软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(8):1659−1670 [doi: 10.13328/j.cnki.jos.004662]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

物联网软件体系结构中的感执模型的求精*

谢开斌[1,2], 陈海明[1], 崔 莉[1]

[1](中国科学院 计算技术研究所,北京 100190)

[2](中国科学院大学,北京 100049)

通讯作者: 崔莉, E-mail: lcui@ict.ac.cn

摘 要: 物联网的基本功能是感知物理环境,并根据实时的感知信息和用户的执行需求实现对物理环境的控制.因此,感执模块是物联网软件的核心模块.为最终将软件体系结构应用于物联网软件的开发,在基于物理模型的物联网软件体系结构(physical-model driven software architecture,简称 PMDA)研究的基础上,对 PMDA 中的感执模型 SEM 进行求精,求精后的感执模型称为 R-SEM.R-SEM 考虑了物联网的特有属性,将 SEM 的内部构件按照物理应用的感执工作流程分解为用通信顺序进程(communication sequential process,简称 CSP)表达的子构件来说明构件端口的功能实现,并采用 CSP 中的导管运算符来说明子构件的端口与构件的端口之间的同步,通过 CSP 描述了子构件与子构件之间的交互.使用进程分析工具(process analysis toolkit,简称 PAT)对 R-SEM 进行了验证.验证结果表明: R-SEM 中子构件之间的交互、子构件的端口与构件的端口之间的同步保持了 SEM 所具有的保证物理应用之间有效互联的性质——不死锁、不中止和不发散.由于 R-SEM 精化了 SEM 的内部组成,且保持了 SEM 的有效互联性质,因此,R-SEM 对最终将 PMDA 应用于物联网软件的开发具有实际的指导意义.

关键词: 物联网;感执模型;构件;通信顺序进程;进程分析工具

中图法分类号: TP311

中文引用格式: 谢开斌,陈海明,崔莉.物联网软件体系结构中的感执模型的求精.软件学报,2014,25(8):1659−1670. http://www.jos.org.cn/1000-9825/4662.htm

英文引用格式: Xie KB, Chen HM, Cui L. Refinement of sense-execute model in a software architecture of Internet of things. Ruan Jian Xue Bao/Journal of Software, 2014,25(8):1659−1670 (in Chinese). http://www.jos.org.cn/1000-9825/4662.htm

Refinement of Sense-Execute Model in a Software Architecture of Internet of Things

XIE Kai-Bin[1,2], CHEN Hai-Ming[1], CUI Li[1]

[1](Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100190, China)

[2](University of Chinese Academy of Sciences, Beijing 100049, China)

Corresponding author: CUI Li, E-mail: lcui@ict.ac.cn

Abstract: Internet of things (IoT) is developed to sense and control physical environment. Its control of physical environment is based on the sensed information and the users' requirements. Therefore, sense-execute model (SEM) is the core module in software architecture of IoT. In order to finally achieve the goal of developing IoT software guided by software architecture, this paper is dedicated to refining SEM in a physical-model driven software architecture (PMDA). The refined sense-execute model is called R-SEM. R-SEM divides the component of SEM into subcomponents according to the features of IoT and the procedures of physical application. Each subcomponent illustrates the functions of a port of a component of SEM, and is expressed by the communication sequential process (CSP). Synchronization between ports of component and subcomponent is illustrated by the pipeline operation of CSP. The interaction between subcomponents is illustrated by CSP as well. R-SEM is verified by the process analysis toolkit (PAT). The result of the verification

* 基金项目: 国家自然科学基金(61100180); 国家国际科技合作专项(2013DFA10690); 中国科学院战略性先导科技专项(XDA06010403)

收稿时间: 2014-01-06; 定稿时间: 2014-04-29

© 中国科学院软件研究所 http://www.jos.org.cn

validates that R-SEM keeps the properties of SEM, namely deadlock-free, nonterminating and divergence-free, which is necessary to guarantee valid interconnections among physical applications. Since R-SEM not only refines components in SEM but also keeps the valid properties of SEM, PMDA can be finally used for guiding software development in IoT.

Key words: Internet of things; sense-execute model; component; communication sequential process; process analysis toolkit

物联网的概念自 1999 年由麻省理工学院提出以来[1],物联网在理论方面的研究[2−5]和工程方面的实践[6,7]越来越多,世界各国也制定了相应的政策来大力发展物联网[8−11].物联网与传统的网络相比,最根本的区别是物联网要对物理环境进行感知,并根据感知信息和物联网用户的需求对物理环境进行控制[12].

我们在之前的研究工作中提出了一种基于物理模型的物联网软件体系结构(PMDA)[13].PMDA 认为:物联网软件系统由多个相互关联的物理应用组成,每个物理应用由物理模型、感执模型和应用模型组成.物理模型是物理应用的感知基础和控制对象;应用模型提取用户对物联网的需求信息;感执模型连接物理模型和应用模型,实现了物理应用之间的水平互连和对物理环境的感知与控制.可见,感执模型是构建物联网软件的核心.为了最终达到将 PMDA 应用于物联网软件开发的目的,本文对 PMDA 中的感执模型(SEM)进行求精.求精后的感执模型称为 R-SEM.

简要来讲,R-SEM 考虑了物联网的特有属性,将 SEM 的内部构件按照物理应用的感执工作流程分解为用通信顺序进程(CSP)[14]表达的子构件来说明构件端口的功能的实现,并采用 CSP 中的导管运算符说明子构件的端口与构件的端口之间的同步,通过 CSP 描述了子构件与子构件之间的交互.本文使用进程分析工具(PAT)[15]对 R-SEM 进行了验证.验证的结果表明:R-SEM 中子构件之间的交互、子构件的端口与构件的端口之间的同步保持了 SEM 所具有的保证物理应用之间有效互联的性质——不死锁、不中止和不发散.由于 R-SEM 精化了 SEM 的内部组成,且保持了 SEM 的有效互联性质,因此,R-SEM 对最终将 PMDA 应用于物联网软件的开发具有实际的指导意义.

本文第 1 节介绍物联网体系结构求精方面的相关工作.第 2 节简要介绍本文工作的研究背景,即,我们之前提出的一种基于物理模型的物联网软件体系结构(PMDA)和其中的感执模型(SEM).第 3 节说明感执模型的求精过程和求精后的感执模型(R-SEM).第 4 节对 R-SEM 进行验证.第 5 节对本文的工作进行总结.

