|  |  |
| --- | --- |
|  | 学校代码： 10246 |
|  | 学 号： 13212010026 |
|  |  |



|  |
| --- |
| 硕 士 学 位 论 文 |

（学术学位）

**基于本体的自适应控制在智能家居中的研究**

**Research on Adaptive Control Based on**

**Ontology in Smart-Home**

|  |  |
| --- | --- |
| 院 系： | 软件学院 |
| 专 业： | 计算机软件与理论 |
| 姓 名： | 曾君 |
| 指 导 教 师： | 李敏波 副教授 |
| 完 成 日 期： | 2016年3月31日 |

**指导小组成员名单**

李敏波 副教授

目录

[摘要 1](#_Toc447063398)

[Abstract 2](#_Toc447063399)

[第一章 绪论 1](#_Toc447063400)

[1.1 研究背景与意义 1](#_Toc447063401)

[1.2 国内外研究现状 1](#_Toc447063402)

[1.3 研究目标与内容 3](#_Toc447063403)

[1.4 章节安排 4](#_Toc447063404)

[第二章 相关技术研究综述 6](#_Toc447063405)

[2.1 关键技术概述 6](#_Toc447063406)

[2.2 本体相关技术 6](#_Toc447063407)

[2.2.1 本体定义与构成 6](#_Toc447063408)

[2.2.2 本体描述语言 7](#_Toc447063409)

[2.3 Web服务技术 8](#_Toc447063410)

[2.4 规则引擎技术 9](#_Toc447063411)

[2.5 关联规则挖掘技术 11](#_Toc447063412)

[第三章 智能家居自适应控制系统架构 13](#_Toc447063413)

[3.1 家居环境活动特征 13](#_Toc447063414)

[3.2 典型的自适应控制应用场景 15](#_Toc447063415)

[3.3 自适应控制系统架构 17](#_Toc447063416)

[3.4 设备编码与统一服务接口 20](#_Toc447063417)

[第四章 智能家居本体建模与规则推理 22](#_Toc447063418)

[4.1 智能家居本体库规划与实现 22](#_Toc447063419)

[4.1.1 本体建模方法和原则 22](#_Toc447063420)

[4.1.2 智能家居本体特征分析 24](#_Toc447063421)

[4.1.3 智能家居本体库设计 25](#_Toc447063422)

[4.1.4 智能家居本体库完善 28](#_Toc447063423)

[4.2 基于规则推理的自适应控制 30](#_Toc447063424)

[4.2.1 规则推理场景需求分析 30](#_Toc447063425)

[4.2.2 规则库定义 32](#_Toc447063426)

[4.2.3 推理流程设计与实现 36](#_Toc447063427)

[第五章 潜在规则挖掘 40](#_Toc447063428)

[5.1 关联规则概念与定义 40](#_Toc447063429)

[5.2 数据内容与预处理 41](#_Toc447063430)

[5.3 实验结果与分析 44](#_Toc447063431)

[第六章 总结与展望 47](#_Toc447063432)

[6.1 总结 47](#_Toc447063433)

[6.2 展望 48](#_Toc447063434)

[参考文献 49](#_Toc447063435)

[致谢 53](#_Toc447063436)

# 摘要

物联网的发展促进了智能家居领域的应用推广，国内外也提出各种智能家居和智能设备的方案和架构。在智能家居场景下迫切需要智能设备可以自动控制和调节，以实现设备的信息获取和智能控制。为了实现家居情景下设备和服务资源的集成管控，本文基于家居情景特点，考虑用户行为习惯，提出一套智能家居自适应控制系统架构，对智能家居场景进行本体模型构建，采用自定义规则的方式推理用户的需求服务，并对用户历史行为数据进行挖掘尝试，寻找潜在规则，进一步充实该自适应控制原型系统。

本文研究了国内外解决设备异构性的框架，通过研究和分析国内家居情景活动特征和家居场景布局方式，提出一套智能家居场景下的自适应控制系统架构。在对国内外现有智能家居本体模型调研的基础上，结合七步法的本体设计方法和原则，从用户、空间、设备、服务、网关和环境等多个维度描述智能家居场景，抽象其中的关键概念和属性信息，最终基于现实实体构建智能家居本体模型。

为了实现系统的自适应控制，本文采用基于规则的推理引擎，设计规则定义方法。在本体模型中，不仅可直接推理出用户请求服务，通过基于现实基础添加自定义规则，考虑用户行为特点，将环境信息和设备服务融合起来，还可推理出贴合用户行为习惯的目标服务。介于自定义规则的局限性，通过关联规则挖掘Apriori算法来挖掘用户历史行为数据中的情景-设备-服务关联规则，进一步充实规则库，提升系统的自适应控制能力。

**关键词：**本体，智能家居，自适应控制，Apriori，关联规则

**中图分类号：**TP311

# Abstract

The development of the Internet of things has promoted the application expanding in the field of Smart Home, and both in China and abroad there are various schemes and architectures for Smart Home and smart device. In the Smart Home scenario, it is an urgent need to control the smart devices automatically and make them adjust to get information accessing and devices controlling intelligently. In order to achieve the integration of smart devices and services based on users’ behavior and household situation characteristics, this paper has proposed a self-adaptive control system architecture for Smart Home and built an Ontology model for the scenario of Smart Home. With the using of self-defined rules and reasoning mechanism, target services can come out by reasoning. What’s more, by trying to find potential rules with data mining method in user’s history behavior data, the degree of self-adaptive for the control system prototype can be further enriched.

Based on the study of framework to resolve heterogeneity devices, by researching and analyzing about the domestic household characteristics and activities scenarios scene layout, a self-adaptive control system architecture for Smart Home scenarios is presented. After the investigate and survey about the existing Smart Home Ontology model in China and abroad, combined with the classic ontology design methods and principles, such as Seven-Step Method, the research describes Smart Home scenarios from multiple dimensions, which are User, Room, Device, Service, Gateway and Environment, then abstract key concepts and attribute information for Ontology model. Finally, a Smart Home Ontology model will be built based on real entity.

In order to achieve the self-adaptive in the control system, this paper uses rule-based inference engine and design a method to define rules. In the Ontology model, the services that user needs can be reasoning out directly. By adding custom rules based on the realistic basis with user’s behavior characteristic, and integrating the device service and environmental information, some potential services can be reasoning out to promote the self-adaptive control capabilities. In addition, taking into account the limitation of custom rules, this paper promotes an idea that to use Apriori association rule mining algorithm to mine the scene – device – service association rules based on the user’s history behavior data. The research can further enrich the rule base and enhance the self-adaptive ability.

**Keywords:** Ontology, Smart-Home, Adaptive Control, Apriori, Association Rules

**CLC Number:** TP311

# 第一章 绪论

## 1.1 研究背景与意义

随着互联网信息技术和基础网络设施的高速发展，最新一代无线通信技术、大规模云计算、物联网技术和数据挖掘等技术已经成为当下热门的技术研究领域。日常所用的无线蜂窝电话通讯协议在经历了第二代、第三代的发展，再到现如今第四代的普及，数据传输速度和大家的上网方式都发生了翻天覆地的变化，原有的大规模PC端的上网行为都已经转移到移动设备端。除了上网方式的改变，硬件的快速发展也为各行各业的变革注入新鲜的血液，进一步催化物联网行业的发展。智能家居作为物联网应用的重要领域，在近几年也掀起不小的浪潮。从单一的家居设备，如环境监测传感器、可远程控制电器，到成体系的家居解决方案，已经有不少“智能化”的家居解决样例和管理系统出现，而日常生活中，大家对智能家居相关产品的关注度更是越来越高。可见，通过网络传输技术、硬件技术和智能技术的发展来实现家居生活的自动化、智能化和高端化的终极目标，是平民百姓对家居生活的不懈追求。

