动态	业务流抽象
REST 风格的 Web 服务	PhysicalMashups ^[50]	(1) (2) (6)	网关	云-物	集中式	动态	编程抽象
	M2M ^[51]	(1) (2)	节点	物-物	分布式	静态	/
	SENSEI ^[52]	(1) (2) (3) (5)	网关	物-物	分布式	动态	编程抽象
	IoT-A ^[53]	(1) (2) (3) (5)	网关	物-物	分布式	动态	业务流抽象

表注：物理实体服务的特性 (1)异构性，(2)大规模性，(3)与物理世界的交互性，(4)资源受限性，(5)动态性，(6)不完整性

模型的设计原则与结构属性进行比较的结果。设计原则包括以下两个方面：

(1) 考虑的物理实体服务的特性不同：物理实体服务在上述的软件体系结构参考模型中具有不同的名字，如在 Physicalnet 中被称为感知服务和网关服务，在 3CoFrameworkwork 中被称为数据层服务组件，在 3Tiers 中被称为物理组件。除了名字不同外，这些参考模型在设计时所考虑的物理实体服务的主要特性也不同。Physicalnet 主要考虑了物理实体服务的异构性和资源受限性，因此将感知设备具有的不同服务以统一的接口提供给软件开发者和用户，并通过访问权限表等服务控制多个应用在同一物理资源上的并发运行；3CoFramework 不仅考虑了物理实体服务的异构性，还考虑了它们的大规模性和不完整性，因此设计了专门的连接器和协调器来完成不同层次的服务的互联；3Tiers 除了考虑物理实体服务的异构性外，还主要考虑了它们与物理世界的交互性、动态性和不完整性，因此它设计了物理组件监测、物理组件配置和服务失效管理等实现物理实体服务的管理。同样道理，我们可以知道，其它几个参考模型的设计也都考虑了物理实体服务的异构性，并且 CSO 和 SmartProducts 还主要考虑了物理实体服务与物理世界的交互性，

PMDA 主要考虑了物理实体服务的大规模性、与物理世界的交互性和不完整性；基于 SOAP 的 Web 服务和基于 REST 的 Web 服务都考虑了物理实体服务的异构性和大规模性，但是只有 DPWS 和 SOCRADES 还考虑了物理实体服务的资源受限性，只有 SENSEI 和 IoT-A 考虑了物理实体服务的动态性，只有 SOCRADES 和 PhysicalMashups 考虑了物理实体服务的不完整性；

(2) 考虑的物理实体服务的提供者不同：以上分析的物联网软件体系结构参考模型中，有些由资源受限的感知节点直接提供物理实体服务，有些由资源相对比较丰富的网关提供物理实体服务，有些可以由节点或网关提供物理实体服务。具体来讲，Physicalnet、3CoFramework、3Tiers、SenseWeb 和 SWE/SSW 认为将物理实体服务部署在感知节点或网关上；CSO、PMDA、PhysicalMashups、SENSEI 和 IoT-A 认为只能将物理实体服务部署在网关上；SmartProducts、DPWS、SOCRADES 和 M2M 则认为将物理实体服务智能部署在感知节点上。

考虑的物理实体服务的特性和提供者不同，这些物联网软件体系结构的参考模型所具有的结构属性也不同，具体表现在以下几个方面：

(1) 协同工作模式：上述软件体系结构参考

模型中, 由于 3CoFramework、3Tiers、PMDA、SOCRADES 和 PhysicalMashups 考虑了物理实体服务的不完整性, 因此包含云端数据处理服务, 支持云端服务与物理实体服务之间的互联与互操作, 适用于复杂物理信息感知和知识提取的物联网系统的构建; 相反, Physicalnet、CSO、SmartProducts、SenseWeb、SWE/SSW、DPWS、M2M、SENSEI 和 IoT-A 没有考虑物理实体服务的不完整性, 因此不包含云端数据处理服务, 只支持物端的物理实体服务之间的互联, 适用于无需对物理信息进行复杂处理和智能分析的物联网系统的构建, 不能构成真正的云端和物端协同工作的物联网系统;

(4) 服务发现方式: 上述软件体系结构参考模型中, Physicalnet 和 PhysicalMashups 考虑以集中式目录或注册中心的方式来组织管理物端和云端的服务, 因此采用目录查找的方式就可以发现服务, 不需要专门的服务发现模块; 其它参考模型都含有服务发现模块, 以实现物端和云端服务的分布式管理与发现;

(5) 服务组合方式: 将相关的物理实体服务以及云服务, 按照一定的业务流程组合起来是构建物联网系统最为关键的一步。这一步骤可以采用人为指定服务间关系的方式来静态实现 (简称为静态组合), 也可以采用专门的组合服务模块来动态实现 (简称为动态组合)。在上述软件体系结构参考模型中, Physicalnet、CSO、SmartProducts、SenseWeb、SWE/SSW、DPWS、M2M 采用前一种方式实现服务组合, 或者没有考虑服务组合; 3CoFramework、3Tiers、PMDA、SOCRADES 和 PhysicalMashups 考虑采用后一种方式实现物理实体服务和云服务的组合; SENSEI 和 IoT-A 考虑采用后一种方式实现多个物理实体服务的组合;

(6) 应用需求描述方式: 应用需求是软件开发者或用户对物联网系统需要实现功能的精确描述, 通过应用需求模块对软件开发者或用户提出的应用需求进行解析。在上述物联网软件体系结构参考模型中, 只有 3Tiers 和 M2M 没有考虑这个模块, 其他参考模型中都将这一模块作为物联网系统提供给软件开发者或用户的接口, 比如 Physicalnet 的编程抽象模型、3CoFramework 的表达层服务组件、CSO 的输入输出组件、PMDA 的需求规划组件、SENSEI 的感知任务创建组件和 IoT-A 的业务流程处理组件等。根据这些参考模型提供的应用需求描述接口的抽象层次不同, 可以将它们分为编程抽象

