基于时间自动机的物联网服务建模和验证

李力行[1),2)] 金 芝[1),3)] 李 戈[3)]

[1)](中国科学院数学与系统科学研究院 北京 100190)

[2)](中国科学院研究生院 北京 100049)

[3)](高可信软件技术教育部重点实验室(北京大学) 北京 100871)

摘 要 物联网服务的建模和验证是当前物联网服务提供中的一个重要问题.文中将物联网服务的行为建模为其与相关环境实体的交互,并引入环境实体以刻画物理世界各种物体的属性和行为,从而将物联网服务能力建模为它能够导致的环境实体发生的期望变化.文中以时间自动机为建模工具,分别为将要监测和要控制的物理环境实体以及不同种类的物联网服务独立建模,以表现它们的独立性和自主性.这些时间自动机形成一个网络,刻画完整的物联网服务的通信并发过程,物联网服务的实施过程表现为时间自动机网络上的状态变迁通路.最后,文中提出一组物联网服务要满足的性质,并利用模型检测工具 UPPAAL 验证物联网服务的正确性.

关键词 物联网服务;时间自动机;环境实体;服务建模;模型验证

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2011.01365

Modeling and Verifying Services of Internet of Things Based on Timed Automata

LI Li-Xing[1),2)] JIN Zhi[1),3)] LI Ge[3)]

[1)](*Academy of Mathematics and System Science, Chinese Academy of Sciences, Beijing* 100190)

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

[3)](*Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing* 100871)

Abstract The modeling and verifying of Internet of Things (IOT) services is now an important aspect of IOT software design. First, we introduce the concept of environment entities that are used to describe both the attributes and behaviors of things in the physical world. Then the behaviors of an IOT service are specified by its interaction with the corresponding environment entities, these interactions lead to the expected changes on the entities, and show the effectiveness of the IOT services. Based on timed automata, an IOT services modeling approach is proposed, in which different kinds of environment entities and IOT services are all modeled as individual timed automata. All these timed automata come into a network that represents the communication and concurrency of the whole IOT system, in which, the running of the IOT services will be represented as some computation path in the network. Based on the proposed approach, we present the properties with which the IOT services should be satisfied. By using the model-checking tool UPPAAL and the environment entities, we also present a verification approach for the correctness of IOT services models.

Keywords Internet of Things services; timed automata; environment entity; service modeling; model checking

---

收稿日期:2011-04-11;最终修改稿收到日期:2011-07-08.本课题得到国家"九七三"重点基础研究发展规划项目基金(2011CB302704)和国家自然科学基金青年科学基金项目(60803010)资助.**李力行**,男,1983 年生,博士研究生,主要研究方向为物联网服务和形式化方法.**金 芝**(通信作者),女,1962 年生,博士,教授,博士生导师,中国计算机学会高级会员,主要研究领域为需求工程、软件工程和知识工程.E-mail: zhijin@sei.pku.edu.cn.**李 戈**,男,1977 年生,博士,副教授,主要研究方向为软件工程和软件复用.

1 引 言

由 MIT Auto-ID Center 于 1999 年提出的物联网(Internet of Things)[1]通过各种信息传感设备,把任何物体与互联网相连接,进行实时信息交换和通信,实现物与物、人与物的互联,以达到智能化识别、定位、跟踪、监控和管理的目的.它被誉为继计算机、互联网之后的第三次信息技术革命.物联网的相关理论和技术已成为当前学术界的研究热点,其中一个重要的研究领域是物联网服务的按需提供,即如何根据物联网物理环境的动态变化,通过组合各种异构设备上的基础服务,提供智能化的物联网服务.

目前,已有工作将面向服务的方法应用于物联网系统的构建上,如文献[2-8].其主要贡献之一是以服务的形式描述设备的功能.设备服务为物联网服务的部署打下了很好的基础,但是要实现物联网服务的按需提供仅有设备服务是远远不够的.

首先,物联网是物理世界和信息世界的无缝融合,物联网设备直接运行于物理环境中,物联网服务不可避免得要同物理环境进行直接交互.由于物理环境的连续变化性及不确定性,对物联网服务往往有较高的实时性、动态性、可靠性和安全性要求.根据不同的物理环境,物联网服务应作出相应的、合适的反应,这种服务对环境的感应性和适应性是物联网服务按需提供需要关注的重点.

其次,时间属性在物联网服务按需提供中是重要的关注点之一.一方面,物联网服务需要关注物理世界的实时状态.各种不同类型的传感器和分布广泛的传感器网络,可以实现对物理世界中实体的实时状态获取和特定行为的监控,如二氧化碳传感器可以监测某一区域的二氧化碳浓度,GPS 可以实时传送车辆的位置等.传感器按一定的频率周期性地采集环境实时信息,物联网服务提供也应该具有与时间相关的行为.另一方面,物联网的很多应用场景很多情况下具有及时响应性,如火灾发生后,警报的及时发送和相关应急设备的及时响应.要刻画这类特殊应急服务的行为,时间属性是一个重要的方面.

我们提出一种基于环境建模的物联网服务建模方法.目的是在物联网服务建模中引入服务环境的描述,以展示物联网服务与服务环境的适配性,体现根据环境场景变化的服务提供的效果.

更进一步地,我们以时间自动机为建模工具同时为物理环境和物联网服务建模,可以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.最后借助 UPPAAL 工具验证物联网服务的时效正确性.

本文第 2 节介绍相关工作;第 3 节提出基于环境建模的物联网服务提供框架;第 4 节首先介绍时间自动机的基本概念,然后分别对环境实体和物联网服务形式化建模;第 5 节描述物联网服务与环境实体的交互,并给出基于模型检测工具 UPPAAL 的物联网服务正确性验证方法;第 6 节以智能会议室为例说明如何具体使用本文所提出的建模和验证方法;第 7 节总结全文,讨论进一步的研究方向.

2 相关工作

近年来,国际上许多研究者开始关注于面向服务的方法与物联网的结合.由于物理设备绝大多数都是资源受约束的,物联网服务需要的是一种尽量精简的服务标准,传统的 Web 服务标准(如 WSDL、SOAP 等)并不适合.DPWS(Device Profile for Web Services)[9]是关于物理设备的一个 Web 服务标准精简子集,是在保证对安全消息通信、服务描述和发现、设备事件等支持的前提下,所定义的一个轻量级的服务协议栈.它使得设备所提供的服务能无缝地集成到企业应用中,目前在物联网中正得到越来越广泛的应用.

IBM 的 Deugd 等人[2]提出了 SODA(Service-Oriented Device Architecture),其目标是为物理世界的设备提供一个高层的抽象,以服务为基础构建起物理世界与数字世界的桥梁.他们认为,利用面向服务的架构(SOA),开发者可以使用各种 Web 服务标准将企业服务通过企业服务总线(Enterprise Service Bus)连接起来,而当 SODA 出现后,开发者同样可以将由各种设备提供的服务连接到企业服务总线中,用户完全能够像访问 Web 服务一样,访问设备服务.

文献[3-5]提出的 SOCRADES,是一种基于 WEB 服务的设备现场整合架构(shop floor integration infrastructure),目标是以 Web 服务为重要连接技术将制造车间中的智能化制造设备同高层后台系统如 ERP 系统紧密地联系起来,以满足未来制造业的要求.整个体系结构由设备监测、服务发现、服务生命周期管理、安全支持等一系列关键模块构成,

从而实现了对设备的实时监测和控制，提高了生产效率和安全保障. 文章中还给出了一套具体实现方法，其中设备层次上的服务描述采用的是 DPWS.