但是现如今，物联网发展中的设备以实现环境中的物与物、人与人和人与物的感知、互联互通为基本功能，并提供更智能化的服务为终极目的；但是根据国家重大专项《无线泛在网络架构研究和总体设计》[1]的研究成果表明，泛在网和物联网所提供的架构发展还处于物理设备的基本通信阶段，而且仅仅能够提供设备本身所能支持的服务功能，但是对于更智能化和在更复杂环境下的服务与交互却缺乏相应的实现机制和应对方案[2]；而在智能家居设备中，这一问题同样存在。除此之外，不同厂家所提供的物理设备不论在通信协议还是在数据处理方式上都存在巨大的差异性，为此，如何有效的处理设备差异带来的异构性，充分利用设备和服务资源，从简单的人为控制层面解放出来，更智能化的使用设备和服务，实现自适应控制的目标，也是现在亟须解决的问题。

为此，如何充分利用现有的硬件设备和其提供的服务功能，并对其所提供的资源在家居场景下进行共享整合和有效管理，以及如何更合理的向用户提供其所需要的服务，并使应用更智能化，都是目前智能家居发展道路上的阻碍。要实现真正意义上的家居智能化，解决如上问题，对于智能家居的发展之路都是非常有意义的尝试。

## 1.2 国内外研究现状

自适应控制的目标是设备，而现实生活中，物联网设备因为需求和功能的不同，在传输协议和数据格式上数量种类繁多。为解决设备异构性所带来的挑战，各研究者都提出相应的解决方案。Joseph Boman，Jonathan Taylor[3]等人设计了一种物联网数据解释器，用以管理物联网异构设备，基于解释器结合Firebase和GSN构建一种通用的物联网中间件，以处理异构设备带来的通信阻碍。Perera C，Zaslavsky A[4]等人提出CASSARAM物联网中间件，基于用户优先级来对情景感知的设备进行搜索和选择，并提供一种语义解析方法，可以让用户借助自然语言来操控底层物理设备。

为更好的实现物理设备、服务与用户需求的整合和信息交流，则需要充分利用所处场景的完整信息。情景建模问题由此而来，而借助于语义和本体技术，则可以更好的管理设备与服务。通过本体建模，可以构建真实物理对象的概念抽象，并提供对情景信息的自由访问。针对本体建模方法这一问题，SUN J.Z和 SAUVOLA J[5]使用一个四元模式(Entity Name, Feature, Value, Time)来描述和现实物体对应的数据对象，对于每个现实物体用唯一标识来区分，并使用特征、值和时间来表述模型中的与情景对应的现实物体内容信息；STRANG和LINNHOFF-POPIEN[6]提出ASC (Aspect-Scale-Context information model)本体模型,用本体通过一种一致的方式描述核心概念，并用Context Ontology Language (CoOL)模型语言在Web service上构建本体推力器(ontology reasoners)，使得实体信息和情景信息可以最大程度的共享与重用。针对如何对物联网中的服务进行融合，降低各个模块之间交互的复杂度这一问题，并进一步优化系统的自适应程度，则需要充分参照信息融合技术的机制，更智能的实现自适应控制过程。《Multi- Sensor Data Fusion》[7]和《Handbook of Multi-Sensor Data Fusion》[8]根据传感器数据传输特点，设计数据处理模型来应对传感器的数据异源性，并提出融合方法来处理数据的海量性，最后叙述如何在一个现实系统中运用数据融合技术[9]，这些都为自适应控制机制的研究提供基础。

除了物理空间情景建模方法，在针对智能空间模型上，国内外也进行过研究。麻省理工学院人工智能实验室Michael H. Coen, Brenton Phillips[10]等人提出的Metaglue System，通过模拟现实家居场景信息，将硬件设备和软件应用通过Agent技术整合在一起，动态响应用户需求，学习用户需求，以便更好地为用户服务。微软研究院的EasyLiving[11]项目，通过尝试解决中间件、情景建模、感知信息处理、服务描述和处理等问题，来构建高度感知的生活空间。都灵理工大学开发的DogOnt[12]模型，是用于智能家居的语义模型，该模型在家居环境下支持对家居设备的语义控制；在这个模型中，基于本体构建五大组件，分别是Building Thing，Building Environment，State，Functionality，Domotic Network Component,除此之外，该实验室还开发出能够跨不同协议对

家用设备进行基本语义控制的网关系统。新加坡信息研究所研制的语义空间[13]，充分利用语义技术，实现情景信息的表示、推理及查询等功能。清华大学的远程智能教室iSmart[14]项目，则利用教室场景为背景，其中的设备和服务为载体，解决教室中服务互动的问题，在试验中获得良好效果。这些成熟的研究成果都为本文系统的研究提供良好的思路和借鉴。

除本体建模以外，为做到在满足用户需求前提下的设备自适应控制，则需要通过考虑用户的行为习惯和偏好来进一步促进系统的自适应程度。针对这一问题，Kishore Ramakrishnan A和Preuveneers D[15]等人对常用的物联网应用情景进行抽象和建模，并通过机器学习的方法来挖掘情景信息之间的关系。Gediminas Adomavicius 和 Alexander Tuzhilin[16]根据上下文内容推荐，提出算法范例。除具体的建模和处理方法外，数据挖掘中的关联规则挖掘也可以用于寻找用户行为习惯偏好[17]。Mobasher，Bamshad[18]基于Apriori算法提出改进算法，基于用户历史上网数据，通过关联规则挖掘方法，寻找用户上网模式。对于现实生活数据，Zhenyu Chen[19]提出BP-Mine框架，处理用户的历史位置信息，基于生活模式来提供对生活活动的推荐。Berkane[20]设计了自适应性系统原型，通过采用模式匹配的方式来映射环境物体和服务。

对于智能家居场景下的情景建模，很多研究者都有涉及，但是从共享和重用的角度考虑，还无法满足本文的研究需求；而且在本体建模基础上的自适应控制研究，更需要一个切实可用的本体模型。本文在上述研究内容基础上，通过调研国内的家居情景信息，对家居情景特征进行分类和提取，为家居场景构建更详细的情景本体模型。同时，为了让家居控制更智能化，充分利用规则引擎处理的优势，定义并抽象规则，让情景本体中的设备与服务可以被更灵活的调用；并探索基于历史数据如何挖掘更合理的潜在用户需求规则。

## 1.3 研究目标与内容

智能家居场景应用的兴起，必然会带来多方面的挑战，其中如何从单纯的控制化转变为更能贴合用户生活需求的自适应控制系统，是不可逾越的问题之一。本文在对智能家居场景下的应用系统和相关技术进行充分调研的基础上，结合现有智能家居应用市场上的应用需求和场景应用特点，提出一套智能家居自适应控制系统架构，并对智能家居场景进行模型构建，为后续的基于规则的服务推理提供应用基础；充分利用成熟的数据挖掘方法，对用户历史行为数据进行挖掘尝试，寻找潜在规则，设计自适应控制原型系统。

为解决以上问题，实现本文相应研究目标，具体研究目标和思路如下：

一、**提出智能家居自适应控制系统架构**。在分析现有常见家居活动和家居场景布置的前提下，提出常见的家居环境下自适应控制场景，推出更贴近需求的自适应控制系统架构，并详细阐述各个层级的划分方式和功能模式，阐述本文的研究侧重点，对所涉及的异构设备的编码方式和统一服务调用接口做明确说明。

二、**设计并实现基于本体的智能家居模型**。利用经典本体建模方法，充分调研国内外现有智能家居本体模型，设计一套更全面可用的智能家居本体库，并根据实验实体添加完善本体库信息，为后续的研究工作提供研究基础。