和业务流抽象两种类型。

从这些物联网软件体系结构参考模型所具有的不同的结构属性可以看出, 它们适用于指导不同规模、不同应用需求的物联网应用系统的构建。软件开发者可以参照表 1 给出的比较结构, 根据所要构建的物联网应用系统的规模、物理实体服务的特点和用户需求的抽象层次, 选择合适的物联网软件体系结构参考模型。比如, 在三种基于分布式对象建立的物联网软件体系结构参考模型中, 由于 3CoFramework 支持云端和物端服务之间的协同工作模式, 采用分布式服务发现和动态服务组合方式, 并且采用基于业务流的需求描述方式, 因此它比其它两种物联网软件体系结构参考模型更加适用于指导系统规模较大、用户需求易变化的物联网应用系统的开发。

4 基于软件体系结构的物联网系统模型检测

在以上对物联网软件体系结构的属性分析中, 我们采用图形化规格说明方法给出了各软件体系结构参考模型的功能视图, 即用方框表示物联网软件的构件, 用连线表示构件之间的连接关系。参照这些图形化的物联网软件体系结构模型, 我们可以开发实现各个软件模块, 定义软件模块的接口, 最终建立可运行的物联网系统。为了在物联网系统最终建立之前验证所设计的系统是否达到用户预期的质量目标, 一般需要建立物联网软件体系结构的形式化模型, 并采用以下两种方法对基于软件体系结构的实现的物联网系统的性质进行验证:

(1) 基于软件体系结构的模型检测 (Model Checking) [55], 即利用测试准则 (如路径覆盖准则) 定义的测试需求, 生成测试用例, 基于形式化模型, 对关联构件、连接件的所有端口行为和约束进行测试;

(2) 基于软件体系结构的形式化推理 (Formal Reasoning), 即采用定理证明的基本思想, 基于形式化模型, 通过公理或推理规则进行推断与分析。

验证的物联网系统的性质一般包括结构性、功能性和非功能性三个方面。结构性是指保证所有接口应该是连接的, 数据流、控制流和命令应该都是可达的, 且在并发时应该保证无死锁的发生; 功能性包括正确性、互操作性、安全性; 非功能性分析

包括可靠性、可用性、有效性、可维护性和可移植性等。

目前常采用基于软件体系结构的模型检测方法对物联网系统的性质进行验证。为此，我们需要采用专门的软件体系结构形式化描述语言，如 Wright^[56]、ACME^[57]、UML^{[58][59]}、xADL^{[60][61]}、和 AADL^{[62][63]}等，对物联网软件体系结构进行建模，并用基本的形式化描述语言扩展这些语言的动态语义，如基于谓词逻辑的语言（如 Z、VDM）^[64]、基于抽象代数的语言（如 OBJ、CLEAR）^[65]、基于时序逻辑的语言（如 PCTL、CTL）^[66]、基于过程代数的语言（如 FSP、CSP、CCS）^[67]、基于网络的描述方法（如 PetriNet、Actor Network）^[67]和基于自动机的描述方法（如 Time Automata）^[67]，将物联网软件体系结构参考模型转换为具有精确动态语义的低层次形式化模型，以采用对应的模型检测工具，如 UPPAAL^[68]、PAT^[69]、LTSA^[70]和 PHAVer^[71]等，对物联网系统的性质进行验证。

从上一节中给出的各种物联网软件体系结构参考模型中可以看出，物联网系统中的基础软件构件包括物端的物理实体服务和云端的云服务。由于物理实体服务的行为具有连续特性，而云服务的行为具有离散特性，因此无法将只具有离散特性的互联网软件的形式化建模与模型检测方法应用于物联网系统的形式化建模与模型检测。为此，目前提出了以下两种方法：（1）统一集成建模与模型检测：在建立物理实体服务的形式化模型时将其离散化，以使物联网的物端和云端构件采用相同的形式化工具进行建模。（2）异构集成建模与模型检测：物端和云端两部分构件的建模采用不同的形式化语言，使得最终基于物联网软件体系结构参考模型建立的物联网系统形式化模型是异构模型的集成。下面对这两种方法相关的研究工作进行分析。

4.1 统一集成建模和模型检测

在表 1 中给出的十四种面向服务的物联网软件体系结构参考模型中，目前只有对基于 3CoFramework 和 PMDA 建立的物联网系统在结构性方面的性质进行了模型检测，并且它们都采用统一集成建模与模型检测的方法。

3CoFramework 采用体系结构描述语言 xSADL 建立系统的静态模型，并用基于过程代数的形式化语言 FSP 描述组件的功能和动态行为。简来讲，xSADL 用三元组（Declared_message_set, Declared_signature_set, Provided_service_set）来描

述 3CoFramework 中的各层组件，其中 Declared_message_set 和 Declared_signature_set 分别定义组件之间传递的消息类型和接口，Provided_service_set 采用 FSP 定义组件的行为。以国家农业决策支持系统（NADSS）为例，其中地图服务（MAP）和标准降雨量指数服务（SPI）分别是常用的云端和物端组件，这两个组件用 xSADL 建立的模型分别如图 12 和图 13 所示。将这两个组件组合连接起来建立新的组件（SPIMAP 服务）被称为连接组件，也用 xSADL 进行建模，建立的模型如图 14 所示。

Declared_message_set:

```
siteInfo=(siteid:string, site_x: float, site_y: float,
site_value: float);
sites=sequence(siteInfo);
mapContent=sequence(short);
mapGenerateError=(reason: string);
```

Declared_signature_set:

```
MapGenerate(input sites, output mapContent)
throw mapGenerateError;
```