1 相关工作

目前,针对物联网软件已经提出了多种体系结构,其中,考虑了感执特征的体系结构主要有:(1) Guinard 等人[16]提出的基于 SOA 的物联网体系结构.该体系结构通过 Web Service 来组织物理世界中存在的物理服务,通过搜索机制查找与用户需求最匹配的物理服务,通过选择机制选取最符合用户需求的物理服务实例,如果查询不到能够满足用户需求的物理服务,还可以动态部署满足用户需求的物理服务.(2) Tsiatsis 等人[17]在 SENSEI 项目研究的基础上提出了物理资源的组织架构.该架构将物理资源的管理分成 4 层:无线感知与执行网络(WS&AN)物理资源层、统一资源接入及资源发现层、资源的语义查询和动态资源创建层、资源的执行管理和会话支持层.其中,WS&AN 物理资源层,资源的执行管理和会话支持体现了物联网的感执特征.

对软件体系结构的求精是指,通过反复迭代的精化,最终达到软件体系结构在软件系统上实现的目的.目前,在软件体系结构求精方面所进行的研究还比较少,主要的研究成果有:(1) Oquendo 等人[18]提出的π-ARL 体系结构精化描述语言,通过描述软件体系结构的端口、数据以及结构的精化活动,根据重写逻辑对软件体系结构逐步求精;(2) Abi-Antoun 等人[19]根据非传递无干扰理论,将构件分解成子构件,描述子构件之间基于信息流的交互,提出了体系结构中不同精化层之间保持信息的语义完整性的策略.

2 物联网软件体系结构和感执模型研究背景

在以前的研究工作中,我们提出了一种基于物理模型的物联网软件体系结构(PMDA).在该软件体系结构中,我们根据三元理论[20]将物联网软件抽象为由多个相互关联的物理应用组成的系统,每个物理应用划分为 3 个部分,每个部分由一个模型构成,分别是应用模型(AM)、感执模型(SEM)和物理模型(PM).物理模型提供物理

© 中国科学院软件研究所 http://www.jos.org.cn

数据并对物理环境进行控制;应用模型实现对社会群体需求信息的提取;感执模型连接物理模型与应用模型,实现了物理应用之间的水平互联和对物理环境的感知与控制.

例如,以一个实现室内温度和亮度监控的智能家居物联网系统为例,实现一个房间监控功能的软件为一个物理应用,多个物理应用相互关联组成一个物联网软件系统.对每个物理应用而言,其中的应用模型描述了用户对室内温度和亮度的监控需求;物理模型描述了所监控的室内环境的温度和亮度状态以及对室内环境进行控制的执行构件,比如空调的温度调节和电灯的功率调节;感执模型根据用户的需求以及感知到的温度和亮度的信息,调用物理模型内的执行构件产生相应的控制命令调节本房间的环境状态,如果本物理应用的命令执行无法满足需求,将控制命令发送到其他物理应用,调节其他房间的环境状态,最终达到用户的需求.

从以上例子可以看出:通过物理应用之间的相互关联实现物联网软件系统,感执模型的设计最为关键.我们按照感执模型的工作流程把它定义为由 6 个构件组成,分别是需求判别构件(RDI)、需求分解构件(RDE)、外部应用关联构件(OAP)、感知构件(SEN)、物理数据处理构件(PRO)和决策执行构件(EXE).6 个构件所要实现的功能描述如下:

1) 构件 RDI 接收应用模型处理后的需求信息,判别哪些需求可以由本应用完成,哪些需求不能由本应用完成,将本应用可以完成的需求发送给构件 RDE,将不能完成的需求发给构件 OAP.
2) 构件 RDE 接收构件 RDI 输出的需求信息以及其他外部应用发送的需求信息,并将这些需求信息分解为数据请求信息和控制请求信息.数据请求信息发送给构件 SEN,控制请求信息发送给构件 EXE.
3) 构件 OAP 接收构件 RDI 输出的需求信息,根据这些需求信息建立与外部应用的对应关系,并将需求信息发送给相应的外部应用,同时还能接收外部应用的请求.
4) 构件 SEN 接收构件 RDE 输出的数据请求信息,并根据数据请求信息从物理模型中获取感知数据,并将感知数据发送给构件 PRO.
5) 构件 PRO 接收来自构件 SEN 的感知数据并进行处理,将处理后的信息发送给构件 EXE.
6) 构件 EXE 接收构件 RDE 的控制请求信息和构件 PRO 发送的感知数据处理信息;根据控制请求信息与感知数据处理信息做出决策,产生决策执行信息,并将决策执行信息发送给物理模型.

图 1 表述了感执模型的 6 个构件以及构件之间通过端口的连接关系.

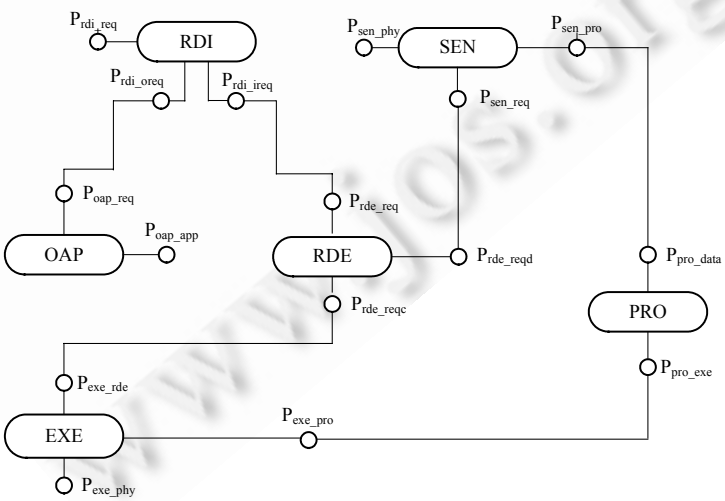

Fig.1 Relationship of components in SEM

图 1 感执模型的构件关系

基于 SEM 进行的物理应用和物联网软件开发,首先需要实现 SEM 的构件端口的功能,接着实现构件端口与端口之间的连接功能.在我们设计的基于物理模型的物联网软件体系结构(PMDA)中,给出了 SEM 中构件端

© 中国科学院软件研究所 http://www.jos.org.cn

口所应该具备的功能,但没有给出实现这些功能的具体内容,从而使得 PMDA 还不能应用于物联网的软件开发.为了最终达到软件体系结构应用于物联网软件开发的目的,需要对 SEM 进行求精.R-SEM 实现了对 SEM 的求精,求精在 SEM 的构件内部按照感执模型的工作流程、所具有的属性(空间属性、关联属性和动态属性)以及所具有的特征(感执特征、资源受限特征、数据异构特征和行业特征)分解为用 CSP 中的进程表达的子构件,以说明构件端口功能的实现.