三、**研究并****设计自适应控制规则的定义方法，结合本体模型实现规则推理逻辑流程**。基于谓词逻辑提出一种规则定义方法，在智能家居本体库和现实规则的基础上，考虑现有实体之间的关联关系，提出一种从用户需求到目标服务的逻辑推理流程，为智能家居环境下的自适应控制提供一种可行方案。

**四、利用关联规则挖掘思路，实现Apriori算法，挖掘关联规则**。通过对用户历史服务调用数据的充分处理，利用关联规则挖掘相关算法-Apriori，来挖掘用户日常行为中服务调用之间的关联关系，以及服务与调用时间的关联关系，从而提炼规则，为进一步提升系统的自适应程度提供一种思路。

## 1.4 章节安排

本文通过对家居场景的调研，充分考虑现实世界的设备异构性，提出一套智能家居场景下的自适应控制框架，并针对应用层的层级功能，设计智能家居场景下的领域本体，研究基于本体、规则和服务的推理逻辑，以实现自适应控制的目标；最终基于用户的历史行为数据，挖掘潜在规则，为该自适应系统更智能化的自适应提供思路。全文的结构安排如下：

第一章，绪论。说明研究背景与意义，阐述研究现状，提出研究内容和目标，介绍本文章节分布。

第二章，相关技术研究综述。对本文用到的本体知识、本体描述方法、和建模技术进行调研，并就自适应控制研究所涉及的规则引擎方法、Web服务和关联挖掘技术进行全面了解。

第三章，智能家居自适应控制系统架构。研究现有的自适应控制场景，提炼典型的家居布置和活动特征，结合物理设备和服务的描述，提出本文的自适应控制系统架构，详细阐述架构的各层级功能。

第四章．智能家居本体建模与规则推理。详细介绍智能家居场景的本体特征，基于本体建模方法，结合现有的各类智能家居本体库，提出智能家居本体库模型，并添加相应的实验实例信息。基于智能家居本体库的设计，提出规则定义方法，设计一套基于规则的服务推理模型。

第五章．潜在规则挖掘。结合用户的历史采集数据，采用成熟的关联规则挖掘方法，挖掘潜在的用户生活-服务规则。

第六章，总结与展望。对全文进行总结，并对工作中的不足作未来展望。

# 第二章 相关技术研究综述

## 2.1 关键技术概述

本文主要研究智能家居情景下的本体模型构建，并在该模型下解决基于规则引擎的服务推理问题，以实现系统的自适应控制，通过数据挖掘方法，挖掘潜在的用户行为习惯规则，提升系统的自适应程度。在研究中涉及到的技术主要分为四个部分，分别是本体建模技术、引擎推理技术、Web服务技术和关联规则挖掘相关技术。

智能家居本体建模的目的是构建现实世界的抽象概念，能够用计算机可以理解的语言来解释物理世界的物体和运行过程。而本体，则正好满足这一点需求，本体用人类的思维方式来表述世界，并可以借助于语义技术，来更好的解释世界。本体的发展已经比较成熟，相关技术也较多，为了更好的为本文研究工作服务，则需要选择更契合的本体建模方法。模型构建成功后，需要对设备以及服务资源进行描述，使得本体和设备都能够准确识别对应实体。Web服务相关技术作为成熟可用的技术，种类繁多，需要对这些技术作充分调研。

自适应系统的控制，本文通过规则处理的方式来模拟实现。在处理规则的过程中，规则引擎是不可或缺的技术，其中引擎又分为前向推理引擎，后向推理引擎，每种引擎处理都有其自身的优缺点，而相关工具也种类繁多。规则的再挖掘是本文的另一个研究重点，它涉及数据处理和数据挖掘等方面内容。

## 2.2 本体相关技术

### 2.2.1 本体定义与构成

本体最先来自于哲学，是对现实世界中客观存在系统的解释、说明和抽象，以体现其客观现实的本质。随着研究的发展，本体也有了新的含义，Guarino，Giaretta[21]等人经过深入分析，给出本体的四层含义：概念模型、明确性、形式化、共享，即用计算机可以理解的语言，用清晰明确的概念和约束明确定义现实世界中的实体和相关关系，最终形成可抽象客观世界的模型，同时该模型和所定义的客观世界中的具体状态没有相关性，最终抽象模型体现的是为大家所接受的知识，也是相应领域中公认的概念集。本体的应用，可以为系统带来多种好处，如本体的形式化描述可以提升软件系统的重用和共享程度，构建过程中的一致性检查也可以提升系统的可靠性，本体作为形式化规约，更可以提升系统的规范性。

根据Gómez-Pérez提出的本体论，本体由五个基本要素组成，分别是概念、关系、函数、公理和实例[22]。概念可以描述任何事物，是现实对象的抽象集合；关系则是概念与概念之间关系的桥梁，形式化定义为；函数则是一种较为特殊的关系，该关系的最后一个元素是由前面的若干元素所决定，形式化定义为；公理代表断言；实例则是本体中的现实元素，是现实反映在抽象模型中的对象。要素中的关系，根据关系主体的不同，分为部分关系（part-of）、所属关系（kind-of）、实例关系（instance-of）和属性关系（attribute-of）等。

### 2.2.2 本体描述语言

本体作为一个领域中概念的形式化规约，需要用良好的语言来加以描述。基于本体的含义，则要求本体描述语言拥有良好的语法结构和形式语义，并且对概念描述具备足够的表达能力，最重要的是能够支持高效的推理。迄今为止，已经出现众多本体描述语言[23]，如XML、RDF、RDFS、DAML、OIL和OWL,其中使用最为广泛的是RDF(Resource Description Framework)和OWL(Web Ontology Language)。

RDF[24]利用Web体系结构中的URI来标识资源（Resource），并用简单的属性、属性值来声明。一个完整的RDF语句（statement）表述一个资源具有的属性—属性值信息，以三元组的形式来展现，如。RDF所提供的描述元数据的方法，只是表述资源本身，缺乏资源之间语义关系的表达，为此有了RDF的扩展——RDFS[25]。RDFS在语义上添加很多机制来加强对资源的描述能力，但是对于达到完全的推理，还是缺乏对应的语义描述能力。针对RDF和RDFS在语义方面表示不足的问题，W3C发布了Web标准本体表示语言OWL。OWL[26]相比其他的描述语言有更多的机制来表达语言，不仅是在信息表示格式上，在知识表示上也更加语义化。OWL中还添加更加复杂的方法描述类和属性，比如不相交性、传递性、对称性等，用以表达更强的语义信息，更重要的一点，OWL支持逻辑推理。在表达能力强弱上，分别有三种子语言：OWL Lite、OWL DL、OWL Full[27]。OWL DL语言相比另外两种子语言在OWL约束上更加的全面具体，约束能力更强，而且在推理能力上也更加丰富。

由于本文要对智能家居场景下的基本概念，以及概念之间的直接关系和间接关系进行描述，概念之间存在的更多关系需要依靠规则推理来获得，所以还是需要更强表示和推理能力的OWL DL描述语言。斯坦福大学医学情报研究组研发的本体构建编辑工具——Protégé，作为支持本体构建和查询推理的软件，能够极大的方便本体模型的构建过程，而且可以自动化的检查本体的一致性和意外关系，以确保本体构建的完备性。

## 2.3 Web服务技术

自适应控制架构中需要将设备与服务进行整合与管理，而且设备接收指令进行操作，这些指令的下发与传送，则借助于服务的方式。通过发布供调用的统一格式的API，使得服务具有完好的封装性、规范性和高度可集成能力。通过对网络资源进行统一标记（URI），并利用相应的传输协议（HTTP）来传输这些资源，在对这些资源进行描述的时候，可以使用多种方式，比如HTML、XML、JSON等。Web服务作为可分布式部署和模块化处理的应用形式，能够在不同的开发平台和应用平台自由工作，为此，需要提供一种能够实现互操作的标准类型系统，为此，服务描述语言WSDL、服务发布语言UDDI和简单对象访问协议SOAP[28]则是为Web服务跨平台的互操作服务。WSDL用基于XML的、机器能够理解的方式来描述Web服务的函数类型、参数以及返回值；UDDI则提供Web服务的注册规范。