Provided_service_set:

```
MapService=MapGenerate.sites -> checkSites-
Avaible -> mapGeneration ->
(MapGenerate.mapContent -> MapService |
MapGenerate.mapGenerateError -> MapService);
```

图 12 xSADL 定义的云端组件（地图服务）模型^[32]

Declared_message_set:

```
siteInfo=(siteid: string, site_x: float, site_y:
float, site_value: float);
sites=sequence(siteInfo);
spiContent=sequence(float);
spiGenerateError=(reason: string);
```

Declared_signature_set:

```
SPITempInfo(input sites, output spiContent)
throw spiGenerateError;
```

Provided_service_set:

```
SPIService=SPITempInfo.sites -> getSitesRawdata
-> spiGeneration ->
(SPITempInfo.spiContent -> SPIService |
SPITempInfo.spiGenerateError -> SPIService).
```

图 13 xSADL 定义的物端组件（SPI 服务）模型^[32]

Declared_message_set:

```

siteInfo=(siteid: string, site_x: float, site_y: float,
site_value: float);
sites=sequence(siteInfo);
mapContent=sequence(short);
mapGenerateError=(reason: string);
spiGenerateError=(reason: string);

```

Declared_signature_set:

```

SPITempInfoMapGeneration(input sites, output
mapContent) throw (spiGenerateError,
mapGenerateError);

```

Provided_service_set:

```

SPIMapService=SPITempInfoMapGeneration.sites ->
(SPITempInfoMapGeneration.mapContent ->
SPIMapService |
SPITempInfoMapGeneration.spiGenerateError ->
SPIMapService |
SPITempInfoMapGeneration.mapGenerateError ->
SPIMapService);

```

Required_service_set:

```

MapService=MapGenerate.sites -> checkSites-
Available -> mapGeneration ->
(MapGenerate.mapContent -> MapService |
MapGenerate.mapGenerateError -> MapService);
SPIService=SPITempInfo.sites -> getSitesRawdata ->
spiGeneration -> (SPITempInfo.spiContent ->
SPIService| SPITempInfo. spiGenerateError ->
SPIService);

```

Glue_set:

```

GLUE=(SPIMapService.SPITempInfoMapGeneration.
sites -> SPIService. SPITempInfo.sites -> GLUE |
SPIService. SPITempInfo. spiContent ->
MapService.MapGenerate.sites -> GLUE |
MapService.MapGenerate.mapContent ->
SPIMapService.SPITempInfoMapGeneration.
mapContent -> GLUE).

```

图 14 xSADL 定义的连接组件 (SPIMAP 服务) 模型^[32]

基于以上用 xSADL 建立的形式化模型, 在[32]中作者用模型检测工具 LTSA 验证基于 3CoFramework 建立的物联网系统的并发性、完备性和正确性。

PMDA 采用体系结构描述语言 Wright 对

Configuration IoTCON**Model AM**

```

Port Preqs( )=send_ereq->Preqs

```

Model SEM

```

Port Prdi_req( )=recv_ereq->judge_ereq-> Prdi_req

```

```

Port Psen_phy( )=sense_data-> Psen_phy

```

```

Port Pexe_phy( )=gen_exeinfo->send_exeinfo-> Pexe_phy

```

Model PM

```

Port Pact_exe( )=recv_con->Pact_exe

```

```

Port Pobj_phy( )=pro_phy-> Pobj_phy

```

Connector REQ-RDI

```

Role Rreqs( )=send_ereq->Rreqs

```

```

Role Rrdi_req( )= recv_ereq->judge_ereq-> Rrdi_req

```

```

Glue= Rreqs || Rrdi_req

```

Connector OBJ-SEN

```

Role Rsen_phy ( )=sense_data-> Rsen_phy

```

```

Role Robj_phy ( )=pro_phy->Robj_phy

```

```

Glue= Robj_phy || Rsen_phy

```

Connector EXE-ACT

```

Role Rexe_phy( )=gen_exeinfo->send_exeinfo->Rexe_phy

```

```

Role Ract_exe( )=recv_con->Ract_exe

```

```

Glue= Rexe_phy || Ract_exe

```

Instances

```

Iam:AM

```

```

Isem:SEM

```

```

Ipm:PM

```

```

Ireq-rdi: REQ-RDI

```

```

Iobj-sen: OBJ-SEN

```

```

Iexe-act: EXE-ACT

```

Attachments

```

Iam.Preqs as Ireq-rdi.Rreqs

```

```

Isem.Prdi_req as Ireq-rdi.Rrdi_req

```

```

Isem.Psen_phy as Iobj-sen.Rsen_phy

```

```

Isem.Pexe_phy as Iexe-act.Rexe_phy

```

```

Ipm.Pact_exe as Iexe-act.Ract_exe

```

```

Ipm.Pobj_phy as Iobj-sen.Robj_phy

```

End IoTCON图 15 Wright 定义的 PMDA 软件构件模型^[35]

PMDA 进行了形式化建模, 其中由物理模型 (PM)、感知执行模型 (SEM) 和应用模型 (AM) 交互连接组成的软件构件 (IoTCON) 的模型如图 15 所示。

基于以上用 Wright 建立的形式化软件构件模型, 通过模型检测工具 PAT 验证了基于 PMDA 构

建的物联网系统的有效性，即不存在死锁、发散和中止这三个性质。

此外，中国科学院和北京大学的研究人员^[72]对类似于 Physicalnet 的物联网软件体系结构参考模型（在文献[72]称之为服务框架），采用时间自动机（Time Automata）对物理环境和物联网服务进行建模，刻画出物理环境和物联网服务行为的时间属性和时间约束，在此基础上将物理环境和物联网服务一起建模为时间自动机网络，并采用 UPPAAL 模型检测工具验证了物联网服务的时效正确性。