3 感执模型求精

R-SEM 内每个构件的端口功能通过构件内部的子构件来实现,同时,构件的端口还与相应的子构件的端口保持同步.根据感执模型内的构件的工作流程,依次对感执模型内的每个构件进行求精,求精所采用的策略是根据构件在物理应用中所体现的属性特征.构件 RDI 根据物理应用的空间属性进行求精;构件 RDE 根据 SEM 本身具有的感执特征进行求精;构件 OAP 根据物理应用的动态性以及物理应用之间的关联性进行求精;构件 SEN 根据物理应用的感知资源受限特征进行求精;构件 PRO 根据物理应用中的感知数据的异构特征进行求精;构件 EXE 根据物理应用所归属的行业特征进行求精.模型 R-SEM 在求精过程中依据 3 个属性:空间属性、关联属性和动态属性以及 4 个特征:感执特征、资源受限特征、数据异构特征和行业特征.R-SEM 内的子构件的功能使用 CSP 中的进程来描述,并通过事件来说明子构件之间以及子构件与构件的端口是如何同步的.最后,通过构件内部的交互图来形象地说明构件内部的组成以及子构件之间的交互.

我们用 4 种符号来表达 R-SEM 的基本组成元素,如图 2 所示.

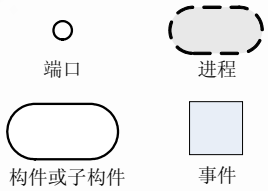

Fig.2 Basic elements in R-SEM

图 2 R-SEM 的基本元素

3.1 需求判别构件的求精

需求判别构件(RDI)通过端口 P_{rdi_req} 接收来自应用模型的需求;通过端口 P_{rdi_ireq} 向需求分解构件(RDE)提供本地需求信息;通过端口 P_{rdi_oreq} 向外部应用关联构件(OAP)提供与外部物理应用相关的需求信息.

构成物联网的物理应用具有空间属性,不同的物理应用所处的物理位置是不相同的.构件 RDI 的主要功能是判别需求信息中所包含的物理位置信息和物联网内的物理应用的物理位置信息之间的关系.我们通过对构件 RDI 实现的这一判别功能进行分解,实现对构件 RDI 的精化.精化后的判别构件(RDI)内部包括 3 个子构件:需求接收子构件(RR)、本地需求子构件(LR)和外部需求子构件(ER).这 3 个子构件分别对应需求判别子构件(RDI)的 3 个端口所提供的功能.

RR 要实现如下的功能:接收来自构件 RDI 的端口 P_{rdi_req} 传送的的需求信息和 LR 发送的本地物理位置信息;比较需求的物理位置信息与本地物理位置信息,将与本地物理位置相关的需求信息发送给子构件 LR,将与本地物理位置无关的需求信息发送给子构件 ER.用 CSP 将 RR 要实现的功能抽象为如下的事件:事件 recv_ereq 表示接收从构件 RDI 的端口 P_{rdi_req} 传送的需求信息,事件 laddr 表示接收 LR 发送的本地物理应用的物理位置信息,事件 ext_paddr 表示从需求信息中提取需求的物理位置信息,事件 comp_addr 表示对比需求物理位置信息与本地物理位置信息,事件 lreq 表示将与本地物理应用相关的信息发送给子构件 LR,事件 exreq 将与外部需求相关的信息发送给子构件 ER.用 CSP 可以将子构件 RR 表示为

$$PRR=recv_ereq\rightarrow laddr\rightarrow ext_paddr\rightarrow comp_addr\rightarrow lreq\rightarrow exreq\rightarrow PRR.$$

LR 要实现如下的功能:更新本地物理应用的位置信息,接收子构件 RR 的需求信息,并将需求信息发送给构

© 中国科学院软件研究所 http://www.jos.org.cn

件 RDI 的端口 P_{rdi_ireq}.LR 所提供的功能用可以用 CSP 抽象为如下的事件:事件 laddr 表示更新本地物理应用的位置信息,事件 lreq 表示接收子构件 RR 发送的需求信息,事件 send_inreq 表示将本地需求信息发送给构件 RDI 的端口 P_{rdi_ireq}.LR 可以用 CSP 中的进程表示为

$$PLR=laddr\rightarrow lreq\rightarrow send_inreq\rightarrow PLR.$$

ER 要实现如下的功能:接收子构件 RR 发送的需求信息,并将需求信息发送给构件 RDI 的端口 P_{rdi_oreq}.ER 所提供的功能可以用 CSP 抽象为如下的事件:事件 exreq 表示接收子构件 RR 发送的需求信息,事件 send_oreq 表示将外部需求信息发送给构件 RDI 的端口 P_{rdi_oreq}.ER 可以用 CSP 中的进程表示为

$$PER=exreq\rightarrow send_oreq\rightarrow PER.$$

进程 PRR 与进程 PLR 在执行事件 laddr 和事件 lreq 时要进行同步,同步的端口分别为 p1 和 p2.进程 PRR 与进程 PER 在执行事件 exreq 时要进行同步,同步的端口分别为 p3 和 p4.进程 PRR 在执行事件 recv_ereq 时,通过端口 p5 与构件 RDI 的端口 P_{rdi_req} 进行同步.进程 PLR 在执行事件 send_inreq 时,通过端口 p6 与构件 RDI 的端口 P_{rdi_ireq} 进行同步.进程 PER 在执行事件 send_oreq 时,通过端口 p7 与构件 RDI 的端口 P_{rdi_oreq} 进行同步.

构件 RDI 求精后的结果如图 3 所示.