SOAP作为远程调用过程的应用协议，已经在Web历史上发展了很长时间。该协议所定义的框架具有良好的封装性，遵循一致的编码规则，且可以绑定多种数据交换协议。该协议简单、自由、与编码语言无关，也和运行平台无关，能够实现设备与服务的“无缝衔接”。相比之下，REST[29]作为另一种Web服务协议，也有以上的诸多优点。REST所制订的指导原则使服务支持“系统组件间交互的可扩展性、接口的一致性、组件的独立部署、降低交互延时的中间组件、安全性的加强、对遗留系统的封装”，而且针对设备控制和服务提供方法，Guinard[30]等人提出的Web of Things的技术架构，将物联网中的智能设备抽象为Web资源，使得资源可以被标识、建模和搜索，同时Web Service降低物联网服务的开发门槛以及物联网终端之间的异构性。

但是从安全性上考虑，SOAP在封装消息头时会对消息内容进行加密，以保证消息不被截取和修改，而REST架构下的应用就缺乏这样的处理。所以，本文采用SOAP作为设备服务的访问协议。

## 2.4 规则引擎技术

推理引擎（Inference Engine）是专家系统的核心模块，常见的有基于案例、基于神经网络和基于规则的推理引擎，其中基于规则的推理引擎比较容易理解和管理，被称为规则引擎。规则引擎[31]在结构上主要有模式匹配器（Pattern Matcher）和议程（Agenda）两个组件；规则引擎在执行前，需要接受两种输入参数，分别是用以表达知识的规则（rule）和待分析的现实事实（fact），这两项参数信息存储在内存中，分别命名为规则库（Production Memory）和工作空间（Working Memory），结构如图2.1所示。

图2.1 规则引擎结构图

模式匹配器基于事实找到与之匹配的规则，并通过议程管理匹配规则的执行顺序，最终，由执行引擎（Execution Engine）来执行具体的操作。模式匹配器是规则引擎的核心部分，该部分包含两种推理方式，分别是正向推理（Forward Chaining）和反向推理（Backward Chaining）。正向推理由事实驱动，往往以一个初始事实作为起点，不断地应用规则库中的规则得出结论；反向推理则是由目标驱动，通过提出假设寻找支持假设的依据和规则来达到。目前常见的Java规则引擎应用主要有JRules、Jess、Blaze Advisor、Drools等，还有一些支持规则引擎的工具，如Pellet、Jena等。

Jena[32]是HP实验室研究的专门用来构建语义网的应用工具，支持前向推理、后向推理和混合模式的推理。Jena2[33]完全以Java开发，该工具定义的很多接口和抽象类，可以方便的读取和分析标准的OWL文件，提取、添加和改变其中的属性和值信息，同时，基于以上实现，其内部的推理引擎也可以帮助实例和结构的推理。Jena由六个组件构成，分别是ARP、RDF API、持久性存储、推理子系统、Ontology API、RDQL。Jena通过ARP解析RDF/XML数据文件，利用RDF API创建、解析、处理RDF模型，并可以创建和存储持久性模型。推理子系统[34]的工作原理如图2.2所示，推理注册机（Reasoner Registry）根据基本的三元组描述断言（base assertions）和定义的本体断言（ontology definitions）创建推理机（Reasoner），之后由该推理机通过API生成对象模型（Inf Graph）；最终借助Model类对对象模型进行推理模型的构建，以支持访问和推理[35]。

图2.2 Jena规则推理机的模型[35]

在这些功能的基础上，Jena不仅包含基本的规则推理功能，还支持使用者根据自身需求，遵循一定语法，自己定义和添加推理规则。为此，Jena定义了基于文本的规则语法结构，如图2.3所示。

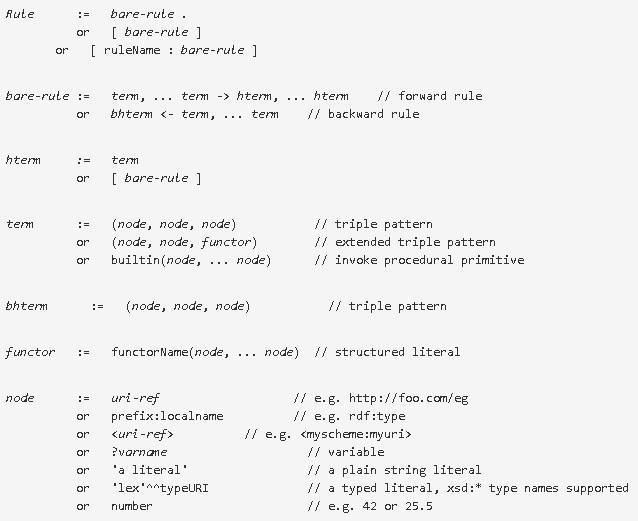
基于Jena的规则语法，自定义规则和规则集，通过使用Jena的规则推理功能，则可以推出三元关系中的任何潜在关系。本文利用该工具，在本体库文件之上推理获取需求对应的服务目标。

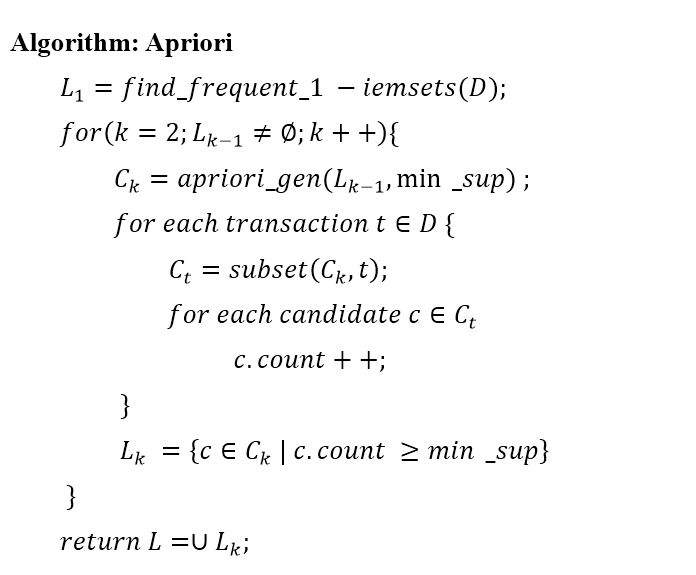
图2.3 Jena文本规则语法结构

## 2.5 关联规则挖掘技术

数据挖掘[36]是从大量的数据中发现隐藏的属性间特殊关联性的过程。从应用分类上分为关联分析、序列模式、分类、聚集和异常检测等；从结果上则分为知识发现和数据模型挖掘。本文所采用的关联规则挖掘，则是数据挖掘中关于知识发现的分支。知识发现是根据特定的模式，来发现其中存在的关联、分类和规则方面的信息内容。

对于用来进行挖掘的数据，存在一定的噪声，不完整或者信息过于冗余等问题，需要进行一定的数据清理。预处理后的数据，则需要根据情景特点，采用相应挖掘算法对数据进行处理，来寻找其中的关联规则。

关联规则挖掘的主体思想是在数据中查找存在于数据项目集合或对象集合之间的频繁模式、关联、相关性或因果结构。基于该思想，Agrawal[37]于1993年提出关联规则挖掘的基本算法——Apriori。该算法的基本思想是找到频繁项集，然后根据找到的频繁项集产生期望的关联规则。核心执行步骤如下所示：