4.2 异构集成建模和模型检测

以上研究工作都采用已有的、相同的体系结构描述语言对物联网软件体系结构中的两部分基础构件（物端的物理实体服务和云端的云服务）进行了建模，并且采用已有的模型检测工具实现基于软件体系结构的物联网系统的性质验证。可见，这种基于统一集成建模的物联网系统模型检测方法易于实现，但是由于物理实体服务的行为具有连续性，而云服务的行为具有离散性，因此它们将无法准确地反映物理实体服务的动态行为特性，从而对物联网系统整体的验证结果的准确度造成一定的影响。以下研究工作将考虑这两部分模型的异构性，对现有的建模方法和模型检测工具进行改进，以提高基于软件体系结构的物联网系统模型检测的准确度。由于物联网系统与信息物理融合系统（CPS）中的计算进程与物理进程具有相同的异构集成特性，因此下面将通过介绍目前针对 CPS 提出的形式化建模与模型检测方法，来分析未来基于软件体系结构的物联网应用系统建模与模型检测方法。

美国范德比尔特大学的研究人员^[74]采用不同的建模方法建立了 CPS 的物理环境、物理系统（硬件平台）、计算平台（软件平台）和应用软件的模型，并提出了一种基于语义集成的建模方法和语言 EsMoL，以模拟和分析在特定硬件平台和软件平台上实现的应用软件与物理环境之间的动态交互性。

美国密苏里科技大学的研究人员^[75]将能够模拟物理基础设施的领域专用模拟器和支持决策控制和并行计算的模拟器连接起来，建立一个统一的 CPS 模型检测工具，并以智慧水利系统为例，将 EPANET、Matlab 和 ns2 分别作为该系统物理部分、信息处理部分和信息传输部分的模型建立工具。

美国卡内基梅隆大学的研究人员^[76]基于 ACME 提出的一种 CPS 模型检测的方法。该方法从

信息空间、物理空间以及信息与物理空间之间的接口三个方面，对原有的软件体系结构描述语言 ACME 进行了扩展。扩展后 ACME 可以描述的信息空间部分的构件包括数据存储构件、计算构件、输入输出接口构件以及两种连接件，即调用-返回连接件和发布-订阅连接件；物理空间部分的构件包括能量源构件、能量存储构件、能量传导构件以及三种连接件，即电流、共享变量和测量；信息与物理空间之间的接口部分包括 P2C（Physical-to-Cyber）连接件和 C2P（Cyber-to-Physical）连接件。这些构件和连接件用于描述 CPS 的组成，另外采用 FSP 和 LHA 来描述 CPS 的行为，用 LTSA 和 PHAVer 来进行模型检测，并在 ACME Studio^[77]中扩展了这些的构件和连接件的描述工具以及模型检测工具。

美国卡内基梅隆大学的这些研究人员^[78]进一步分析了 CPS 基于异构模型的体系结构建模方法，提出了一种基于体系结构视图（Architecture View）的多领域建模方法。该方法先建立异构体系结构模型，然后通过异构模型与视图之间的关系（ R_M^V ）以及视图与基础体系结构之间关系（ R_{BA}^V ），将异构体系结构模型统一映射到基础体系结构（Base Architecture）。这些研究人员还对异构体系结构视图与基础体系结构之间的结构一致性进行了精确的定义，并给出了一种基于类型图匹配的结构一致性评价方法，以保证 CPS 软件体系结构建模方法的准确性^[79]。

此外，由美国加州伯克利大学的研究人员创建的 CPS 体系结构建模与模型检测集成平台 Ptolemy^{[80][81]}采用角色模型（Actor）以分层的形式将信息空间部分和物理空间部分的异构模型集成起来。Lanese 等人^[82]为准确刻画物联网的特性，即通过感知器、执行器和计算进程之间交互向用户提供有用的服务，以及节点之间通信的局部性，提出了一种新的基于进程演算（Process Calculus）的物联网系统建模语言 IoT-calculus。

4.3 物理环境建模和模型检测

从以上对信息物理融合系统与物联网系统的异构模型集成建模方法的介绍中可以看到，要想根据物联网软件体系结构对物联网系统进行模型检测时，需要建立的形式化模型除了物理空间部分的构件（物理实体服务）和信息空间部分的构件（云服务）外，还有物理环境。虽然物理环境不是物联

网软件体系结构中的一个构件,但它是驱动模型检测的基础,因为物理实体服务之间、物理实体服务与云服务之间,以及云服务之间的交互行为都是由物理环境的变化引起的。因此,目前也有研究人员对物理环境的形式化建模方法进行了专门的研究。

对物理环境的建模主要是采用结构化或形式化语言描述物理环境的时间和空间属性。可以采用的形式化语言也包括基于时序逻辑的语言(如 PCTL、CTL)、基于过程代数的语言(如 FSP、CSP、CCS)、基于网络的描述方法(如 PetriNet、Actor Network)和基于自动机的描述方法(如 Time Automata)等几种类型。例如,在健康监护应用系统 ACPS^[83]中,对物理环境建模主要是通过特征提取和数据挖掘技术从多层感知数据中获得整个环境中出现的任何异常事件(如跌倒或受伤事件等),并采用 OWL 语言对这些异常事件的出现时间和地点进行精确描述。

Tan 等人^[84]不仅考虑了物理环境中的事件,而且还考虑了系统内其它事件,包括物理观察事件、感知事件、信息物理融合事件和信息空间事件,同时采用一个三元组对事件进行描述,分别表示事件的发生事件、地点和属性,其中属性是指事件发生的条件。在文献[85]中,这些研究人员对事件模型进行了改进,通过类型、内部属性和外部属性来描述事件,其中内部属性是指事件的发生的事件、地点和条件,外部属性是指观察者监测到事件的事件和地点,并提出了一种基于概念格(Concept Lattice)的事件建模方法,以支持物理事件的组合。基于该方法建立的事件模型既适用于系统性质的离线分析,还可以作为运行时实现模型。