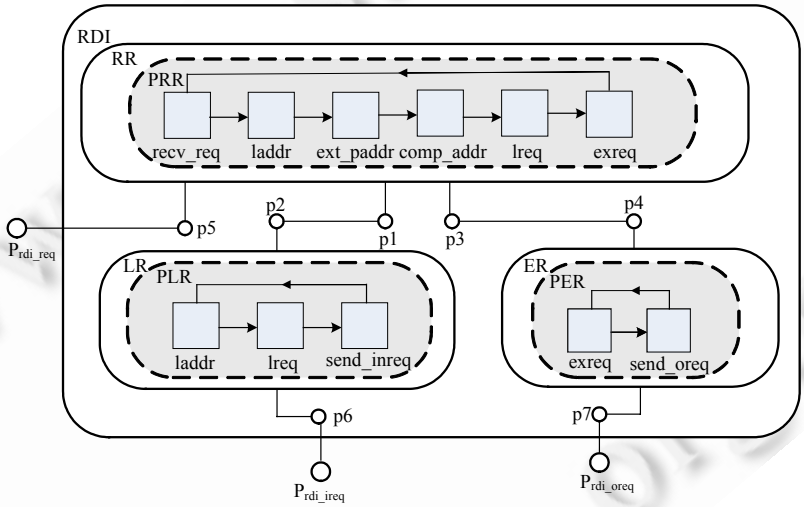

Fig.3 Refinement in component RDI

图 3 RDI 构件求精

3.2 需求分解构件的求精

需求分解构件(RDE)通过端口 P_{rde_req} 接收来自构件 RDI 的本地需求信息,通过端口 P_{rde_reqd} 向感知构件(SEN)发送感知需求信息,通过端口 P_{rde_reqc} 向决策执行构件(EXE)发送控制请求信息.

物联网中的物理应用既要感知物理环境,又要对物理环境进行控制.我们通过分解物理环境感知以及控制过程,实现对构件 RDE 的精化.精化后的需求分解构件(RDE)内部包括 3 个子构件:需求分类子构件(RC)、感知类型子构件(SC)和控制类型子构件(CC),这 3 个子构件分别对应 RDE 的 3 个端口所提供的功能.

RC 要实现如下的功能:接收构件 RDE 的端口 P_{rde_req} 传送的本地需求信息,接收来自 OAP 构件的外部需求信息,提取需求信息中的感知物理参数和需求信息中的控制参数,包括控制类型、控制值等.用 CSP 将 RC 要实现的功能抽象为如下的事件:事件 recv_inreq 表示接收 RDE 的端口 P_{rde_req} 传送的需求信息,事件 recv_oap 表示接收来自 OAP 的端口 P_{oa_oap} 传送的外部应用的需求信息,事件 phy_par 表示获取需求信息中的感知物理参数,事件 creq 表示获取需求信息中的控制需求参数.用 CSP 将子构件 RC 的行为表示为

$$PRC=recv_inreq\rightarrow recv_oap\rightarrow phy_par\rightarrow con_req\rightarrow PRC.$$

SC 要实现如下的功能:接收子构件 RC 发送的感知物理参数,将同类型的物理参数归为一类,类型由物理应

© 中国科学院软件研究所 http://www.jos.org.cn

用自身拥有的物理参数类型决定.用 CSP 将 SC 要实现的功能抽象为如下的事件:事件 phy_par 表示接收物理参数,事件 phy_cla 表示将物理参数进行归类,事件 send_datareq 表示将物理参数所归属的类型传送给 RDE 的端口 P_{rde_reqd}.用 CSP 将子构件 SC 的行为表示为

$$PSC=phy_par\rightarrow phy_cla\rightarrow send_datareq\rightarrow PSC.$$

CC 要实现如下的功能:接收子构件 RC 发送的控制需求;判断控制类型,对非精确控制类型,要给出一个符合要求的具体值;将具有明确意义的控制信息发送给构件 RDE 的端口 P_{rde_reqc}.用 CSP 将 CC 要实现的功能抽象为如下的事件:事件 creq 表示接收子构件 RC 发送过来的控制需求,事件 gen_conreq 表示生成控制需求信息,事件 send_conreq 表示将控制需求信息发送给构件 RDE 的端口 P_{rde_reqc}.用 CSP 将子构件 CC 的行为表示为

$$PCC=creq\rightarrow gen_conreq\rightarrow send_conreq\rightarrow PCC.$$

进程 PRC 与进程 PSC 在执行事件 phy_par 时要进行同步,同步的端口分别为 p8 和 p9.进程 PRC 与进程 PCC 在执行事件 creq 时要进行同步,同步的端口分别为 p10 和 p11.进程 PRC 在执行事件 recv_inreq 时,通过端口 p12 和构件 RDE 的端口 P_{rde_req} 进行同步.进程 PRC 在执行事件 recv_oap 时,通过端口 P_{oa_rde} 与 OAP 的端口 P_{oa_oap} 进行同步.进程 PSC 在执行事件 send_datareq 时,通过端口 p13 和构件 RDE 的端口 P_{rde_reqd} 进行同步.进程 PCC 在执行事件 send_conreq 时,通过端口 p14 和构件 RDE 的端口 P_{rde_reqc} 进行同步.

构件 RDE 求精后的结果如图 4 所示.

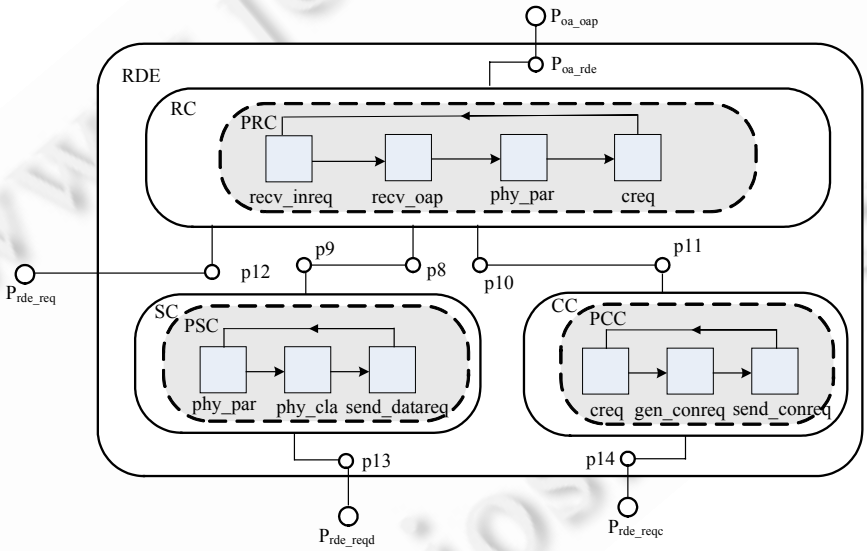

Fig.4 Refinement in component RDE

图 4 RDE 构件求精

3.3 外部应用关联构件的求精

外部应用关联构件(OAP)通过端口 P_{oap_req} 接收来自构件 RDI 的外部需求信息.