Apriori算法可以解决关联规则挖掘的基本问题，但是它还是存在一些效率上的缺陷，比如可能产生大量的候选集，无法对稀有信息进行分析等。为此，更多改进的算法被后续研究者所发明，比如Apriori Hybrid算法，FP-growth[38]算法，在效率和处理结果上，都有明显提升。即便如此，本文因为不对算法本身做改进和实验，仅希望通过使用关联规则挖掘方法来验证潜在规则挖掘的可行性，所以还是采用最经典的Apriori算法。

# 第三章 智能家居自适应控制系统架构

在充分调研智能家居自适应控制的国内外研究现状基础上，本章研究目前常见家居环境布局特征和经典家居生活情景，并通过介绍典型的自适应控制应用场景，分析实际需求，提出一套智能家居场景下的自适应控制系统。在该自适应控制系统的框架下，充分考虑家居环境特征和需求，明确设备中间层所提供的支撑和服务，为构建智能家居本体库和基于规则的自适应控制提供研究和内容基础。

## 3.1 家居环境活动特征

从漫长的历史发展看，家庭一直以来都是社会发展的核心单位，而家庭成员所生活的家居环境也经历了漫长的演变。单从住房面积上来说，自改革开放以来，已经经历翻天覆地的变化。据中国住房和城乡建设部部长姜伟新在2010年举办的全国住房城乡建设工作会议上公开表示[39]，中国城镇居民人均住房面积从1978年的6.7平方米，增长为2008年的28.3平方米，实现近422%的增长。不论从住房质量上，还是在住房环境上，都有很大的改善。

除了住房面积的变化，随着生活水平的提升和经济生活的改变，在住房功能方面，变化也随之产生。在改革初期，住房在城镇居民生活中主要起到卧室的作用。因为住房资源和土地资源的限制，住房配套的卫生间和厨房主要采用多户共享的方式。随着改革开放的进行和经济的发展，城市发展进一步加快，资源的挖掘使得住房在功能上不仅仅局限于休息，随之而来的是更适合居民生活需求的功能，比如休闲、娱乐、盥洗、工作、烹饪、收纳等。因此，更响应时代需求的住房，带着更丰富的家居功能，成为人们生活的主流。中国住房和城乡建设部于2012年实施的国家强制性标准《住宅设计规范》（GB50096-2011）[40]，其中强调“住宅应按套型设计，每套住宅应设卧室、起居室（厅）、厨房和卫生间等基本空间”、“厨房应设置洗涤池、案台、炉灶及排油烟机、热水器等设施或为其预留位置”等。国内各大房地产开发商也严格按照国家强制性标准来建设住宅，虽然在实施细节上有些许变动，但是总体上还是严格依照《住宅设计规范》，包括卧室、起居室、厨房、卫生间等基本空间，或者为更进一步满足用户需求，设计书房、餐厅、收纳室、车库等空间。

在包含基本空间的家居环境中，卧室主要供睡眠用，起居室主要进行娱乐、休闲等活动，厨房主要进行炊事活动，卫生间主要供住户盥洗用，书房主要提供住户工作的场所，餐厅为就餐处，收纳室是对生活物资进行基本整理处；其中有些空间因为功能特性不一样，所主要进行活动的行为人也有一些差异，导致空间特性也不同，比如起居室、餐厅和卫生间为公共活动空间，而卧室、书房为私密活动空间，余下的厨房、收纳室以及车库等，为家庭服务空间。除了这些明显的划分以外，可以发现，所有空间的主要活动，都是由一个一个的原子行为所组成，比如在厨房的烹饪，可能需要进行开火/水、用冰箱/微波炉/热水器、预防煤气泄露等行为；进入书房工作学习，可能需要进行开门、开灯、关窗帘、开空调等行为。这些行为的控制对象，映射到家居空间中，分为家电控制、灯光控制、安全防范等。详细划分如表3.1所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 空间名称 | 空间类别 | 常规活动 | 控制对象 | 环境因素 |
| 起居室 | 公共空间 | 娱乐，社交，休息 | 空调，电视，音响，窗帘，灯光等 | 光照，  温度，  湿度，  声音，  烟雾，  红外线，磁震动，CO等 |
| 卫生间 | 洗澡，洗衣 | 热水器，洗衣机，灯光等 |
| 餐厅 | 就餐，社交 | 灯光 |
| 卧室 | 私密空间 | 睡觉，休息 | 空调，窗帘，灯光等 |
| 书房 | 工作，学习 | 灯光，空调，打印机等 |
| 车库 | 服务空间 | 车出/入库 | 报警装置，自动门等 |
| 厨房 | 烹饪，洗涤 | 热水器，微波炉，冰箱等 |

表3.1 典型空间类型及其相关活动和活动对象

随着传感技术和智能硬件的发展，表3.1所示的各个空间活动下被划分的原子粒度的家居行为，在市场中多半都存在与行为相对应的传感器和智能设备，比如，预防煤气泄露可以使用感知CO浓度的传感器，开灯/窗帘可以采用智能开关，对包括空调、加湿器、热水器等常用家电的控制，可以采用相对应的智能家电。所以，通过家居环境中使用的传感器和智能硬件的类型分类，可以将家居环境中的活动分为感知家居环境参数的感知设备和对照明、电器等进行作用的控制设备。一个典型的家居环境，如果配上相应常见的感知设备和控制设备，以及用于感知设备和控制设备进行信息传输的家庭网关，其平面示意图如图3.1所示。

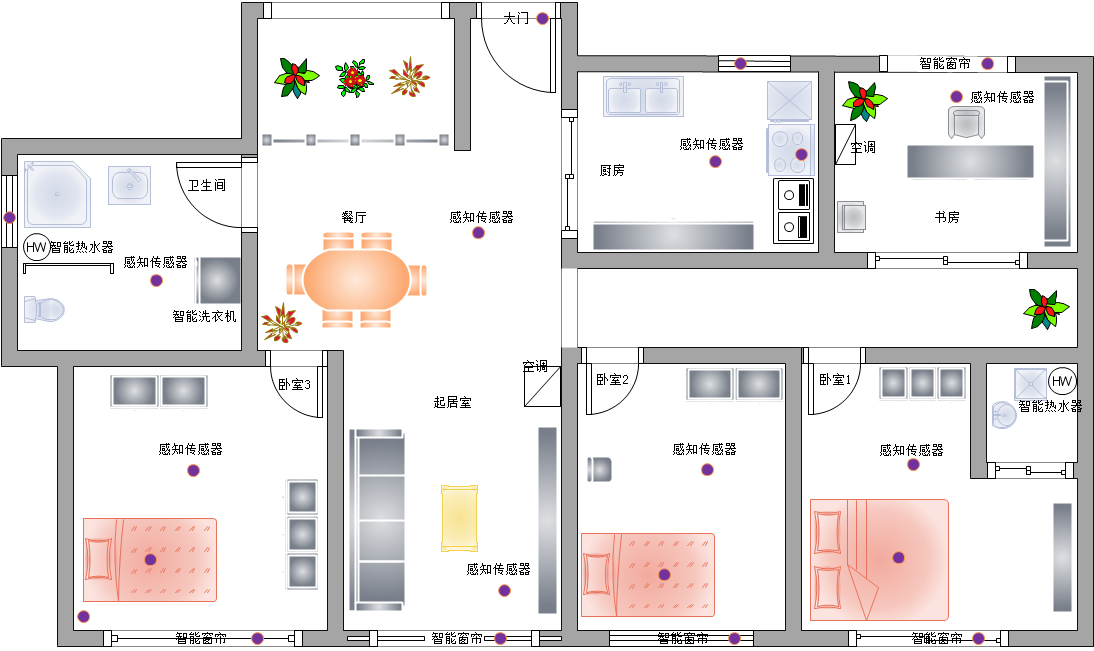
该智能家居环境由3个卧室，1个起居室，1个餐厅，2个卫生间，1个厨房，1个书房组成。在所有的门窗处均安装磁震动报警器和智能窗帘，所有空间中均采用智能开关，不仅可以手动开关，还可以支持远程开关；所有的独立空间中都有用来监测空间环境的传感器，包括温度传感器、光照传感器、湿度传感器、烟雾传感器、CO传感器、声音传感器、红外线传感器等；在卧室的床上，安装压力传感器，用以感知用户是否处于平静的睡眠状态；除此以外，还有一些智能家电，比如可以远程控制和调节温度的智能热水器和空调，通过压力感应来控制洗衣时间的智能洗衣机等。所有的这些感知设备和控制设备，都有其相应的归属空间，负责一定的功能，或者感知环境参数信息，或者改变空间环境状态。