Jin 等人^[86]提出了一个可以描述具有复杂时空特性的事件的表达语言 OPS4ST (OncePubSub for Spatial Temporal events),该语言可以描述是十三种空间相关操作和六种时间相关操作,主要用于描述订阅服务的规则,与匹配算法相结合,将相关的时间推送给相应的应用程序。

参考以上这些关于 CPS 计算进程、物理进程和物理环境异构集成建模方法对基于软件体系结构的物联网系统进行形式化建模与模型检测,不仅能够考虑物端构件和云端构件的异构性,还能够考虑物端构件的不可靠互联性、移动性,以及与环境紧耦合性,从而提高对物联网系统的性质进行模型检测的准确度。需要指出的是,为更好地支持物联网软件体系结构的集成建模和模型检测,需要对现

有的建模方法和模型检测工具在异构应用支持、物理环境建模支持、规模可扩展性支持、移动性支持,以及与现有模型检测工具的集成这几个方面进行改进。

5 总结与展望

至此,我们对目前物联网系统设计阶段的相关工作研究进展,即物联网软件体系结构的设计,以及基于软件体系结构的物联网系统模型检测,进行了详细的分析。总结来讲,为构建支持水平互联、异构集成、资源共享和动态维护的物联网系统,同时提高物联网软件构件的可复用性和可维护性,并降低物联网应用系统的开发与维护成本,目前构建物联网系统常用面向服务的软件体系结构风格。本文以“物端”为中心的物联网系统为研究对象,对目前已经提出的四类面向服务的物联网软件体系参考模型进行了比较分析,指出了这些参考模型基于不同的软件构件类型、采用的不同设计原则及具有的不同的结构属性,因此适用于指导不同规模和不同应用需求的物联网应用系统的构建。为检测基于软件体系结构实现的物联网系统在结构性、功能性和非功能性方面是否满足开发者或用户的预期要求,需要对这些物联网体系结构参考模型进行形式化建模。本文给出了 3CoFramework 和 PMDA 的统一集成建模方法和模型检测结果,同时指出了可以参考 CPS 的形式化建模方法,对这些物联网软件体系结构中的物理实体服务、云服务和物理环境进行异构集成建模,以使模型检测结果更加精准。

未来研究人员还可以建立基于体系结构的物联网系统开发方法(Architecture Based Development[®])^{[87][88][89]},即将软件体系结构引入到物联网系统开发的各个阶段,建立全生命周期软件体系结构视图,包括需求分析阶段的需求视图、体系设计阶段的设计视图、组装阶段的实现视图、部署阶段的部署视图、维护与演化阶段的运行视图,作为软件系统开发的蓝图,利用工具支持的自动转换机制缩小从高层设计到实现的距离,而后在构件运行平台(物联网中间件)的支持下实现软件构件的自动组装和系统代码的直接生成。具体来讲,为

[®] 基于体系结构的开发(Architecture Based Development)也被称为模型驱动的开发(Model Driven Development)、基于模型的开发(Model Based Development)、模型集成开发(Model Integrated Development)和基于集成模型的开发(Integrated Model Based Development)

建立基于体系结构的物联网系统开发方法，未来还需要开展以下研究工作：

(1) 综合各类物联网应用的需求特性，形成统一的物联网软件体系结构，采用经过扩展的软件体系结构建模方法和模型检测工具对基于统一的物联网软件体系结构建立的物联网系统的性质进行验证，确保最终实现的物联网系统满足开发者或用户的共性质量要求；

(2) 将物联网软件体系结构引入到物联网系统开发的各个阶段，尤其是物联网系统的实现阶段，依据经过模型检测的物联网软件体系结构，以统一的服务实现架构建立物端和云端构件，并建立支持物联网系统开发的物端和云端中间件，形成基于软件体系结构的物联网系统开发方法；

(3) 对建立的物联网软件体系结构进行求精，为物联网系统的实现提供更细粒度的属性和特征定义，从而降低构建物联网系统的复杂度，缩小从高层设计到代码实现的距离，并对物联网软件体系结构进行支持演化的扩展设计，以适应物联网用户需求的动态变化性，降低物联网系统的维护成本。