Buckl 等人认为虽然 DPWS 相比于其它 Web 服务标准精简了许多，但是它对计算能力、资源的要求并不能被所有设备所接受. 因此需要一种更高效的方法处理资源受限的设备. 文献[6-7]提出了服务网关(Service gateway)的概念，用于 Web 服务与嵌入式系统的信息传递. 它作为互联网与嵌入式系统间的中介(mediator)，将双方所传送的信息进行解释、转换，使得双方能无缝结合，但又保留着各自的描述语言和通信协议. 文献[8]还在此基础上讨论了服务的语义集成，即把相关领域知识加入到各种服务的描述中，从而使针对用户需求的服务发现更高效、准确.

上述这些研究的关注点集中在物联网服务整体框架的建立以及服务的具体描述上，但是都没有提供服务形式化建模和正确性验证方法. 建立一种物联网服务的建模和验证方法是本文的研究动机.

在服务的形式化建模和验证方面，主要的研究工作侧重在使用的形式化理论，如自动机、Petri 网和进程代数.

文献[10]提出了一个用于分析和验证 Web 服务组合性质的框架. 在此框架下，描述 Web 服务组合的 BPEL 进程首先被转换为一种特殊形式的自动机，之后这些自动机被解释为能被模型检测工具 SPIN 所接受的 Promela 语言. 最后，使用 SPIN 验证服务组合的性质. Wombacher 等人[11]提出了一种扩展了逻辑表达式的自动机用于对服务行为形式化建模，并形式化描述了服务的消息序列，从而实现准确的服务行为匹配.

文献[12]提出了一个基于 Petri 网的 Web 服务组合设计和验证框架，该框架可用于 BPEL 进程的可视化、创建和验证. 文献[13]为 BPEL 给出了一套完整的、形式化的 Petri 网语义. 此外，文中的语法分析器还可以将 BPEL 进程自动转换为 Petri 网，这使得多种 Petri 网验证工具都可以对 BPEL 进程做自动分析.

Salaün 等人[14]认为 Web 服务的本质特点，特别是服务之间的交互，最适合用进程代数(CCS，CSP，π 演算等)来描述. 他们建立了一个基于 Web 服务与进程代数映射的通用框架，并说明了如何使用进程代数在抽象层次上对 Web 服务进行表述、组合和推导. 文献[15]结合描述逻辑和 π 演算分别对服务的静态部分(数据和架构)、动态部分(行为)建模. 文献[16]提出了使用 π 演算形式化表示服务交互模式的方法.

同上述关于 Web 服务的建模和分析工作相比，本文关注于物联网服务的建模，并结合物联网服务特点，在上述工作的基础上，增加了对物理环境及时间属性的刻画，强调物联网服务的时效性.

3　基于环境建模的物联网服务提供框架

本节我们通过分析物联网的典型应用场景，根据其性质和作用对物联网系统的各组成部分进行分类，研究各类组成部分之间的关系，最后给出一个基于环境建模的物联网服务提供框架.

3.1　一个物联网应用场景

图 1 为一个智能会议室应用场景描述，可以用它示例性说明物联网应用系统的各组成部分及其功能. 其应用需求为：通过温度感知器实时获取室内空气温度，并自动调节空调以维持舒适温度；通过光照感知器采集环境光光照强度，由日光灯和窗帘的自动开关控制调节室内光线强度；当投影仪开启后，日光灯自动关闭.

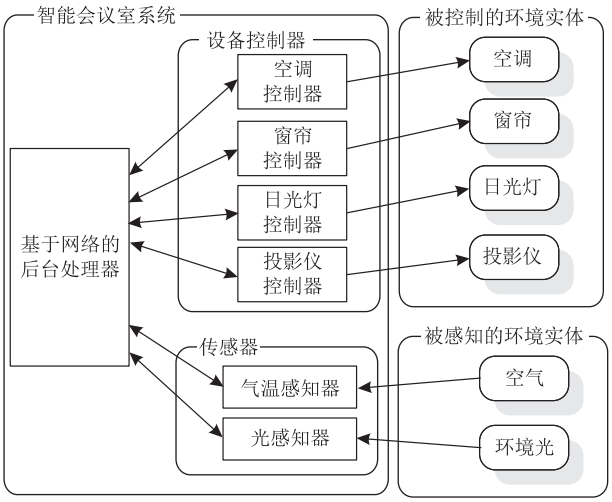

图 1　智能会议室应用场景

整个智能会议室应用场景由两大部分组成：

(1) 外部环境. 即整个智能会议系统所处的物理环境. 我们将与物联网系统发生交互的物理世界中的所有物体称为环境实体，每个环境实体都拥有自己的静态数据部分(标识、属性)和动态行为部分. 在智能会议室应用场景中涉及到的环境实体包括空调、日光灯、环境光、人等.

(2) 物联网系统. 具体由以下 3 部分组成：

① 由光感知器和温度感知器所构成的小型传感网络,用于采集会议室内环境实体的各项信息;

② 各类物理设备(空调、窗帘、日光灯、投影仪)的控制器,用于控制物理设备的动态行为;

③ 基于网络的各类后台处理器,根据所采集到环境信息,按特定处理规则向设备控制器发出合适的指令.

上述对智能会议室应用场景的分析,可以推广到其它物联网系统应用场景.

3.2 环境实体

由图 1 可以看出物联网系统与其环境的相互作用应满足应用需求,其环境由一组环境实体组成.这些环境实体分为两大类:

(1) 被感知的环境实体.这类环境实体仅可被服务以感知的方式获取状态值,但服务无法直接改变这些实体的状态,实体的状态以一种自主的、无法被外界准确预知的方式变化着.例如,智能会议室中涉及到的室内空气、光照、人等;

(2) 被控制的环境实体.这类环境实体不仅其状态值(即各项属性值)可以被服务获取,并且服务可以向这些环境实体发送事先定义好的指令,改变实体的状态,从而控制实体的状态变化.各种嵌入式设备均属于该类实体,如智能会议室中的日光灯、空调、电动窗帘等.

值得注意的是,环境实体之间存在相互关联,例如,根据会议室内空调的运行状态的变化,会议室内空气的温度、湿度发生相应变化.因此,对于被感知的环境实体,服务依然有可能以一种间接的方式影响其状态变化.

环境实体反映出物理世界的状态,而物联网服务反映出信息世界,两者的交互体现出了物理世界和信息世界的融合,这正是物联网技术的实现目标.

所有的环境实体都是独立于物联网服务而存在的,即它们并不依附于任何一个物联网服务.物联网服务的能力体现在与环境实体的交互,使环境实体的状态发生改变.针对每个特定的应用场景,服务关注的将是所有环境实体集合的一个子集.例如,在智能会议室系统中,与该系统所提供的服务交互的环境实体包括日光灯、投影仪、空调等,而在对于一个智能交通服务,其所关注的环境实体将会是各种车辆、道路、人等.一个物联网服务关注并与之交互的环境实体的集合,称为这个服务的特定环境.

3.3 物联网服务

物联网的基础是互联网,同时在此基础上延伸扩展到各种 RFID、无线传感器、全球定位系统等信息传感设备.这些信息传感设备可以提供各类基本服务,我们称为设备服务.

提供设备服务的各种物理设备部署于自然环境中,往往是资源(包括存储容量、计算能力、电量等)受约束.设备服务一般为原子服务.原子服务是指实现自包含的,在执行时不可分割,不可分解为更细粒度的服务.设备服务按其功能可分为如下两种类型:

(1) 感知型服务.这类服务通过与环境实体的交互,采集信息,感知外部环境实体的状态变化,并能将获取的信息传输给其它服务.通常情况下,感知型服务周期性地采集被感知环境实体的信息,并且单个感知型服务只提供对一类环境实体的某项属性的感知能力.