物理环境是动态变化的,当物理环境发生变化时,会使物联网内的相关物理应用进行交互以满足用户的需求.我们通过分解物理环境变化时相关联的物理应用与物理空间之间的交互过程,实现对构件 OAP 的精化.精化后的外部应用关联构件(OAP)内部包括 3 个子构件:外部应用登记子构件(PR)、外部应用分解子构件(PD)以及接收外部需求子构件(OA).这 3 个子构件分别对应 OAP 的 3 个端口所提供的功能.

PD 要实现如下的功能:接收构件 OAP 的端口 P_{oap_req} 传送的外部需求信息,提取需求信息中的物理位置信息;分解外部需求的物理位置信息,使分解后的每个物理位置都在相应的外部应用的物理位置范围内;根据分解后的物理位置信息,分解接收到的需求信息;发送分解后的物理位置信息以及与之对应的需求信息.用 CSP 将

© 中国科学院软件研究所 http://www.jos.org.cn

PD 要实现的功能抽象为如下的事件:事件 recv_oreq 表示接收构件 OAP 的端口 P_{oap_req} 传送的外部需求信息,事件 ext_phy 表示提取外部需求信息中的物理位置信息,事件 phy_preq 表示分解外部需求的物理位置和与之对应的需求信息并发送给子构件 PR.用 CSP 将子构件 PD 的行为表示为

$$PPD=recv_oreq\rightarrow ext_phy\rightarrow phy_preq\rightarrow PPD.$$

PR 要实现如下的功能:接收子构件 PD 分解后的需求信息;判断外部物理应用的物理位置是否有更新,并更新有变化的物理位置;将与外部应用相关的需求信息发送给对应的外部应用.用 CSP 将 PR 要实现的功能抽象为如下的事件:事件 phy_preq 表示接收分解后的物理位置信息以及与外部应用相关的需求信息,事件 upd_phy 表示更新外部物理应用的物理位置,事件 send_oreq 表示将需求信息发送给构件 OAP 相应的端口 P_{oap_app} 去实现.用 CSP 将子构件 PR 的行为表示为

$$PPR=phy_preq\rightarrow upd_phy\rightarrow send_oreq\rightarrow PPR.$$

OA 要实现如下的功能:接收外部应用发送的需求信息,并将外部应用发送的需求信息转发给 RDE 构件去处理.用 CSP 将 OA 要实现的功能抽象为如下的事件:事件 recv_oareq 表示接收外部其他应用的需求信息,事件 send_oareq 表示将外部应用发送的需求信息通过端口 P_{oa_oap} 发送给构件 RDE 去处理.用 CSP 将子构件 OA 的行为表示为

$$POA=recv_oareq\rightarrow send_oareq\rightarrow POA.$$

进程 PPD 与进程 PPR 在执行事件 phy_preq 时要进行同步,同步的端口分别为 p15 和 p16.进程 PPD 在执行事件 recv_oreq 时,通过端口 p17 和构件 OAP 的端口 P_{oap_req} 进行同步.进程 PPR 在执行事件 send_oreq 时,通过端口 p18 和构件 OAP 的端口 P_{oap_app} 进行同步.OA 子构件通过端口 P_{oa_oap} 与构件 RDE 的端口 P_{oa_rde} 进行同步.

构件 OAP 求精后的结果如图 5 所示.

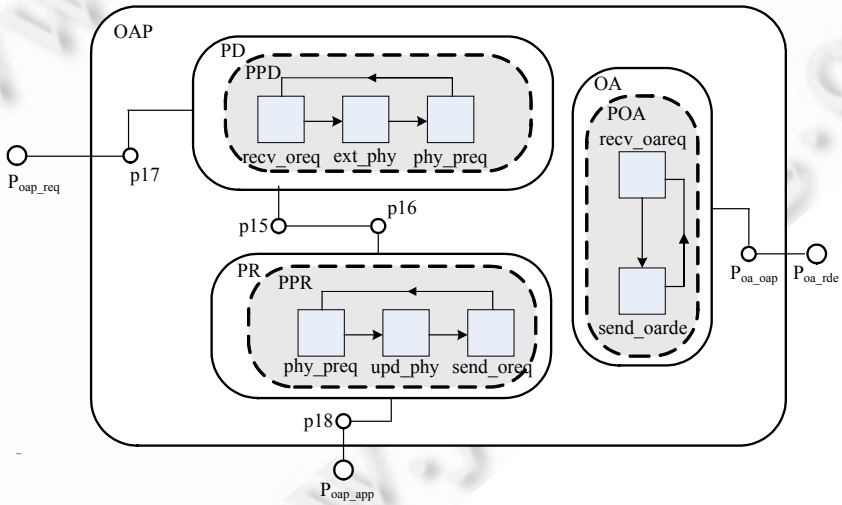

Fig.5 Refinement in component OAP

图 5 OAP 构件求精

3.4 感知构件的求精

感知构件(SEN)通过端口 P_{sen_req} 接收来自构件 RDE 的感知需求信息,通过端口 P_{sen_phy} 从物理模型中获取感知数据,通过端口 P_{sen_pro} 向处理构件(PRO)发送感知处理信息.

由于物联网中的感知资源具有资源受限的特征,因此构件 SEN 在处理感知请求时,需要同时管理感知资源.我们通过分解感知资源的管理逻辑和感知请求的处理过程来实现对构件 SEN 的精化.精化后的感知构件(SEN)内部包括两个子构件:感知资源管理子构件(SRM)和感知资源子构件(SR).这两个子构件分别对应 SEN 的 3 个端口所提供的功能.

© 中国科学院软件研究所 http://www.jos.org.cn

SRM 要实现如下的功能:接收构件 SEN 的端口 P_{sen_req} 传送的感知需求信息,根据感知需求信息与感知资源之间的对应关系,得到实现感知需求的物理资源需求信息;根据子构件 SR 内感知资源的动态变化,动态更新子构件 SRM.用 CSP 将 SRM 中要实现的功能抽象为如下的事件:事件 recv_datareq 表示接收构件 SEN 的端口 P_{sen_req} 传送的感知需求信息,事件 sr_preq 表示获取与感知需求对应的物理资源需求,事件 upd_sr 表示动态更新感知资源管理子构件.用 CSP 将子构件 SRM 的行为表示为

$$PSRM = recv_datareq \rightarrow sr_preq \rightarrow upd_sr \rightarrow PSRM.$$

SR 要实现如下的功能:接收子构件 SRM 发送的感知资源需求,动态更新感知资源的变化,根据感知资源需求接收从构件 SEN 的端口 P_{sen_phy} 传送的感知数据,将感知数据发送给构件 PRO 去处理.用 CSP 将 SR 要实现的功能抽象为如下的事件:事件 sr_preq 表示接收物理资源需求,事件 upd_sr 表示更新物理资源,事件 sense_data 表示接收构件 SEN 的端口 P_{sen_phy} 从物理模型中采集的感知数据,事件 send_data 表示将采集到的感知数据发送给构件 SEN 的端口 P_{sen_pro}.用 CSP 将子构件 SR 的行为表示为

$$PSR = sr_preq \rightarrow upd_sr \rightarrow sense_data \rightarrow send_data \rightarrow PSR.$$