图3.1 智能家居场景布局平面示意图

通过以上家居环境和活动的分析，发现居住人在生活环境中的活动具有目标相似性，可划分性。对于空间中的各类活动，同一用户在执行过程中，各类活动相似度比较高，这可能是基于用户个人的性格和习惯，比如睡前的习惯是拉上窗帘，调节空调，关灯；比如进入书房工作习惯是开灯，调空调等。同时，对于任何一个目标活动，都可以通过其接触的家居对象，将活动划分为粒度更小的行为，比如睡前拉上窗帘，调节空调，关灯，分别作用于窗帘，空调和照明设备。

## 3.2 典型的自适应控制应用场景

在前文所描述的智能家居环境下，空间活动中的多个方面都能够因为智能设备的使用而带来意想不到的改变，让生活更加的便捷、舒适、惬意。在本节中，将就生活中的典型场景，来描述智能家居环境下，如何使用智能设备和服务，来实现自适应控制。

在表3.1家居环境活动特征中，提出智能家居环境中通常包含的环境感知设备和智能控制设备，为了更明确地说明以下场景，在此详细说明模拟的智能家居环境中包含的感知设备和控制设备。在各个空间中，包括三个卧室，两个卫生间，一个厨房，一个书房，一个餐厅和一个起居室，都包含感知设备，比如温度传感器、红外线传感器、烟雾传感器、湿度传感器、声音传感器和光照传感器，除此之外，厨房中还包含C0传感器和天然气阻断阀[41]，各个卧室的床上还有压力传感器。在各空间的窗户上，都有磁震动传感器，用以感知窗户的开启和关闭状态。同时，各个空间中都有一个对应的智能开关，用以控制空间中的灯光。卫生间中的智能热水器、洗衣机，起居室和书房的空调，以及各个卧室的智能窗帘，都为可控制的智能设备，支持远程遥控，模式设置和参数调节。基于以上列举的智能家居环境中的感知和控制设备，列举以下三种典型自适应控制场景。

用户主动请求：正值隆冬季节，用户A正在下班途中，预计30分钟后到家，希望家里有一个舒适的环境，于是通过与智能家居终端应用交互，传达“10分钟后把起居室空调调到26℃的制热模式，30分钟后将厨房和起居室的大灯打开”的指令。30分钟后，用户A到家，厨房和起居室的灯已经打开，空调也处于工作状态。

基于自定义规则自适应触发服务：用户A离家后，家中处于无人状态，设置防御模式，十分钟后，接到来自智能家居终端应用的报警，提示厨房的CO浓度超过65 ppm（parts per million，百万分比浓度），并请求是否开启天然气阻断阀服务，以防止天然气泄露等突发事件，此时用户为了安全起见，选择开启天然气阻断阀服务。半个小时后，又接到来自智能家居终端应用的红外线报警，提示家中有不安全的外来人入侵，此时，用户拨通小区物业的电话，请人过来查看一下，是否有异常情况。

基于用户历史行为的自适应控制：用户A每天晚上6点左右来到书房，开灯，开空调，并调节到24℃，拉上窗帘，开始学习，在晚上10点左右关灯，关空调，离开书房。如此持续若干周，某一天，下午5：50，智能家居终端应用提醒用户A，是否需要打开书房的灯、打开空调并调节到24℃，拉上窗帘，用户A确认后，相应的服务被调用，快到六点时，用户A来到书房，所有应该准备的环境都已经准备妥当，用户A直接开始工作学习。

以上列举的典型应用场景，其运行逻辑是通过获取用户请求指令，或者监测实时的环境参数，根据解析请求指令和监测的环境参数，融合智能家居自适应系统中的规则库，来最大程度匹配智能家居系统中的服务，并对服务进行触发调用。在自适应系统中的规则，包含预先自定义的原生规则，用户熟知的常见生活规则，还包含用户不知，却在其生活习惯中潜在的衍生规则。

## 3.3 自适应控制系统架构

基于家居环境中活动所具有的特征，以及在智能家居环境中典型的应用场景，本文设计基于本体的自适应控制系统架构，如图3.2所示，系统架构主要分为五个层面，从下至上分别为物理设备层、设备中间件层、服务层、应用层和交互层。

物理设备层：该层主要是现实世界中的各种物理设备，主要包括控制设备、感知设备和提供各个设备进行通信的网关。设备主要通过常见的无线网络传输协议来与网关进行通信，进而传输数据，比如蓝牙，Wi-Fi，ZigBee等。感知设备根据设置的间隔时间发送传感数据给网关，之后再向目标接收点发送数据；控制设备则等待上层下发的控制指令，改变控制设备运行状态。该层是所有上层应用的基础，也是用户最终实际控制的目标。

设备中间件层：控制设备和感知设备可能来自不同的厂家，采用不同的传输协议和数据包格式，设备的异构性，可能造成设备与服务的不统一和紊乱。为此，为了更好的统一控制底层的物理设备，并让系统的耦合度最小，在物理设备层之上添加设备中间件，用以作为现实与虚拟的中间桥梁，支撑底层设备和上层应用。该层主要负责为异构设备建立统一资源描述模型，用来抽象和映射显示设备；负责异构传输协议的解析抽象和封装处理，以使得新接入的设备能够快速匹配适用的传输协议；负责服务封装和响应，通过一套标准的封装格式，将底层设备暴露的所有服务封装成统一的格式，并通过WSDL方式发布。

服务层：该层主要负责封装设备中间件层发布的原始服务。中间件所暴露的服务相对统一和单一，但是设备具有多样性，所具备的服务模式和调用方式也存在异构性，所以，为了便于应用层的本体构建和统一管理，在服务层，进一步对中间件层监测的设备信息和设备所包含的服务进行解析和封装。

应用层：现实生活用户与现实设备，通过应用层得以进行更深一层的交互和应用。该层通过解析智能家居场景中的情景信息，根据实际的应用场景，构建能够与现实情景匹配的本体库，并在本体库中添加现实生活中实际存在的实例信息。同时，根据生活，构建原生规则库。除此之外，推理引擎根据交互层的用户请求，充分利用智能家居本体库的情景信息，推理出最符合请求的服务，并下发请求调用服务。日常生活中的请求-服务调用信息，都被记录在历史行为库中，当服务调用数据积累到一定的量，则通过关联规则的数据挖掘方法，从历史数据中挖掘出符合要求的潜在规则，进一步充实规则库。

图3.2 智能家居自适应控制系统层级结构

交互层：用户通过智能家居终端应用与整个自适应控制系统进行交互，对于语音或者文本输入的指令，需要进行预处理，首先验证交互指令是否有效，然后将指令根据词性进行切割，并转换成系统可以理解的资源信息，通常包含“执行时间”、“空间”、“环境”、“设备”等多个维度信息。除此以外，