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] Huadong Ma. Internet of Things: Objectives and Scientific Challenges. *Journal of Computer Science and Technology*, 2011, 26(6): 919-924
- [3] Zhi-Wei Xu. Cloud-Sea Computing Systems: Towards Thousand-Fold Improvement in Performance per Watt for the Coming Zettabyte Era. *Journal of Computer Science and Technology*, 2014, 29(2): 177-181
- [4] 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)
- [5] E.A. Lee. Cyber Physical Systems: Design Challenges. //*Proceedings of the 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC 2008)*, Orlando, FL, USA, 2008: 363-369
- [6] K. J. Murakami. Enabling Cyber-Physical-Social Systems (CPSS) – For A Better Future. Technical Report: Kyushu University & ISIT, 2012
- [7] L. Atzori, A. Iera, G. Morabito, M. Nitti. The Social Internet of Things (SIoT) – When Social Networks Meet the Internet of Things: Concept, Architecture and Network Characterization. *Computer Networks*, 2012, 56(11): 3594-3608
- [8] 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)
- [9] A. Zanella. Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 2014, 1(1): 22-32
- [10] M.P. Papazoglou, D. Georgakopoulos. Service Oriented Computing: Introduction. *Communication of the ACM*, 2003, 46(10): 24~28
- [11] 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
- [12] N. Koshizuka and K. Sakamura. Ubiquitous ID: Standards for Ubiquitous Computing and the Internet of Things. *IEEE Pervasive Computing*, 2010, 9(4):98-101
- [13] F. Thiesse, C. Floerkemeier, M. Harrison, F. Michahelles, and C. Roduner. Technology, Standards, and Real-World Deployments of the EPC Network. *IEEE Internet Computing*, 2009, 13(2):36~43
- [14] T. S. Lopez, D. C. Ranasinghe, M. Harrison, D. McFarlane. Adding Sense to the Internet of Things: An Architecture Framework for Smart Object Systems. *Personal and Ubiquitous Computing*, 2012, 16(3):291~308
- [15] 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, 2006:1-9
- [16] S. Tompro, 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
- [17] 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
- [18] 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
- [19] 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)
- [20] R. K. Ganti, F. Ye, and H. Lei. Mobile Crowdsensing: Current State and Future Challenges. *IEEE Communication Magazine*, 2011, 49(11): 32-39
- [21] J. Vasseur, A. Dunkels. *Interconnecting Smart Objects with IP: The Next Internet*. USA: Morgan Kaufmann Press, 2010:102-108
- [22] 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
- [23] J. Lin, S. Sedigh, and A. Miller. Modeling Cyber-Physical Systems with Semantic Agents. //Proceedings of the 34th IEEE Annual Computer Software and Applications Conference Workshops (COMPSACW 2012), Seoul, Korea, 2012: 13-18
- [24] S. Alam, M.M.R. Chowdhury, and J. Noll. SenaaS: An Event-Driven Sensor Virtualization Approach for Internet of Things Cloud. //Proceedings of the IEEE International Conference on Networked Embedded Systems for Enterprise Applications (NESEA 2010), Suzhou, China, 2010:1-6
- [25] A. Elci, B. Rahnama. Considerations on a New Software Architecture for Distributed Environments using Autonomous Semantic Agents. *Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC 2005)*, Edinburgh, Scotland, UK, 2005: 133-138
- [26] A. Dunkels. The uIP Embedded TCP/IP Stack. Technical Report: The uIP 1.0 Reference Manual, Swedish Institute of Computer Science, 2006
- [27] IETF Network Working Group. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. *IETF RFC 4919*, Aug., 2007
- [28] 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
- [29] 孙昌爱, 金茂忠, 刘超. 软件体系结构研究综述. *软件学报*, 2002, 13(7):1228-1237 (in Chinese)
(Chang'ai Sun, Maozhong Jin, Chao Liu. Overviews on Software Architecture Research. *Journal of Software*, 2002, 13(7):1228-1237)
- [30] P.B. Kruchten. The 4+1 View Model of Architecture. *IEEE Software*, 1995, 12(6):42~50
- [31] P. A. Vicaire, Z. Xie, E. Hoque, and J. A. Stankovic. Physicalnet: A Generic Framework for Managing and Programming Across Pervasive Computing Networks. Technical Report : CS-2008-2, University of Virginia, 2008
- [32] S. Zhang, S. Goddard. A Software Architecture and Framework for Web-based Distributed Decision Support Systems. *Decision Support Systems*, 2007, 43(2007): 1133-1150
- [33] 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), Yamagata, Japan, 2010: 895-900
- [34] 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
- [35] 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)
- [36] I. Agranat. Engineering Web Technologies for Embedded Applications. *IEEE Internet Computing*, 1998, 2(3): 40-45
- [37] S. de Deugd, R. Carroll, K. Kelly, B. Millett, and J. Ricker. SODA: Service Oriented Device Architecture. *IEEE Pervasive Computing*, 2006, 5(3): 94-96
- [38] Gang Li, Hongmei Sun, Zhi Li, et al. Resource Constrained Web Services. *Chinese Journal of Computers*, 2010, 33(2):193-207 (in Chinese)
(李刚, 孙红梅, 李智, 等. 资源受限 Web 服务. *计算机学报*, 2010, 33(2): 193-207)
- [39] 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
- [40] A. Kansal, S. Nath, J. Liu, and F. Zhao. SenseWeb: An Infrastructure for Shared Sensing. *IEEE Multimedia*, 2007, 14(4):8-13
- [41] M. Botts, G. Percivall, C. Reed, J. Davidson. OGC Sensor Web Enablement: Overview and High Level Architecture. OGC White Paper, Open Geospatial Consortium Inc., Wayland, MA, USA, 2007
- [42] A. P. Sheth, C. Henson, S.S. Sahoo. Semantic Sensor Web. *IEEE Internet Computing*, 2008, 12(4):78-83
- [43] Devices Profile for Web Services Version 1.1, Standard, OASIS