(2) 控制型服务.这类服务首先从其它服务处接收到对某个环境实体的控制信息,然后将这些信息转换为环境实体可接受的控制命令,并将其发送给相应的实体.值得注意的是这类服务只能与可控环境实体发生交互.

除了设备服务之外,物联网服务中还有一类处理型服务.这类服务并不直接与环境实体进行交互,而是负责提供业务逻辑,把从感知型服务中获取的信息进行综合处理,然后根据特定规则将控制信息发送给控制型服务.

在上述两类原子设备服务的基础上,根据处理型服务所提供的业务逻辑,通过服务的组合可以构建出满足用户不同需求的物联网服务.例如,在智能会议室场景中,由气温感知服务、光照感知服务等感知型服务实时采集到关于会议室中的各项环境信息,这些信息被及时发送给相应的处理型服务.处理型服务接收到这些不同类型的信息后,对它们进行整合、综合分析和计算,将处理后的结果发送到空调控制服务、日光灯控制服务等.这些控制型服务对接收到的信息进行翻译,然后根据预定义的规则,向空调、日光灯、窗帘等可控环境实体发出控制信息.这一系列的服务以及环境实体之间的交互,实现了对会议室的智能监控.

3.4 物联网服务提供框架

在我们以前提出的基于环境的 Web 服务建模[17-18]的基础上,本文进一步提出基于环境建模的物联网服务提供框架,如图 2 所示.

整个框架由物联网服务池、环境知识库、物联网服务建模和验证模块、物联网服务提供策略模块四大部分组成.

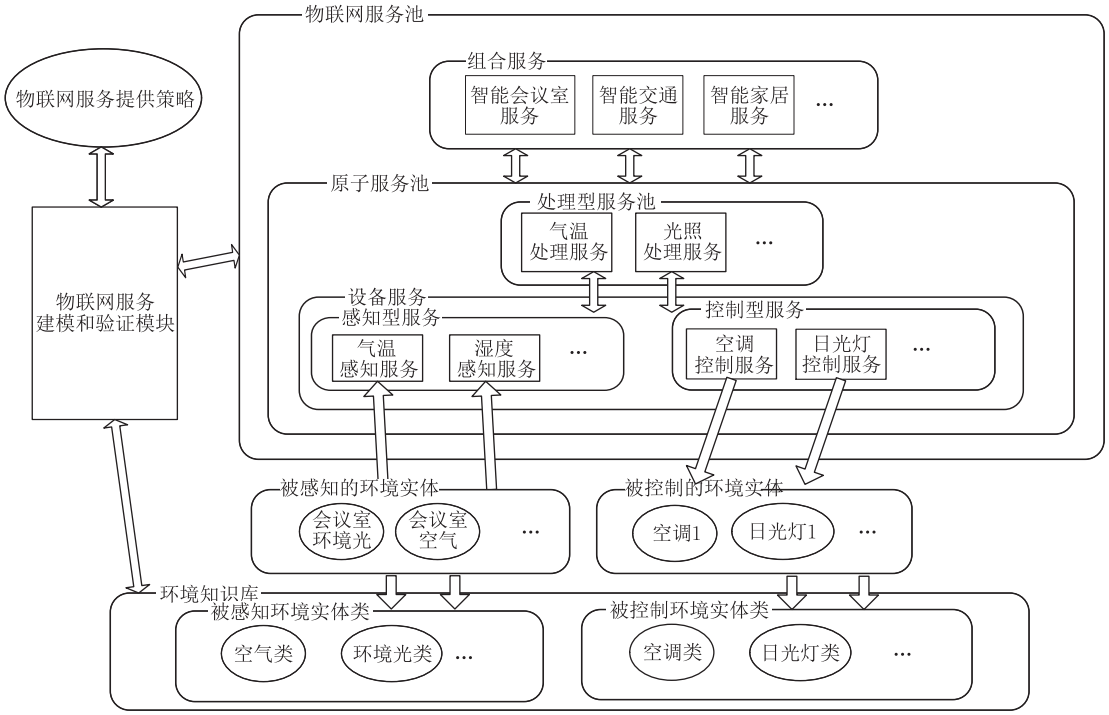

图 2　基于环境建模的物联网服务提供框架

环境知识库存放着各种环境实体的模型，包括环境实体的类型、标识、属性以及行为描述. 此外，环境实体之间的相关联系在其中被描述，例如，空调的不同运行状态可以导致附近空气温度、湿度的相应变化.

物联网服务池包括原子服务池和组合服务池两部分. 原子服务池包含设备服务和具有原子性的处理型服务组成，其中设备服务又分为感知型服务和控制型服务两大类. 组合服务池由物联网系统的综合性服务构成，它们利用设备感知服务采集被监测的环境实体的状态信息，并通过处理型服务按照预定的业务逻辑，计算需要控制的物理量，再利用设备控制服务对物理设备实施相应的控制，从而满足物联网系统的应用需求.

物联网服务建模和验证模块用于对物联网服务池中的服务进行形式化建模，并结合环境知识库中的相关环境实体的模型，对所建模的物联网服务进行正确性验证.

物联网服务提供策略存放着各种物联网服务的组合策略，它们均通过了服务建模和验证模块的严格验证，分别针对不同的应用环境，满足特定的应用需求.

4　基于时间自动机的物联网服务建模

我们以时间自动机作为环境实体和物联网服务行为的形式化建模工具. 下面先简单介绍时间自动机的基本概念.

4.1　时间自动机

由 Alur 和 Dill[19] 所提出的时间自动机，是一种用于描述、分析实时系统行为的形式化模型. 它在有限状态自动机的基础之上扩充了取实数值的时钟变量(clock variables)，用以刻画连续变化的时间. 该理论的一个重要性质是：检测时间自动机中任意一个状态是否可达(reachable)的问题是可判定的. 人们研发出多种模型检测工具，如 UPPAAL[20]、KRONOS[21] 等. 这些工具为时间自动机的行为模拟、性质自动检测提供了强有力的支持.

本文使用的时间自动机中的符号如下：

$Chan$：所有信道名的集合；

Act：所有动作的集合，包含输入、输出和内部三类动作，

$Act=\{a?\mid a\in Chan\}\cup\{a!\mid a\in Chan\}\cup\{\tau\}\}$；

$Clock$：所有的时钟变量；

Var：所有的数据变量.

在时钟变量集合 $C\subseteq Clock$ 和数据变量集合 $V\subseteq Var$ 的取值上可以建立一些约束条件，统称为保卫公式(guards)，记为 $\Phi(C,V)$，由变量集合、基本运算符和比较运算符按一定语法构成.

除了对变量取值进行约束外，还可以对变量进行赋值操作. 假设 $C\subseteq Clock$，$V\subseteq Var$，则其上的赋

值操作记为 $R(C,V)$. 时钟变量只有重置赋值,即 $c\in C$, $c:=0$,而数据变量的赋值操作形式为 $v\in V$, $v:=e$, e 可为常量或各种运算表达式. 赋值操作序列表示为 $R(C,V)^*$.

下面给出时间自动机的定义.

定义 1. 一个时间自动机是一个七元组 (S,s_0,A,C,V,I,E),其中 S 是有限状态集合;$s_0\in S$ 表示初始状态;$A\subseteq Act$ 是动作的集合;$C\subseteq Clock$ 是时钟变量的集合;$V\subseteq Var$ 是数据变量的集合;$I:S\rightarrow\Phi(C,V)$,将一个状态映射为一个保卫公式,称为状态的不变量(invariant);$E\subseteq S\times\Phi(C,V)\times A\times R(C,V)^*\times S$ 是有向边的集合,元素 $(s,\varphi,\alpha,r,s')\in E$ 表示从状态 s 到状态 s' 的有向边,边上标识有保卫公式 φ,动作 α 和赋值操作序列 r.