进程 PSRM 与进程 PSR 在执行事件 sr_preq 和 upd_sr 时要进行同步,同步的端口分别为 p19 和 p20.进程 PSRM 在执行事件 recv_datareq 时,通过端口 p21 和构件 SEN 的端口 P_{sen_req} 进行同步.进程 PSR 在执行事件 sense_data 时,通过端口 p22 和构件 SEN 的端口 P_{sen_phy} 进行同步.进程 PSR 在执行事件 send_data 时,通过端口 p23 和构件 SEN 的端口 P_{sen_pro} 进行同步.

构件 SEN 求精后的结果如图 6 所示.

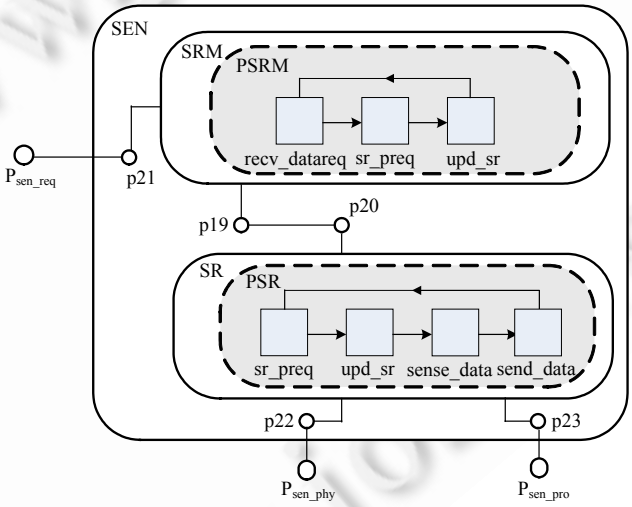

Fig.6 Refinement in component SEN

图 6 SEN 构件求精

3.5 物理数据处理构件的求精

物理数据处理构件(PRO)通过端口 P_{pro_data} 接收来自构件 SEN 的感知数据,通过端口 P_{pro_exe} 向执行构件(EXE)发送感知处理信息.

物理应用所要处理的感知数据具有异构性,因此在对感知数据进行处理和分析之前,需要选择合适的规则对感知数据性进行预处理.我们通过分解构件 PRO 完成对感知数据的预处理和处理过程,实现对构件 PRO 的精化.精化后的物理数据处理构件(PRO)内部包括两个子构件:预处理子构件(DP)和处理子构件(DI).这两个子构件分别对应 PRO 的两个端口所提供的功能.

DP 要实现如下的功能:接收构件 SEN 发送的感知数据;动态更新数据预处理规则;选择与感知数据相关的规则;根据这些规则去除感知数据中不符合逻辑、具有明显错误以及不满足用户要求等条件的“脏”数据,得到符

© 中国科学院软件研究所 http://www.jos.org.cn

合处理规则的数据.用 CSP 将 DP 要实现的功能抽象为如下的事件:事件 recv_data 表示接收构件 PRO 的端口 P_{pro_data} 传送的感知数据,事件 upd_rul 表示动态更新预处理规则,事件 sel_rul 表示选择与感知数据相关的预处理规则,pre_data 表示获取经过预处理后的感知数据.用 CSP 将子构件 DP 的行为表示为

$$PDP=recv_data\rightarrow upd_rul\rightarrow sel_rul\rightarrow pre_data\rightarrow PDP.$$

DI 要实现如下的功能:接收预处理后的感知数据,更新与感知数据相关的感知处理程序;根据感知数据的类型,选择合适的感知处理程序;将处理后的感知信息发送给构件 PRO 的端口 P_{pro_exe}.用 CSP 将 DI 要实现的功能抽象为如下的事件:事件 pre_data 表示接收预处理后的感知数据,事件 upd_pro 表示更新与感知数据相关的感知处理程序,事件 sel_pro 表示选择合适的感知数据处理程序,事件 send_info 表示将处理后获得的感知信息发送给物理数据处理构件 PRO 的端口 P_{pro_exe}.用 CSP 将子构件 DI 的行为表示为

$$PDI=pre_data\rightarrow upd_pro\rightarrow sel_pro\rightarrow send_info\rightarrow PDI.$$

进程 PDP 与进程 PDI 在执行事件 pre_data 时要进行同步,同步的端口分别为 p24 和 p25.进程 PDP 在执行事件 recv_data 时,通过端口 p26 和构件 PRO 的端口 P_{pro_data} 进行同步.进程 PDI 在执行事件 send_info 时,通过端口 p27 和构件 PRO 的端口 P_{pro_exe} 进行同步.

构件 PRO 求精后的结果如图 7 所示.

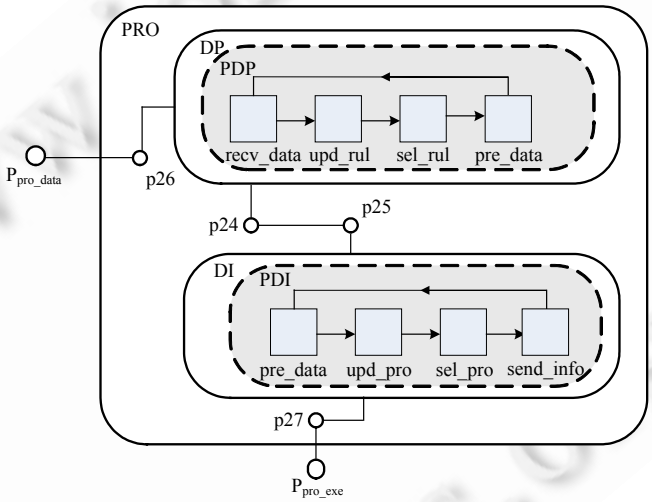

Fig.7 Refinement in component PRO

图 7 PRO 构件求精

3.6 决策执行构件的求精

决策执行构件(EXE)通过端口 P_{exe_rde} 接收来自构件 RDE 的控制需求信息,通过端口 P_{exe_pro} 接收物理数据处理构件(PRO)发送感知处理信息,通过端口 P_{exe_phy} 向物理模型发送决策执行信息.