- WS-DD Technical Committee, 2009
- [44] L. Souza, P. Spiess, and D. Guinard, et al. SOCRADES: A Web Service Based Shop Floor Integration Infrastructure. //Proceedings of the 1nd Internet of Things Conference (IOT 2008), Stockholm, Sweden, 2008:50-67
- [45] R.T. Fielding and R.N. Taylor. Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology, 2002, 2(2):115-150
- [46] T. Luckenbach, P. Gober, 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: 1-5
- [47] W. Drytkiewicz, I. Radosch, 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, 2004: 340-348
- [48] IETF CoRE Working Group. Constrained Application Protocol (CoAP). IETF Internet Draft: draft-ietf-core-coap-04, 2004
- [49] IETF CoRE Working Group. Embedded Binary HTTP (EBHTTP). IETF Internet Draft: draft-tolle-core-ebhttp-00, 2010
- [50] 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: 1-8
- [51] Machine-to-Machine communications (M2M): Functional architecture, Technical Specification : 102 690 V1.1.1, ETSI, Oct., 2011
- [52] F. Carrez, M. Bauer, and Tim Baugé et al. SENSEI Reference Architecture. Deliverable Report : D3.2, 2009
- [53] 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
- [54] M. Bauer, N. Bui, P. Giacomini, N. Gruschka, et al. Initial Architecture Reference Model for IoT, Internet-of-Things Architecture (IoT-A). Project Deliverable: D1.2, 2011
- [55] Huimin Lin, Wenhui Zhang. Model Checking: Theories, Techniques and Applications. ACTA ELECTRONICA SINICA, 2002, 30(12A):1907- 1912 (in Chinese)
(林惠民, 张文辉. 模型检测: 理论、方法与应用. 电子学报, 2002, 30(12A):1907-1912)
- [56] N. Medvidovic, N.T. Richard. A Classification and Comparison Framework for Software Architecture Description Languages. IEEE Transactions on Software Engineering, 2000, 26(1):70~93
- [57] D. Garlan, R. T. Monroe, and D. Wile. ACME: Architectural Description of Component-Based Systems. Foundations of Component-Based Systems, UK: Cambridge University Press, 2000: 47-68.
- [58] I. Jacobson, G. Booch, and J. Rumbaugh. The Unified Software Development Process. USA: Addison Wesley Longman, 1998
- [59] Wei Yu, Wanhai Yang, Xiyao Cai. Research on Description of Software Architecture. Journal of Computer Research and Development, 2000, 37(10): 1185-1191 (in Chinese)
(于卫, 杨万海, 蔡希尧. 软件体系结构的描述方法研究. 计算机研究与发展, 2000, 37(10): 1185-1191)
- [60] E.M. Dashofy, A. van der Hoek, and R.N. Taylor. A Highly Extensible XML-based Architecture Description Language. //Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA 2001), Amsterdam, Netherlands, 2001: 103-112
- [61] E. Dashofy, A. van der Hoek, and R.N. Taylor. An Infrastructure for the Rapid Development of XML-based Architecture Description Languages, Proceedings of the 24th International Conference on Software Engineering (ICSE2002), Orlando, Florida, 2002:1-6
- [62] P. Feiler, D. Gluch, and J. Hudak. The Architecture Analysis and Design Language (AADL): An Introduction. Technical Report: CMU/SEI-2006-TN-011, Software Engineering Institute, Carnegie Mellon University, 2006
- [63] Zhibin Yang, Lei Pi, Kai Hu, Zonghua Gu, Dianfu Ma. AADL: An Architecture Design and Analysis Language for Complex Embedded Real-time Systems. Journal of Software, 2010, 21(5):899-915 (in Chinese)
(杨志斌, 皮磊, 胡凯, 顾宗华, 马殿富. 复杂嵌入式实时系统体系结构设计与分析语言:AADL. 软件学报, 2010, 21(5): 899-915)
- [64] Shengrong Zhou, Guoliang Zheng. Comparison of Z and VDM with B. Computer Science, 2002, 29(10): 136-138 (in Chinese)
(邹盛荣, 郑国梁. B 语言和方法与 Z、VDM 的比较. 计算机科学, 2002, 29(10): 136-138)
- [65] Huiqun Zhao, Guoren Wang, Yuan Gao. An Abstract Model of Software Architecture. Chinese Journal of Computers, 2002, 25(7): 730-736 (in Chinese)
(赵会群, 王国仁, 高远. 软件体系结构抽象模型. 计算机学报, 2002, 25(7): 730-736)
- [66] Yingxin Zhou, Bo Ai. A Study of Software Architecture Modeling. Journal of Software, 1998, 9(11): 866-872 (in Chinese)
(周莹新, 艾波. 软件体系结构建模研究. 软件学报, 1998, 9(11): 866-872)
- [67] Jun Qian, Yulin Feng. Semantic Models and Formal Descriptions of