多个并发的时间自动机可构成一个时间自动机网络(Timed Automata Network). 网络中的多个时间自动机共享一些时钟变量和数据变量,但都有各自的状态. 格局(Configuration)是用于描述网络中所有自动机运行时的状态(即全局状态)的概念.

定义 2. 假设有 n 个时间自动机 $TA_1,\cdots,TA_n$,其中给定任意自然数 i,对 $1\leq i\leq n$, $TA_i=(S_i,s_i^0,A_i,C_i,V_i,I_i,E_i)$,由这 n 个时间自动机构成的网络记为 $N\equiv TA_1\parallel TA_2\cdots\parallel TA_n$. 一个格局被表示为一个三元组 $\langle\bar{s},\nu,\rho\rangle$,其中

$\bar{s}=\langle s_1,\cdots,s_n\rangle$,对 $1\leq i\leq n$, $s_i\in S_i$;

$v:\bigcup_{i=1}^{n}V_i\rightarrow Int$,表示数据变量的一个赋值;

$\rho:\bigcup_{i=1}^{n}C_i\rightarrow Time$,表示时钟变量的一个赋值. ρ 必须满足 $\bigwedge_{i=1}^{n}I_i(s_i)$.

当 $n=1$ 时,上述定义表示单个时间自动机的格局.

时间自动机之间不能通过信道进行传值通信,因为输入、输出动作没有附加任何数值和变量,但可以通过共享变量的方式来实现同步传值通信. 为了实现传值通信,可以首先由输出方给一个共享变量赋值,之后输入方再直接访问该共享变量,获取数值.

4.2 环境实体的建模

每个环境实体应该具备以下几个基本特征:

(1)拥有唯一的身份标识,以区别其它的环境实体;

(2)拥有一定的物理或虚拟属性,如空气具有的属性为温度、湿度等,人的属性有身高、体重等. 我们将环境实体的所有属性的一个赋值,称为环境实体的一个状态;

(3)具有通信能力或允许被感知,属性值可以通过主动或被动的方式由物联网服务或其它环境实体获取.

如果将环境实体看成对象,那么具有相同属性、动作的环境实体可以被抽象为类,我们称这些类为环境实体类,其形式化定义如下.

定义 3. 一个环境实体类表示为 $T\equiv\langle Tid,Attr,Op,Dom\rangle$,其中 Tid 为环境实体类的标识符;$Attr$ 为环境实体属性的集合;Op 为环境实体允许的操作集合;$Dom:Attr\rightarrow DataType$ 表示从属性集合到各种数据类型的一个映射,这里 $DataType$ 既包含各种基本数据类型如 Int、$Boolean$、$Float$ 等,也包含用户定义的枚举类型和复合类型.

我们将所有环境实体类的集合记为 EET,对于任意 $T\in EET$, $T.Attr$ 表示类型 T 的所有属性,$T.Op$ 表示类型 T 的所有动作.

两类不同的环境实体的行为均可以被合理地、以时间自动机的形式表示出来.

被感知的环境实体:这类环境实体的各项属性值可以被外界感知,同时它们也可能有自己的主动行为,这些行为既可能是与其它环境实体的交互,也可能是对服务所关注事件的触发. 为了表示的方便,我们假设这类环境实体都是主动地将自身的各项属性值输出. 并且它们一般都具有一个内部动作(在时间自动机中用 τ 动作对应),以外界不可见的方式对各项属性值做修改.

被控制的环境实体:这类环境实体不仅属性值可以被获取,而且通过接收指令改变相应状态,从而受控. 相对于被感知的实体,我们对被控制的实体行为的了解会更全面,同时对它们的描述往往更复杂. 根据属性值的不同可得到多个状态,状态间的变迁可由指令触发.

下面分别以被感知的环境实体类 Air 和被控制的环境实体类 $AirCond$ 为例具体说明环境实体类的定义:

$Air\equiv\langle air,\{loc,temp,humidity\},\{showLoc,showTemp,showHumi,selfAdj\},dom_{air}\rangle$,

其中,air 为 Air 实体类标识符;$loc,temp,humidity$ 分别代表 Air 类应具有 3 个属性:位置、温度、湿度;$showLoc,showTemp,showHumi$ 分别表示 Air 类将自身的各种信息告知外界的动作,$selfAdj$ 表示属于 Air 类的实体的自我调节动作,外界不可见;dom_{air} 定义为

$dom_{air}(loc)=Location$，$dom_{air}(temp)=Float$，

$dom_{air}(temp)=Float$，

这里 $Location$ 为表示地理位置的特殊数据类型，$Float$ 为浮点数类型.

$$AirCond\equiv\langle ac,\{OpMode,FanSpeed\},\{TurnOn,TurnOff,setOM,setFS\},dom_{ac}\rangle,$$

其中，ac 为 $AirCond$ 实体类型标识符；$OpMode$ 表示 $AirCond$ 类的运行模式属性，$FanSpeed$ 表示 $AirCond$ 类的风速属性；$TurnOn$，$TurnOff$，$setOM$，$setFS$ 分别表示 ac 的开、关、运行模式设定以及风速设定操作；dom_{ac} 的定义为

$dom_{ac}(OpMode)=\{COOL,HEAT,FAN\}$，

$dom_{ac}(FanSpeed)=\{LOW,MED,HIGH\}$.

根据上述定义所得到的环境实体类和环境实体就构成了基于环境建模的物联网服务提供框架中的环境知识库. 在环境实体类的基础上，可创建具体的环境实体.

定义 4.　一个环境实体表示为三元组 $e\equiv\langle Eid,T,TA\rangle$，其中 Eid 为环境实体的唯一标识符；$T\in EET$ 表示该环境实体所属的实体类；$TA=(S,s_0,A,C,V,I,E)$ 为符合定义 1 的时间自动机，用于描述环境实体的动态行为，TA 的每一个格局 $\langle s,\nu,\rho\rangle$ 表示环境实体的一个状态，并且 TA 还应满足以下要求：

(1) 存在双射 op：$T.Op\leftrightarrow A$；

(2) 存在单射 var：$T.Attr\rightarrow V$.

采用时间自动机，一个属于 Air 类的实体行为可如图 3 所示，该实体用变量 Temp 和 Humidity 分别存储温度和湿度，通过信道 showTemp 和 showTemp 将温度、湿度值告知外界.

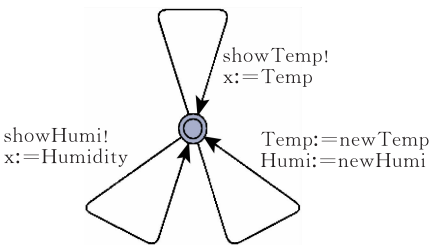

图 3　空气实体时间自动机描述

一个属于 $AirCond$ 类的环境实体可表示如图 4 所示，该实体通过信道 On 与 Off 获取开关指令，通过信道 Cool 和 Heat 接受运行模式信息.

4.3　物联网服务的建模

本节我们先给出物联网原子服务的模型，然后在此基础上给出组合服务的定义.