该层还需要作为系统的窗口，与用户进行“对话”，告知用户当前智能家居中的服务调用请求处理结果和预警信息等。

以上各层的逻辑关系如下：物理设备层映射现实世界的物理设备，设备包括设备本身、通信和服务，并通过设备标识来唯一的标识每个设备，该层的设备产生原始的环境信息和原子服务，并响应上层的指令调用；设备中间层将底层的设备进行虚拟抽象，映射成具有统一模式的设备资源，包括设备信息、设备状态、设备服务等，该层作为虚拟与现实设备的桥梁，接收来自上层的标准化操作指令和数据请求指令，分析处理后向物理设备层下发相关原始设备指令；服务层是对设备中间件层服务的封装，以更加标准化和模式化的方式被本体所理解、管理和应用；应用层是构建用户-设备关联关系的核心，该层通过构建情景本体模型，将用户与用户家居环境映射成更容易理解和管理的计算机资源，并基于用户请求和原生规则的推理获取符合需求的目标服务，最后，通过历史服务调用数据，挖掘潜在规则，进一步丰富规则库，提升系统的自适应控制能力；交互层是系统与用户交互的窗口，并负责用户请求指令的信息预处理，为下层的应用层提供符合标准规范的输入。

图3.3 智能家居自适应控制系统应用层架构图

在应用层中，通过全面分析现实家居场景信息，分析其中的关键属性，构建多维模型，形成智能家居本体库；通过分析常见家居行为，提炼原生规则，构建规则库；在用户请求且服务响应调用后，所有的历史数据以一定的格式记录在历史行为库中；推理引擎获取交互层的请求信息，通过结合智能家居本体库中的多维信息，并基于规则库，推理出符合请求的服务集；规则挖掘则不定时的对历史行为库信息，通过关联规则挖掘方法挖掘出潜在规则，充实规则库，应用层具体的交互逻辑如图3.3所示。

通过上述对智能家居自适应控制系统的层级描述，不难发现要实现一个完整可实施的智能家居自适应控制系统，需要全面、正确的运用诸多相关技术和方法。本论文以这一智能家居自适应控制系统框架为背景，在该层级架构下对应用层的智能家居本体模型构建、推理引擎扩展应用和基于用户历史行为数据的规则挖掘三个方面的内容进行研究。具体的内容将在第四章和第五章给出。

## 3.4 设备编码与统一服务接口

为响应用户请求和管理异构设备，以达到最终精确匹配现实设备服务的目标，在该自适应控制系统中，将对物理设备层的设备提出一套编码标识方法[42]，以唯一标识智能家居自适应控制系统中的每一个设备。

设备通过网关与外部网络进行通信，网关作为智能家居环境中的物理设备，也具备相应特点标识。因为设备之间的异构性，同时设备与网关的绑定传输[43]，所以通过网关的唯一标识，从而使原本标识不唯一的设备也具有唯一标识，进而可以匹配其下的设备和服务。设备和网关来自不同的厂家，通过充分利用厂家标识信息也可以标识该厂家的网关和设备。利用唯一的厂家标识、唯一的网关标识和设备标识，则可以标识智能家居环境下的所有设备[44]。

定义厂家编码集合为“company”，网关编码集合为“gateway”，设备编码集合为“sensor”，那么对于智能家居场景中网关设备的标识定义如下：

厂家编码具有全球唯一性，虽然多个厂家在定义网关编码时可能出现一样的可能，但是加上唯一性的厂家编码标识后，所定义的网关也将被唯一标识。同理，多个厂家定义的设备编码有很大概率重合，但是基于厂家编码和网关编码的设备标识，则也具备了唯一性。定义智能家居场景下感知设备和控制设备的标识定义如下：

例如南京物联的厂家编码为“09088266”，其下某个网关编码为“C85B3A134EE6”,某一个温度传感器的编码为“001”，根据以上标识定义方法，网关的唯一标识“gatewayId”为“09088266+C85B3A134EE6”，温度传感器的唯一标识“sensorId”为“09088266+C85B3A134EE6+001”。

除了对设备的标识外，对于设备的服务也进行标准化统一。设备的服务分为感知环境参数的服务和控制设备的服务，前者主要是获取当前环境的参数信息，而后者主要是设置控制设备的不同模式状态。感知服务通过获取感知设备唯一标识来查找设备，获取感知参数；也可以通过地理位置和参数类型搜索匹配设备，获取感知参数；最终的感知参数通过XML的形式封装，包括time，propertyName，unit，value，sensorAddr，onlineStatus等属性信息。控制服务通过获取控制设备唯一标识来查找设备，并可以选择开启、关闭、设置模式等服务，最终的返回消息通过XML的形式封装，包括time，sensorAddr，onlineStatus，operatorResult等属性信息。设备中间件层提供的统一接口如表3.1所示。

表3.1 设备服务统一接口

|  |  |
| --- | --- |
| **类型** | **服务接口** |
| 感知 | getCurrentDeviceData(String sensorAddr，String protocol) |
| 控制 | startDevice(String sensorAddr, String protocol) |
| shutDownDevice(String sensorAddr, String protocol) |
| setDevice(String operation, String parameter, String sensorAddr, String protocol) |

# 第四章 智能家居本体建模与规则推理

在前文中，详细描述智能家居自适应控制系统的框架，阐述整个系统的交互逻辑流程，并提出本文重点研究的应用层的三个研究要点。下面将从自适应控制系统应用层的智能家居本体库的设计与建模，以及基于本体库的规则推理扩展应用两方面展开。

## 4.1 智能家居本体库规划与实现

在调研国内外多个智能家居本体模型后，发现至今为止，还没有一个标准全面的、充分考虑设备异构性特点的本体模型能够为大家作为数据共享使用，为此，本小节将借鉴成熟的本体建模方法，分析实际智能家居场景中的信息和需求，设计一个尽可能完备的智能家居本体模型，最大程度的为以后的研究者提供研究基础。

### 4.1.1 本体建模方法和原则

根据描述对象和应用领域的不同，本体被划分为多个种类[45]，每种类型的本体也在不同的研究方向有着用武之地，根据本体对领域内容的依赖程度，可以将本体分为四大类，具体划分如表4.1所示。考虑到智能家居作为一个具有完整情景和特定性质的概念体系，本文主要采用领域本体的构建思路来构建智能家居本体库。

表4.1 本体分类

|  |  |
| --- | --- |
| 名称 | 特点 |
| 通用本体 | 覆盖通用概念的本体，独立于特定问题和领域 |
| 任务本体 | 特定任务或行为的概念关系 |
| 领域本体 | 特定类型领域或方向的相关术语，领域内可重用 |
| 应用本体 | 描述特定应用，可复用任务和领域本体概念 |

在具体的本体构建方法上，国内外提出多个指导本体构建的方法，比如常见的有TOVE法、METHONTOLOGY法、SENSUS法、KACTUS工程法和七步法等[46]。在综合分析以上的本体构建方法后，考虑本文中所采用的智能家居本体的思路，最终选用斯坦福大学医学院所开发的七步法的本体构建方法。具体步骤如下：