- System Dynamic Behaviors. *Journal of Computer Research and Development*, 1999, 36(8): 907-914 (in Chinese)
- (钱军, 冯玉琳. 系统动态行为语义模型及其形式描述. *计算机研究与发展*, 1999, 36(8): 907-914)
- [68] J. Sun, Y. Liu, J. S. Dong. Model Checking CSP Revisited: Introducing a Process Analysis Toolkit. //*Proceedings of the 3rd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2008)*, Porto Sani, Greece, 2008: 307-322
- [69] G. Behrmann, A. David, K. G. Larsen, J. Hakansson, P. Petterson, W. Yi, and M. Hendriks. UPPAAL 4.0. //*Proceedings of the 3rd international Conference on the Quantitative Evaluation of Systems (QEST 2006)*, Washington, DC, USA, 2006: 125-126
- [70] J. Magee. Behavior Analysis of Software Architectures using LTSA. //*Proceedings of the IEEE International Conference on Software Engineering (ICSE 1999)*, Los Angeles, CA, USA, 1999: 634-637
- [71] G. Frehse. PHAVer: Algorithmic Verification of Hybrid Systems past HyTech. *International Journal on Software Tools for Technology Transfer*, 2008, 10(3): 263-279
- [72] Lixing Li, Zhi Jin, Ge Li. Modeling and Verifying Services of Internet of Things Based On Timed Automata. *Chinese Journal of Computers*, 2011, 34(8): 1365-1377 (in Chinese)
- (李力行, 金芝, 李戈. 基于时间自动机的物联网服务建模与验证. *计算机学报*, 2011, 34(8): 1365-1377)
- [73] K. Wan, K.L. Man, and D. Hughes. Specification, Analyzing Challenges and Approaches for Cyber-Physical Systems (CPS). *Engineering Letters*, 2010, 18(3): 14-21
- [74] G. Karsai and J. Sztipanovits. Model-Integrated Development of Cyber-Physical Systems. //*Proceedings of the 6th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS 2008)*, Capri Island Italy, 2008: 46-54
- [75] J. Lin, S. Sedigh, and A. Miller. Towards Integrated Simulation of Cyber-Physical Systems: A Case Study on Intelligent Water Distribution. //*Proceedings of the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC 2009)*, Chengdu, China, 2009: 690-695
- [76] A. Rajhans, S.-W. Cheng, B. Schmerl, D. Garlan, B.H. Krogh, et al. An Architectural Approach to the Design and Analysis of Cyber-Physical Systems. //*Proceedings of the 3rd International Workshop on Multi-Paradigm Modeling (MPM 2009)*, Denver, USA, 2009:1-10
- [77] B. Schmerl and D. Garlan. AcmeStudio: Supporting Style-Centered Architecture Development. //*Proceedings of the 26th International Conference on Software Engineering (ICSE 2004)*, Edinburgh, Scotland, 2004:1-10
- [78] A. Bhavé, D. Garlan, and B. Schmerl. Multi-domain Modeling of Cyber-Physical Systems using Architectural Views. //*Proceedings of the Embedded Real Time Software and Systems Conference (ERTSS 2010)*, San Diego, CA, USA, 2010:11-19
- [79] A. Bhavé, B. Krogh, D. Garlan, and B. Schmerl. View Consistency in Architectures for Cyber-Physical Systems. //*Proceedings of the 2nd ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs 2011)*, Chicago, IL, USA, 2011:21-30
- [80] E.A. Lee. Disciplined Heterogeneous Modeling. *Proceedings of the ACM/IEEE 13th International Conference on Model Driven Engineering, Languages, and Systems (MODELS 2010)*, LNCS 6395, Oslo, Norway, 2010: 273-287
- [81] P. Derler, E.A. Lee and A.L. Sangiovanni-Vincentelli. Addressing Modeling Challenges in Cyber-Physical Systems. Technical Report: UCB/EECS-2011-17, EECS Department University of California, Berkeley, 2011
- [82] I. Lanese, L. Bedogni, M. Di Felice. Internet of Things: A Process Calculus Approach. //*Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC 2013)*, Coimbra, Portugal, 2013: 1339-1346
- [83] F. Makedon, Z. Le, H. Huang, E. Becker, and D. Kosmopoulos. An Event Driven Framework for Assistive CPS Environments. *SIGBED Review*, 2009, 6(2): 3:1-9
- [84] Y. Tan, M.C. Vuran, and S. Goddard. Spatio-Temporal Event Model for Cyber- Physical Systems. //*Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW 2009)*, Montreal, Quebec, Canada, 2009: 44-50
- [85] Y. Tan, M.C. Vuran, S. Goddard, Y. Yu, M. Song, and S. Ren. A Concept Lattice-based Event Model for Cyber-Physical Systems. //*Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs 2010)*, Stockholm, Sweden, 2010: 50-60
- [86] B. Jin, H. Chen. Spatio-Temporal Events in the Internet of Things. *Proceedings of the IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2010)*, Hong Kong, China, 2010: 353-358
- [87] Hong Mei, Gang Huang, Haiyan Zhao, Wenping Jiao. An Architecture Centric Engineering Approach to Internetware. *Science in China Series E: Information Sciences*, 2006, 36(10): 1100-1126 (in Chinese)
- (梅宏, 黄罡, 赵海燕, 焦文品. 一种以软件体系结构为中心的网构软件开发方法. *中国科学 E 辑: 信息科学*, 2006, 36(10):

1100-1126)

- [88] Fuqing Yang, Jian Lv, Hong Mei. An Architecture Centric Approach to Internetware. *Science in China Series E: Information Sciences*, 2008, 38(6): 818-828 (in Chinese)
(杨芙清, 吕建, 梅宏. 网构软件技术体系: 一种以体系结构为中心的途径. *中国科学 E 辑: 信息科学*, 2008, 38(6): 818-828)



CHEN Hai-Ming, born in 1981, Ph. D., research assistant professor. He is a member of the CCF, the IEEE and the ACM. His research interests include wireless networking and networked embedded computing systems.

Background

With occurrence of smart objects, which are capable of automatically being identified, sensing context information, and interacting with people, a new integrated system called Internet of Things (IoT) is emerging. The IoT is practically an overlay system built over the Internet by connecting smart objects in physical space and computing resources in cyber space, which are regarded as frontend and backend of the IoT respectively. Currently, researchers make efforts to accumulate enabling technologies for building IoT, such as architecture prototypes, precise sensing chips, heterogeneous networking protocols, intelligent data analysis algorithms, and so on. In this paper, we make an deep overview on current researches on software architecture of IoT, which is a principal blueprint for guiding design and implementation of IoT.

So far, Service Oriented Architecture (SOA) has been widely used in constructing IoT to improve its reusability and maintainability, by which smart objects and cloud platforms are both abstracted as resources to provide services. In view of whether fundamental services of providing sensing data is located at the frontend or the backend, we divide IoT into two

- [89] Hong Mei, Gang Huang, Lu Zhang, Wei Zhang. ABC: A Method of Software Architecture Modeling in the Whole Lifecycle. *Science in China: Information Sciences*, 2014, 44(5): 564-587 (in Chinese)
(梅宏, 黄罡, 张路, 张伟. ABC: 一种全生命周期软件体系结构建模方法. *中国科学: 信息科学*, 2014, 44(5): 564-587)

CUI Li, born in 1962, Ph.D., professor and Ph. D. supervisor. She is a senior member of the CCF, and a member of the IEEE and the ACM. Her current research is focused on sensor technology, wireless sensor networks and Internet of Things.

categories, namely frontend-centric (or thing-centric) IoT and backend-centric (or cloud-centric) IoT respectively. In this paper, we take thing-centric IoT as system to be developed, and make a comparative analysis on currently proposed reference models based on SOA for IoT in terms of component type, design principle and structural characteristics. In addition, we point out that it is necessary to verify structural, functional and non-functional characteristics of IoT systems to be developed based on these reference architectures. So we also make an investigation on ways to formalize and do model checking of these reference architectures. Finally, we expect that software architecture can be used in each step of system development of IoT, and a software-architecture-based development method of IoT system will be established in future.

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 No. 61100180, and the "Strategic Priority Research Program" of the Chinese Academy of Sciences under Grant No. XDA06010403.