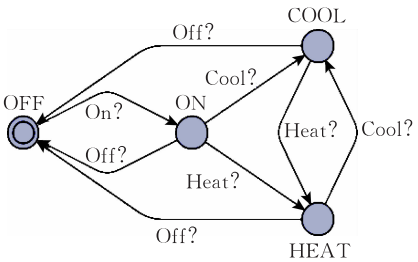

图 4　空调实体时间自动机描述

4.3.1　原子服务建模

物联网服务的功能通过其与环境之间的交互体现，服务通过交互感知到环境的状态，及时地实施对环境的控制，实现改变环境状态的目的. 在对物联网服务进行建模时需要描述与服务发生交互的环境实体类. 物联网服务形式化描述如下.

定义 5.　一个物联网原子服务形式化表示为一个三元组 $\langle Sid,Eset,STA\rangle$，其中 Sid 表示物联网服务的标识符；$Eset\subseteq EET$ 表示一个环境实体类型的集合；STA 为一个时间自动机，用于描述服务的动态行为.

三类原子服务分别有不同的表示方式：

(1) 感知型服务. 因为这类服务的功能在于感知外部环境实体的状态，所以服务的 $Eset$ 必为非空. 由于能力的限制，一般情况下，原子感知服务只涉及一类环境实体，此时 $Eset$ 中有且仅有一个元素. 感知行为被表示为一个输入动作，同时读取指定共享变量的值. 此外，感知服务需要一个时钟变量，用于控制对信息的定时获取.

例如，一个空气温度感知服务表示为 $\langle SensorA,\{Air\},SA\rangle$，其中 SA 表示如图 5 所示，该服务每 10 个单位时间感知一次空气的温度，并且可将该温度信息发送给其它服务，若出现连续 10 个单位时间没有发生感知动作，说明设备发生故障，服务进入 $Fault$ 状态.

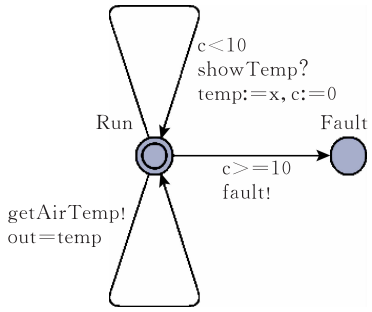

图 5　气温感知服务时间自动机描述

(2) 控制型服务. 该类型服务的控制对象为可控环境实体，因此其 $Eset$ 必须非空. 该类服务的主

要功能是通过输入动作获取其它服务发来的指令，再通过输出动作将合适的指令发送给受控环境实体.

例如一个空调控制服务定义为〈*ControlA*，{*AirCond*}，*CA*〉，其中 *CA* 的定义如图 6 所示，该服务通过信道 ACOn，ACOff，ACCool，ACHeat 从其它服务接收到空调控制信息，然后通过信道 On，Off，Cool，Heat 将相应信息发送给空调实体.

(3)处理型服务. 这类服务的特点在于并不会直接与任何环境实体交互，因此其 *Eset* 为空. 例如，一个温度调节服务表示为〈*ProcessA*，∅，*PA*〉，其中 *PA* 的描述如图 7 所示.

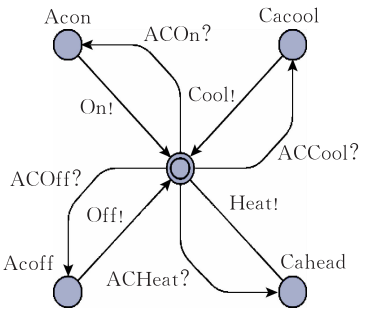

图 6 空调控制服务时间自动机描述

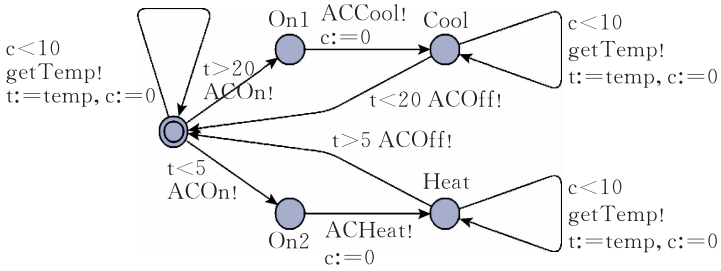

图 7 温度调节服务时间自动机描述

该服务每隔 10 个单位时间从感知服务处获取空气温度. 若温度高于 20℃，服务向空调控制服务发出启动及制冷指令，若温度低于 5℃，则发出启动及制热指令. 当空调启动后一段时间，若温度恢复正常，则发出关闭空调指令.

4.3.2 组合服务建模

以原子服务模型为基础，利用时间自动机网络可以表示出多个原子服务所构建出的组合服务，其形式化描述如下.

定义 6. 假设有 n 个物联网服务 $S_1,\cdots,S_n$，对 $i(1\leq i\leq n)$，$S_i\equiv\langle SID_i,Eset_i,STA_i\rangle$，则由这 n 个服务所组合成的服务表示为

$$S\equiv\langle SID,\bigcup_{i=1}^{n}Eset_i,STA_1\|STA_2\cdots\|STA_n\rangle,$$

其中 *SID* 为一个新的服务标识符，$S_1,\cdots,S_n$ 是 S 的成员服务.

根据时间自动机网络的操作语义[19]，网络中的各时间自动机既可以并发执行各自的动作，又可以在共享的信道上，由一方做输出动作，一方做输入动作，从而实现同步通信. 这种通信实际上就对应着成员服务之间的交互. 由于外部环境关注的是服务与环境实体的交互，而服务之间的通信为内部通信，因此这些服务通过同步通信构成了组合服务.

例如，上文提到的温度感知服务〈*SensorA*，{*Air*}，*SA*〉、空调控制服务〈*ControlA*，{*AirCond*}，*CA*〉和温度调节服务〈*ProcessA*，∅，*PA*〉，它们组合而成一个新的物联网服务：〈*Comp*，{*Air*，*AirCond*}，$SA\|CA\|PA$〉. 其中，*SA* 首先定时获取气温信息，再通过与 *PA* 的共享通道将此信息传输给 *PA*，*PA* 根据设定好的规则和实时获取的信息，将控制信息通过相应的共享通道传输给 *CA*，最终由 *CA* 控制空调的运行. 整个组合服务的功能为：根据空气温度的实时变化，自动控制空调的运行模型，从而调节温度.

5 物联网服务验证

在上文中，我们分别将环境实体和物联网服务的行为建模为时间自动机. 服务的行为能力通过与环境的交互体现出来，不应该脱离环境而对物联网服务做孤立的分析. 物联网服务的正确性不仅仅在于其自身的行为，更进一步地要在合适的环境进行合适的行为，即必须与环境的变化适配. 这一节我们讨论物联网服务的时效正确性验证.

我们将物联网服务的时效正确性分解为如下 4 类性质，可以统一采用 UPPAAL 中的时序逻辑公式[20]来表达：

(1)状态可达性(reachability). 表示物联网系统期望到达的的某个状态，存在从初始状态开始到该状态的一条运行轨迹. 状态可达主要关注环境实体的状态，表示在物联网服务的控制下，环境实体的期望状态可达. 例如希望空调实体的开启状态是可达的，其逻辑公式表示为 $\exists\Diamond air.on$；

(2) 系统安全性(safety). 我们不仅期望物联网服务在运行过程中能保证其安全性，同时也希望控制服务使得被控制的环境实体不会进入错误的状态. 系统安全性表示整个物联网系统不期望的事件永不发生，例如希望整个整个物联网系统的运行不出现死锁，表示为 $\forall\Box not\ deadlock$；

(3) 系统活性(liveness). 表示期望的事件最终能发生，一般用于描述当被感知的环境实体处于某一状态时，被控制的环境实体也最终能进入一个特定的状态. 例如希望当空气实体的温度高于 20℃时，空调实体将处于制冷状态，可表示为 $air\_temp>20\rightarrow ac.cool$.