对物理环境所要执行的操作是根据感知处理信息和控制需求信息以及由应用行业特征生成的执行规则库决定的.我们通过分解执行操作的决定过程,实现对构件 EXE 的精化.精化后的决策执行构件(EXE)内部包括子构件:决策子构件(ST).这个子构件对应 EXE 的 3 个端口所提供的功能.

ST 要实现如下的功能:接收构件 EXE 的端口 P_{exe_rde} 传送的控制需求信息,获取控制类型、控制数据等与控制相关的参数;接收构件 EXE 的端口 P_{exe_pro} 传送的感知处理信息,获取感知类型、感知信息值等与感知相关的参数;更新子构件 ST 内的用来生成决策执行信息的执行规则库;根据控制信息与感知信息来选择应该执行的规则;根据选择的规则,控制需求信息和感知处理信息来生成决策执行命令,并将决策执行命令传送给构件 EXE 的端口 P_{exe_phy}.用 CSP 将 ST 要实现的功能抽象为如下的事件:事件 recv_conreq 表示接收 EXE 的端口 P_{exe_rde} 传送的执行需求信息,事件 recv_info 表示接收 EXE 的端口 P_{exe_pro} 传送的感知处理信息,事件 upd_lib 表示更新

© 中国科学院软件研究所 http://www.jos.org.cn

执行规则库,事件 sel_esr 表示根据执行需求和感知信息来选择执行规则,事件 send_coninfo 表示传送决策执行命令给构件 EXE 的 P_{exe_phy} 端口.用 CSP 将子构件 ST 的行为表示为

PST=recv_conreq→recv_info→upd_lib→sel_ser→send_coninfo→PST.

进程 PST 在执行事件 recv_conreq 时,通过端口 p28 和构件 EXE 的端口 P_{exe_rde} 进行同步.进程 PST 在执行事件 recv_info 时,通过端口 p29 和构件 EXE 的端口 P_{exe_pro} 进行同步.进程 PST 在执行事件 send_coninfo 时,通过端口 p30 和构件 EXE 的端口 P_{exe_phy} 进行同步.

构件 EXE 求精后的结果如图 8 所示.

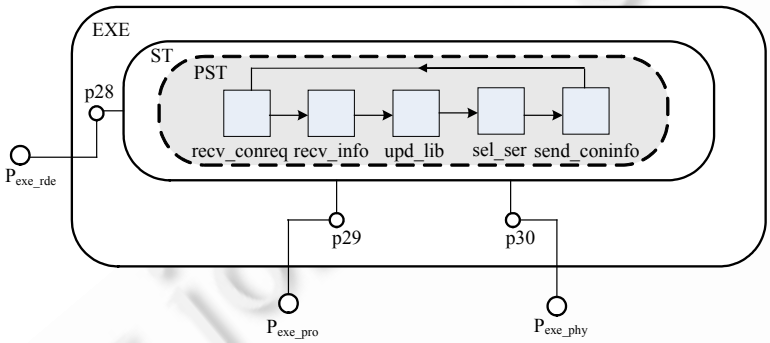

Fig.8 Refinement in component EXE

图 8 EXE 构件求精

4 R-SEM 的验证

为了验证 R-SEM 是否保持了原模型所具有的性质——不死锁、不中止和不发散,我们进行了以下两部分内容的验证:(1) 每个构件内部的子构件之间的交互保持原有模型的性质;(2) 子构件的端口与构件端口之间进行的同步保持原模型的性质.下面以 R-SEM 内的需求判别构件 RDI 为例,说明验证的过程.

首先,对构件 RDI 进行第 1 部分的验证.构件 RDI 内部有 3 个子进程 PRR,PLR 和 PER;存在着两个交互:PRR 和 PLR 之间的交互以及 PRR 和 PER 之间的交互,分别用进程 PR_LR 和进程 PR_ER 表示.相应的验证代码如下所示:

- PRR=recv_ereq→laddr→ext_paddr→comp_addr→lreq→exreq→PRR;
- PLR=laddr→lreq→send_inreq→PLR;
- PER=exreq→send_oreq→PER;
- PR_LR=PRR||PLR;
- PR_ER=PRR||PER.

用模型检验工具 PAT 验证交互进程是否满足不死锁、不中止和不发散的性质,相关的验证语句在 PAT 中可表示为

- #assert PR_LR deadlockfree;
- #assert PR_LR divergencefree;
- #assert PR_LR nonterminating;
- #assert PR_ER deadlockfree;
- #assert PR_ER divergencefree;
- #assert PR_ER nonterminating.

验证结果如图 9 所示.从图 9 可以看出,构件 RDI 的内部交互是保持感执模型的原有性质的.

© 中国科学院软件研究所 http://www.jos.org.cn

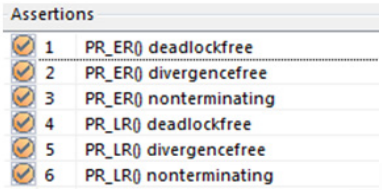

Fig.9 Result of verification about interaction inside the component RDI

图 9 构件 RDI 的内部交互验证结果

其次,对构件 RDI 进行第 2 部分的验证.子构件 RR 通过端口 p5 与构件 RDI 的端口 $P_{rdi\text{-}req}$ 进行同步,子构件 LR 通过端口 p6 与构件 RDI 在端口 P_{rdi_ireq} 进行同步,子构件 ER 通过端口 p7 与构件 RDI 在端口 P_{rdi_oreq} 进行同步.子构件的端口与构件的端口之间具有输入/输出的同步关系,假设子构件的端口上运行着进程 P,构件的端口上运行着进程 Q,则对 RDI 的第 2 部分的验证等价于验证具有导管连接的进程 P 和进程 Q 是否满足感执模型的性质.P 和 Q 通过导管连接,可用 CSP 表达为 R=P》Q.因为进程 P 和进程 Q 都不死锁和不中止,因此,进程 R 也不死锁和不中止.由于 P 和 Q 都与其他构件的进程或构件内部的进程有连接,因此,P 和 Q 进行导管运算后也不会发散,即进程 R 不会发散的.因此,构件 RDI 内部的端口与构件 RDI 的端口之间的同步也满足感执模型的性质.

因此,构件 RDI 保持了感执模型原有的性质.