(4) 时间约束(timed constraint). 这类性质描述中至少包含有一个时间变量，用于强调对服务行为的时间约束是否被满足. 例如，希望空调在室内温度高于 25℃后 60 s 内自动启动，则该性质表示为 $air\_temp>25\rightarrow ac.on\wedge air.c<60$，其中 $air.c$ 为时钟变量，它在温度高于 25℃时自动置零.

物联网服务的定义中包含了与之交互的环境实体类集合. 将这些类实例化后所得到的环境实体集合就构成了该服务的一个特定环境.

定义 7.　假设一个物联网服务 $S\equiv\langle sid,Eset,STA\rangle$，环境实体集合 $Env\equiv\bigcup_{i=1}^{n}\langle i,T_i,ETA_i\rangle$ 称为 S 的一个环境，若满足对于任意 $T\in Eset$，存在 $1\leq i\leq n$，$T_i=T$，即对于服务中的每个环境实体类，在环境中都至少存在该类的一个实例化环境实体. 服务 S 与它的环境 Env 之间的交互用时间自动机网络 $STA\|ETA_1\|\cdots\|ETA_n$ 的转换关系表示.

例如，前文提到的气温自动调节服务 $\langle Comp,\{Air,AirCond\},SA\|CA\|PA\rangle$，它的一个可能的运行环境由一个空气类环境实体 $\langle air,Air,TAa\rangle$ 和一个空调类环境实体 $\langle ac,AirCond,TAac\rangle$ 构成. 它们之间的交互表示为时间自动机网络

$$SA\|CA\|PA\|TAa\|TAac.$$

物联网服务与环境实体的交互行为已经完全建模为时间自动机网络，那么对服务正确性验证实际上转换为了相应时间自动机网络的分析. 我们选用模型检测工具 UPPAAL[14] 作为物联网服务时效正确性验证的工具. 在由物联网服务和其相关联的环境实体所构成的整个时间自动机网络输入到 UPPAAL 后，用 UPPAAL 协助完成下面的工作：

(1) 服务行为的模拟. 通过 UPPAAL 所提供的模拟器(simulator)，我们可以模拟运行物联网服务与环境实体的交互过程，选择希望执行的动作，并观察数据变量(记录着环境实体的属性值、服务所使用的数据值)的变化过程，整个过程都通过运行轨迹(trace)记录.

(2) 服务正确性检测. 我们将上文所提到表达系统性质的逻辑公式输入到 UPPAAL 的验证器(verifer)后，它能自动检测性质是否可被满足，并给出一些相应的运行轨迹用于分析.

由于引入了取值为实数的时钟，使时间自动机状态空间爆炸的问题更加突出. 因此，如何利用相关的优化技术来约简时间自动机的状态空间，是我们在服务建模和验证过程中必须关注的问题. 目前比较流行的时间自动机状态约简技术可分为针对时钟约束和针对路径遍历两大类.

针对时钟约束方法的基本思想是：利用状态之间时钟约束的依赖关系，减少中间状态的生成，以控制状态空间的规模. 文献[22]提出了一种约简不活跃时钟与等价时钟的技术，在模型检测的开始阶段就将不必要的时钟变量消除，从而使得验证需要的状态存储空间减少. 文献[23]提出了通过区分时间自动机时钟的最大上界与下界，来获取比时间区域更加宽泛的表达形式.

针对搜索路径的研究中最具代表性的是偏序约简技术. 其基本思想是在检验过程中，对于独立的转换，有许多不同的转换的交叉组合，因为独立转换的发生次序对结果没影响，所以可以用一条代表性的路径来替代多个交叉组合. 文献[24]提出了一种方法消除时间自动机网络中隐含的时钟同步信息，将标准的偏序约简技术应用于时间自动机上. 文献[25]在此基础上，将偏序技术用到了针对时间自动机时间发生的 LTL 模型检验上.

上述方法均可以约简时间自动机的状态空间. 我们将考虑结合使用这多种优化方法，提高模型验证的效率.

6　案例研究

本节以智能会议室应用场景为例，说明如何基于环境知识库，采用时间自动机对环境实体和物联网服务进行建模，并以 UPPAAL 为工具对物联网服务正确性验证.

假设一个智能会议室场景为：该会议室中装有一盏日光灯和一台投影仪. 日光灯的开关应按如下要求被自动控制：当投影仪开启时，日光灯自动关

闭;当投影仪关闭,环境光线由亮变暗时,日光灯在20 s内将会自动开启;当环境光由暗变亮时,日光灯在20 s内自动关闭.其建模过程如下:

1. 创建环境实体类

被感知的环境实体类:

$Light \equiv \langle Lit, \{Intensity\}, \{showInt\}, Intensity \mapsto Bool\rangle$,表示环境光类型.其中 *Intensity* 表示环境光的光照强度,这里假设它只有强和弱两个取值,因此它的值域类型是 *Bool*.

被控制的环境实体类:

$Lamp \equiv \langle Lamp, \varnothing, \{lon, loff\}, dom_l\rangle$,表示日光灯类型;

$Projector \equiv \langle Proj, \varnothing, \{pon, poff\}, dom_p\rangle$,表示投影仪类型.

2. 创建环境实体

实例化环境实体类,获得被感知的环境实体和被控制的环境实体:

$\langle lamp, Lamp, TA_{lamp}\rangle$,表示一个日光灯实体,其行为描述 TA_{lamp} 如图8(a)所示.这里假设日光灯从 *lon* 和 *loff* 两个信道分别接收开启和关闭的指令;

$\langle light, Light, TA_{light}\rangle$,表示一个环境光实体,其行为描述 TA_{light} 如图8(b)所示.这里假设环境光每隔一个固定的时间就会变换一次光照强度,用变量 intensity 记录光照的强弱,并且可通过信道 showInt 输出光照强弱值;

$\langle projector, Projector, TA_P\rangle$,表示一个投影仪实体,其行为描述 TA_P 如图8(c)所示.这里假设投影仪每隔一个固定的时间被轮流开启、关闭,并通过信道 pon 和 poff 表明自身开启或关闭的状态.

3. 为物联网服务建模

上述场景蕴含4个物联网服务:

$\langle s_1, \{Light\}, S_1\rangle$,表示光照感知服务,其行为描述 S_1 如图8(d)所示,该服务定时通过 showInt 信道获取光照强度值,存放于变量 l_int 中,并根据光照强度的不同分别通过信道 dark 和 bright 与其它服务通信;

$\langle c_1, \{Lamp\}, C_1\rangle$,表示日光灯控制服务,其行为描述 C_1 如图8(e)所示,该服务通过 turnOnLamp 和 turnOffLamp 两个信道与其它服务通信,并及时将对日光灯的开关指令分别通过信道 lon,loff 发出.注意到收到消息后的瞬时开关灯指令发出动作可通过 UPPAAL 所提供的 Committed 状态实现(图中状态标为 *C*);

$\langle p_1, \{Proj\}, P_1\rangle$,表示处理服务1,其行为描述 P_1 如图8(f)表示.若该服务从信道 pon 接收到投影仪开启的消息,则立即通过信道 turnOffLamp 发出日光灯控制服务发出信号;

$\langle p_2, \varnothing, P_2\rangle$,表示处理服务2,其行为描述 P_2 如图8(g)所示,该服务通过信道 bright 和 dark 获取环境光照的强弱变化情况,当光照强度变弱时,它通过信道 turnOnLamp 向日光灯控制服务发出信号;当光照强度变强时,它通过 turnOffLamp 向日光灯控制服务发出信号.

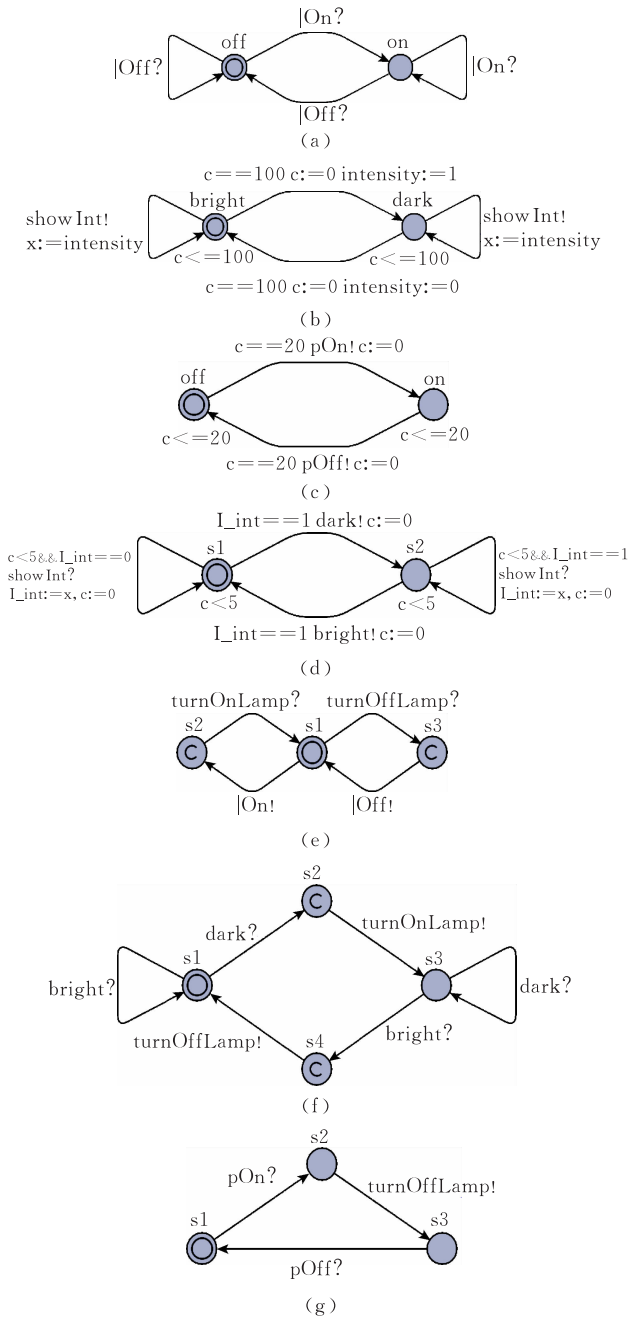

图8 智能会议室时间自动机网络

上述4个服务组合在一起构成了智能会议室服务,其行为表示为时间自动机网络:

$$S_1 \| C_1 \| P_1 \| P_2.$$

4. 验证物联网服务

对物联网服务的时效正确性进行验证,需要将该服务与相关环境实体组合成一个整体系统:

$$S_1 \| C_1 \| P_1 \| P_2 \| light \| projector \| lamp.$$

首先我们希望验证一条重要的服务安全特性:服务在运行过程中不停机,即整个系统不出现死锁,该性质表示为时序逻辑公式:$\forall\Box$not deadlock.

将该公式输入到 UPPAAL 的验证器中,得到

的结果是性质满足. 说明上述模型描述的智能会议室服务可正常运行.

关于日光灯控制的 3 条性质需要分别转化为以下 3 条逻辑公式：

(1) 当投影仪打开时，日光灯自动关闭：

$$projector.on \rightarrow lamp.off;$$

(2) 当投影仪关闭，环境光线由亮变暗时，日光灯在 20 s 内将会自动开启：

$$light.dark \wedge projector.off \rightarrow lamp.on \wedge light.c < 20.$$

(3) 当环境光由暗变亮时，日光灯在 20 s 内自动关闭：

$$light.bright \rightarrow lamp.off \wedge light.c < 20.$$

这 3 条性质中第 1 条属于活性性质，后两条属于时间约束. 下面我们验证其中的第 1 条性质.

将第 1 条性质要求所对应的公式输入到 UPPAAL的验证器中，得到的结果是该性质不能被满足. UPPAAL 给出了一个导致性质不满足的运行轨迹，如图 9 所示.

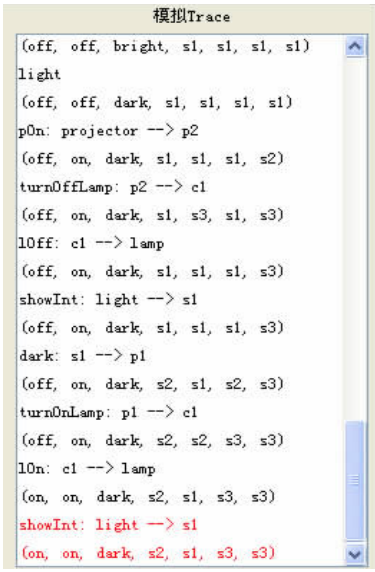

图 9　智能会议室时间自动机网络运行轨迹示例

图 9 所表达的运行轨迹中，偶数行表示通信动作，奇数行记录着各时间自动机状态的变化，其中括号内前 3 项依次记录着环境实体 *Lamp*，*projector* 和 *light* 的状态. 通过对该运行轨迹的分析可以发现，导致错误发生的情况是：当 *projector* 处于开启状态后，尽管服务 p_1 将日光灯关闭，但是由于光线的变暗，服务 p_2 又将使日光灯重新开启. 解决这一问题的方法是，为服务 p_2 新增一个状态(如图 10)，当投影仪开启后，服务进入到该新状态，并不控制日光灯，直到投影仪关闭.

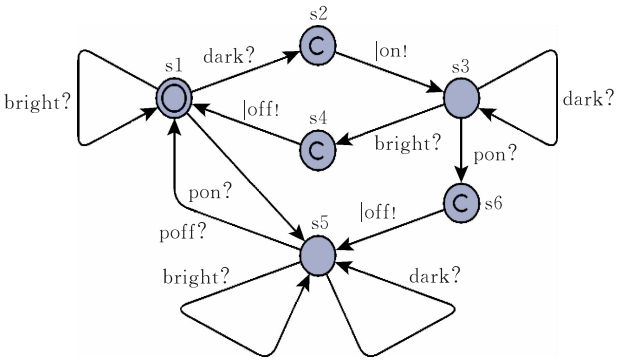

图 10　修改后的控制服务 p_2

根据以上对控制服务 p_2 的修改，通过 UPPAAL 可验证上述性质被满足. 采用同样方法可以验证上面提到的物联网服务的另外两条时间约束相关性质的可满足性. 从而说明，上述设计的物联网服务满足时效正确性.

7　结束语

物联网是物理世界和信息世界的无缝融合，物联网服务的正确建模和验证需要准确地描述物理世界中的物理实体. 基于以上观点，本文提出了一个基于环境建模的物联网服务提供框架. 在此框架中，环境实体作为一个重要的概念被引入，用于刻画物理世界中各种物体的属性和行为. 物联网服务的行为则通过与环境实体的交互体现出来. 本文以时间自动机为工具，对环境实体和物联网服务的行为进行建模，并用时间自动机网络来刻画服务的组合和它们与环境实体的交互. 本文以 UPPAAL 为工具并以智能会议室为例，展示了物联网服务建模和验证的完整过程.