同样地,可以通过上述方法对构件 RDE、构件 OAP、构件 SEN、构件 PRO 和构件 EXE 进行分析.分析结果表明,这 5 个构件都能保持感执模型原有的性质.

综上所述,感执模型经过精化后,仍然保持原有的性质.也就是说,基于求精后的感执模型设计的物理应用以及物理应用互联构成的物联网软件满足不死锁、不中止和不发散的性质.这将为物理应用的设计以及物联网软件的设计提供有效的指导.

5 结束语

本文在基于物理模型的物联网软件体系结构(PMDA)研究的基础上,根据物联网软件设计的需求,实现了对 SEM 求精的模型 R-SEM.R-SEM 考虑了物联网的特有属性,将 SEM 的内部构件按照物理应用的感执工作流程分解为用通信顺序进程(CSP)表达的子构件来说明构件端口的功能的实现,并用 CSP 中的运算符导管说明子构件的端口与构件端口是如何同步的.采用 CSP 对子构件之间的交互进行了描述,并验证了精化后的感执模型保持了原模型所具有的保证物理应用之间有效互联的性质:不死锁、不中止和不发散.由于 R-SEM 精化了 SEM 的内部组成,且保持了 SEM 的有效互联性质,因此,R-SEM 对最终将 PMDA 应用于物联网软件开发具有实际的指导意义.

下一步的工作是将 PMDA 中的其他两个模型进行精化,使得 PMDA 更加明确物联网在用户需求以及物理实体方面所需要实现的功能,并基于精化后的软件体系结构开发实际的物联网软件系统.

References:

[1] Gershenfeld N, Cohen D. Internet 0: Interdevice internetworking-end-to-end modulation for embedded networks. IEEE Circuits & Devices, 2006,22(3):48−55. [doi: 10.1109/MCD.2006.273000]

[2] Koshizuka N, Sakamura K. Ubiquitous ID: Standards for ubiquitous computing and the Internet of things. IEEE Pervasive Computing, 2010,9(4):98−101. [doi: 10.1109/MPRV.2010.87]

[3] Pujolle G. An autonomic-oriented architecture for the Internet of things. In: Proc. of the IEEE John Vincent Atanasoff 2006 Int'l Symp. on Modern Computing (JVA 2006). Sofia: IEEE, 2006. 163−168. [doi: 10.1109/JVA.2006.6]

[4] ETSI. Machine-to-Machine communications (M2M): Functional architecture. TS 102 690 V2.1.1, Nice: ETSI, 2011. 17−50.

© 中国科学院软件研究所 http://www.jos.org.cn

[5] Wu M, Lu TJ, Ling FY, Sun J, Du HY. Research on the architecture of Internet of things. In: Proc. of the 3rd Int'l Conf. on Advanced Computer Theory and Engineering (ICACTE 2010). IEEE, 2010. 484−487. [doi: 10.1109/ICACTE.2010. 5579493]

[6] EU FP7 Project. IoT-A: Internet of things-architecture. 2011. http://www.iot-a.eu

[7] EU FP7 Project. SENSEI: Integrating the physical with the digital world of the network of the future. 2009. http://www.ict-sensei.org

[8] National Intelligence Council. Six technologies with potential impacts on US interests out to 2025. Disruptive Civil Technologies Conf. Report, 2008. http://www.dni.gov/nic/confreports_disruptive_tech.html

[9] Japan IT Strategic Headquarters. i-Japan strategy 2015. 2009. http://www.kantei.go.jp/foreign/policy/it/i-JapanStrategy2015_full.pdf

[10] Commission of the European Communities. Internet of things—An action plan for Europe, Brussels. 2009. http://ec.europa.eu/information_society/policy/rfid/documents/commiot2009.pdf

[11] Ministry of Science and Technology, The People's Republic. National "12th 5-Year" plan for science & technology development planning (in Chinese). 2011. http://www.most.gov.cn/kjgh/sewkjfzgh/

[12] Chen HM, Cui L, Xie KB. A comparative study on architectures and implementation methodologies of Internet of things. Chinese Journal of Computers, 2013,36(1):168−188 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2013.00168]

[13] Xie KB, Chen HM, Cui L. PMDA: A physical model driven software architecture for Internet of things. Computer Research and Development, 2013,50(6):1185−1197 (in Chinese with English abstract).

[14] Hoare CAR. Communicating Sequential Processes. Englewood Cliffs: Prentice-Hall International, 1985. 30−200.

[15] CS Department NUS. PAT: Process analysis toolkit. 2008. http://www.patroot.com/

[16] Guinard D, Trifa V. Interacting with the SOA-based Internet of things: Discovery, query, selection, and on-demand provisioning of Web services. IEEE Trans. on Services Computing, 2010,3(3):223−235. [doi: 10.1109/TSC.2010.3]

[17] Tsiatsis V, Gluhak A, Bauge T, Montagut F, Bernat J, Bauer M, Villalonga C, Barnaghi P, Krco S. The SENSEI real world Internet architecture. In: Proc. of the Future Internet Assembly. Ghent: European Commision, 2010. 247−256.

[18] Oquendo F. π-ARL: An architecture refinement language for formally modelling the stepwise refinement of software architectures. ACM SIGSOFT Software Engineering Notes, 2004,29(5):1−20. [doi: 10.1145/1022494.1022517]

[19] Abi-Antoun M, Medvidovic N. UML'99—The Unified Modeling Language. Beyond the Standard. Berlin, Heidelberg: Springer-发 Verlag, 1999. 17−31.

[20] Easley D, Kleinberg J. Networks, Crowds, and Markets. New York: Cambridge University Press, 2010. 10−60.

附中文参考文献:

[11] 中华人民共和国科学技术部.国家"十二五"科学和技术发展规划. 2011. http://www.most.gov.cn/kjgh/sewkjfzgh/

[12] 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. [doi: 10.3724/SP.J.1016.2013.00168]

[13] 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.

谢开斌(1981－),男,山西运城人,博士生,CCF 学生会员,主要研究领域为无线传感器网络,物联网的体系结构.
E-mail: xiekaibin@ict.ac.cn

崔莉(1962－),女,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为传感器技术,无线传感器网络,物联网.
E-mail: lcui@ict.ac.cn

陈海明(1981－),男,博士,助理研究员,CCF 会员,主要研究领域为无线网络,网络化嵌入式计算系统.
E-mail: chenhaiming@ict.ac.cn

© 中国科学院软件研究所 http://www.jos.org.cn