以 UPPAAL 为工具对时间自动机网络做模型检测时，随着时间自动机数量以及自动机内状态的数量、时钟变量的增多，验证过程的复杂性将增高、耗时增加，甚至有可能由于运算资源耗尽，导致验证无法顺利完成. 因此，下一步工作的重点是刻画出一种合理的服务行为等价关系及服务优化方法，使服务在功能等价的情况下，状态尽可能精简，从而提高正确性验证的效率. 其次，本文中所涉及的物联网服务组合是一种静态组合方式，即无法直接根据用户所提出的需求由原子服务动态组合出满足要求的服务. 因此，在本文的基础上，构思一种物联网服务按需动态组合方法也是我们关注的方向. 此外，我们还将进一步研究环境知识库的构建、管理和应用.

参 考 文 献

[1] Sarma S, Brock D, Ashton K. The Networked Physical World. Technical Report MIT-AUTOID-WH-001, 1999

[2] Deugd S, Carroll, Kelly K, Millett, Ricker J. SODA: Service oriented device architecture. IEEE Pervasive Computing, 2006, 5(3): 94-96

[3] Souza L, Spiess P, Guinard D, Khler M, Karnouskos S, Savio D. SOCRADES: A Web service based shop floor integration infrastructure//Proceedings of the Internet of Things 2008(IOT'08). Zurich, Switzerland, 2008: 50-67

[4] Spiess P, Karnouskos S, Guinard D, Savio D, Baecher O, Souza L, Trifa V. SOA-based Integration of the Internet of Things in enterprise services//Proceedings of the IEEE International Conference of Web Services (ICWS'09). Los Angeles, USA, 2009: 968-975

[5] Guinard D, Trifa V, Karnouskos S, Spiess P, Savio D. Interacting with SOA-based Internet of Things: Discovery, query, selection, and on-demand provisioning of Web services. IEEE Transactions on Services Computing, 2010, 3(3): 223-235

[6] Buckl C, Sommer S, Scholz, Knoll A, Kemper A. Generating a tailored middleware for wireless sensor network applications//Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing(SUTC'08). Taichung, China, 2008: 162-169

[7] Buckl C, Sommer S, Scholz A, Knoll A, Kemper A. Services to the field: An approach for resource constrained sensor/actor networks//Proceedings of the 4th Workshop on Service Oriented Architectures in Converging Networked Environments(SOCNE'09). Bradford, UK, 2009: 476-481

[8] Sommer S, Scholz, Buckl C, Kemper A, Knoll A, Heuer J, Schimtt A. Towards the Internet of Things: Integration of Web services and field level devices//Proceedings of the International Workshop on the Future Internet of Things and Services-Embedded Web Services for Pervasive Devices (FITS' 2009). Berlin, Germany, 2009

[9] Devices Profile for Web Services. http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01

[10] Fu X, Bultan T, Su J. Analysis of interacting BPEL Web services//Proceedings of the 13th International Conference on the World Wide Web(WWW'2004). New York, USA, 2004: 621-630

[11] Wombacher A, Fankhauser P, Mahleko P, Neuhold B. Matchmaking for business based on choreographies. International Journal of Web Services Research, 2004, 1(4): 14-32

[12] Yi X, Kochut K. A CP-nets-based design and verification framework for Web services composition//Proceedings of the IEEE International Conference on Web Services(ICWS'04). San Diego, USA, 2004: 756-760

[13] Hinz S, Schmidt K, Stahl Ch. Transforming BPEL to Petri nets//Proceedings of the 3rd International Conference on Business Process Management(BPM'05). Nancy, France, 2005: 220-235

[14] Salaün G, Bordeaux L, Schaerf M. Describing and reasoning on Web services using process algebra//Proceedings of the IEEE International Conference on Web Services (ICWS'04). San Diego, USA, 2004: 43-50

[15] Agarwal S, Studer R. Automatic matchmaking of Web services//Proceedings of the IEEE International Conference on Web Services (ICWS'06). Salt Lake City, USA, 2006: 45-54

[16] Decker G, Puhlmann F, Weske M. Formalizing services interactions//Proceedings of the 4th International Conference on Business Process Management (BPM'06). Vienna, Austria, 2006: 414-419

[17] Hou Li-Shan, Jin Zhi, Wu Bu-Dan. Modeling and verifying Web services driven by requirements: An ontology based approach. Science in China (E), 2006, 36(10): 1189-1219 (in Chinese)
(侯丽珊, 金芝, 吴步丹. 需求驱动的 Web 服务建模及其验证:一种基于本体的方法. 中国科学 E 辑, 2006, 36(10): 1189-1219)

[18] Wang P, Jin Z, Liu L, Cai G. Building toward capability specifications of Web services based on an environment ontology. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(4): 547-561

[19] Alur R, Dill D. A theory of timed automata. Theoretical Computer Science, 1994, 126(2): 183-235

[20] Larsen K, Pettersson P, Wang Y. UPPAAL in a nutshell. International Journal on Software Tools for Technology Transfer, 1997, 1(1-2): 134-152

[21] Yovine S. KRONOS: A verification tool for real-time systems. Internal Journal on Software Tools for Technology Transfer, 1997, 1(1-2): 123-133

[22] Daws C, Yovine S. Reducing the number of clock variables of timed automata//Proceedings of the 17th IEEE Real-Time Systems Symposium (RTSS96). Washington, USA, 1996: 73-81

[23] Behrmann G, Bouyer P, Emmanuel-Fleury E. Static guard analysis in timed automata verification//Proceedings of the 9th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS03). LNCS 2619. Warsaw, Poland, 2003: 254-277

[24] Bengtsson J, Jonsson B, Lilius J. Partial order reductions for timed systems//Proceedings of the 9th International Conference on Concurrency Theory (CONCUR98). Nice, France, 1998: 485-500

[25] Minea M. Partial order reduction for modeling checking of timed automata//Proceedings of the 10th International Conference on Concurrency Theory (CONCUR99). Zindhoven, Netherlands, 1999: 431-446

LI Li-Xing, born in 1983, Ph. D. candidate. His research interests include service-oriented computing and formal methods.

JIN Zhi, born in 1962, Ph. D., professor, Ph. D. supervisor. Her main research interests include requirements engineering, software engineering and knowledge engineering.

LI Ge, born in 1977, Ph. D., associate professor. His main research interests include software engineering and software reuse.

Background

This work was supported financially by the National Basic Research Program (973 Program) of China (Grant No. 2011CB302704).

The Internet of Things (IOT) refers to extending the Internet to devices such as home appliances and sensor networks. Millions of devices will be interconnected, provide and consume data about the physical world and make it available to business processes that run in the information world. Service-Oriented Computing (SOC) is a developing computing paradigm that utilizes services as the basic constructs to support developing of rapid, low-cast and easy composition of distributed applications. Recently, many efforts have explored the integration of IOT and SOC. Devices provide their functionality as services. These devices services and traditional Web services together constitute the IOT services. They can communicate and interoperate with each other, and can be composed dynamically for specific requirements. How to model and verify the behavior of these IOT services is becoming a significant challenge.

This paper proposes an environment-based framework for IOT services. In this framework, the concept of environment entities is introduced. They are used to describe anything in the physical world, both their attributes and behavior. Then the behavior of an IOT service is specified by its interaction with the corresponding environment entities. Timed automata are used to model the behavior of the environment entities and IOT services in a formal way. The composition of IOT services, the interaction between the services and environment entities are also specified as networks of timed automata. Based on that, we present an approach for verification of IOT services by using the famous model-checking tool UPPAAL and the environment entities.