1. 确定专业领域和范畴：明确本体所覆盖的专业领域范围、作用和最终应用目标，本体在哪些方面发挥着作用，以及考虑所存在系统的维护者与应用对象。在刚开始构建本体时，确定每个工作阶段的范围和内容，最终有助于形成复杂本体模型。
2. 考虑复用现有本体：充分考虑现有的领域本体，是否存在他人或组织构建的本体可以共享和重用，以此来杜绝可能重复的劳动，也可以用来验证自己所构建本体是否正确。
3. 列出领域中重要术语：领域本体描述的是现实世界中概念之间的关系，要达到此目的，需要首先根据现实事实，列举出该领域中的所有存在的概念，并为这些概念定义其可能存在的属性以及对应的属性值等，暂时不考虑概念间的重复性。
4. 定义类的等级体系：类等级体系有三种方法，分别是自顶向下法、自底向上法和综合法。自顶向下法是从领域中最大的概念开始，然后再细分成更小的概念；自底向上法是从等级体系树的最小概念开始，然后逐渐合并成更大的概念；综合法是结合自顶向下法和自顶向上法，从定义领域中显而易见的概念，然后在对这些概念进行合理的划分和归纳。在实际本体构建中，选择哪种方法作为等级体系的构建方法，取决于开发者的实际需求。
5. 定义类属性：有了枚举的概念，和概念抽象成类之后的层级关联关系，那么该步骤中需要考虑概念内的信息，也就是类的属性。
6. 定义属性分面：在对属性进行赋值描述的过程中，从赋值类型、可选赋值和赋值的基数，都需要被详细定义，以便更明确本体内容。
7. 创建实例：构建本体的最后一步，之前的步骤已经能够通过本体看到领域的大致雏形，通过现实世界中存在的信息内容或者实体，向本体中添加具体的信息，不仅充实本体，同时也可以验证本体的有效性和完备性。

以上列出的七个步骤，将成为下文构建智能家居本体库的纲领性指导方法，除了遵循上面的方法外，在构造本体过程中，还需要充分考虑本体的设计原则，以使构建的本体在日后的研究中能够为其他研究者所共享和复用。在过去众多研究者提出的设计原则中，以Gruber提出的五大原则最为广大本体构建者所推崇。五大原则分别是清晰性、一致性、可扩展性、编码偏好程度最小和最小本体承诺[47]，详细释义如下：

1. 清晰性：本体对所定义术语给出客观明确语义定义，能有效传达属性含义。
2. 一致性：本体支持与其定义相一致的推理，得出的推论不产生矛盾或冲突。
3. 可扩展性：设计本体的时候要使用领域内通用的词汇术语，还要考虑可能的应用范围，使得本体扩展时不用修改原有内容。
4. 编码偏好程度最小：概念不应依赖于特定符号编码。
5. 最小本体承诺：对待建模对象给出的约束尽可能少。

除了参考以上列举的本体构建原则外，考虑到智能家居信息的多维度性和智能家居信息表达的通用和互操作性等问题，构建本体时采用OWL DL来进行本体描述，并充分利用本体构建的相关工具。

### 4.1.2 智能家居本体特征分析

在智能家居场景中，所包含的信息内容可以划分为多个维度，比如智能家居的主要参与者用户，被控制的设备，智能家居场景的划分，空间的划分，服务的定义等。遵循七步法和本体设计原则，通过分析并提炼多个维度的术语，并基于其上构建维度之间的关系网络，最终使用本体构建工具Protégé来搭建和完善智能家居本体库。

在智能家居环境中，物理空间的布局和划分是家居用户和设备存在的基础，用户在各个家居空间中进行日常活动，而设备则被安装和部署在各个空间中。为此，家居空间成为智能家居本体库的一个关键概念，而其中的家居用户又是另外一个关键概念。结合这两个关键因素，发现空间中的设备也是不可或缺的部分。设备本身是物理硬件设备，软件层面的控制器等功能可以对外提供多种服务，而家居用户在家居环境中表面上控制的是各个设备，而实际上是每个设备所提供的服务，以此来使用设备相对应提供的功能，所以，服务也是一个关键概念。现在分析智能家居特征，暂时得出家居空间、家居用户、设备、服务这四个本体概念。但是将这几个概念关联起来的因素也可能成为关键概念，比如环境因素，家居用户因为感知到家居环境的变化，对家居环境有一定的需求和想要做出改变的意愿，才去控制设备，调用服务达到这一目的，所以家居环境也是一个非常重要的潜在关键概念。除了这些以外，根据设备中间件中消息协议的规范，提供设备进行消息传输的各个网关，也将成为本文中另一关键本体概念。为此，本体概念分为家居空间、家居用户、设备、服务、家居环境和家居网关这六个关键概念。

参照七步法的第四步，定义类（概念）之间的等级关系。结合之前得出的六个关键概念，发现概念之间更多的是平级概念，相互之间有着一定的关联关系，所以，采用综合法来对概念层级关系进行划分。而且每个独立的概念之间也有密不可分的关系，比如家居用户是定位在家居空间中，设备也绑定在家居中间中，而服务和设备也是相互绑定；服务和环境因素有一定的关联关系；设备绑定在对应的通信网关下等，这些成为七步法中定义类属性中的内容。智能家居本体库所需构建信息来源于实际的智能家居情景，但是从哲学层面又高于家居情景，是对其更高层次的抽象和统一。

在第三章中详细论述家居空间的类型和划分方式，以及考虑到设备的异构性和多样性，根据设备提供服务的能动性上，将设备划分成感应设备和控制设备。除这些更具体的划分外，家居用户有姓名、性别、年龄、环境偏好等属性，家居环境有类型、单位、数值等属性，网关也有唯一标识、协议等属性。这些属性的充实和添加，才能够让本体更加丰富，更能全面展现现实智能家居情景的信息。

### 4.1.3 智能家居本体库设计

图4.1 智能家居本体库概念属性关系图

根据前文对智能家居本体特征的分析，提取六个关键概念，分别是家居用户 User、家居空间 Room、家居设备 Device、设备服务 Service、家居网关 Gateway和家居环境 Environment。六大本体概念之间的定义相对独立，没有重叠，并通过概念属性构建概念之间的关联关系。在此，举几个本体概念属性的样例，考虑到易读性和之后设计开发的方便性，属性的命名采用的格式，其中。对于User和Room，通过属性和Room\_belongTo\_User来构建关联关系；User和Device，通过User\_hasExclusiveDevice\_Device和Device\_occupiedBy\_User来建立联系；Room和Device，通过Room\_hasDevice\_Device和Device\_locatedIn\_Room产生关联；Room和Gateway通过Room\_hasGateway\_Gateway和Gateway\_locatedIn\_Room建立联系等，具体的类属性关系一共有13个。详细的本体概念属性关系图如图4.1所示。

根据智能家居情景和每个概念的特点，将每个本体进行更详细的划分。最终构建智能家居本体库结构如图4.2所示。

用户User：User本体定义智能家居情景中实际参与的用户，记录用户的基本信息，如用户id、用户年龄、用户性别、用户名字、用户当前状态等。

家居空间Room：Room本体根据空间性质，被分为三种类型，分别是PrivateRoom、PublicRoom、ServiceRoom。对于每个子Room本体，从使用方式上分成更详细的子空间。该子本体的划分也是严格依照现实生活场景。除此之外，该子本体都具有记录Room信息的id、名称、类型等数据属性信息。

家居设备Device：根据设备的属性，分为控制设备ControlDevice和感知设备SensorDevice。考虑到现实中存在具有感知功能的控制设备，所以列出复合设备CompositeDevice。在该本体中，基于常见家居设备的分类，也对各个子本体进行对应概念的创建，以充实本体内容。Device本体中包含记录设备相关的设备id、设备传输协议、设备状态、设备类型、设备相关环境类型和设备服务类型等关键信息。

设备服务Service：该本体根据服务层所提供的底层物理设备的服务来封装。根据设备分类，也分为两种服务，分别是控制服务ControlService和感知服务SensorService。Service本身也包含id、名称、函数名称、服务状态和服务类型等数据属性信息。

家居网关Gateway：家居网关是设备的通讯通道，所以和设备有着密切联系。Gateway本体包含id、名称和状态等数据属性信息。

家居环境Environment：该本体主要用来记录家居环境中可记录和被用户所监控的环境参数信息，比如房间的适宜温度范围、厨房的合法烟雾浓度范围等。该本体所需包含的数据属性信息有环境参数名称、环境参数单位、环境参数上限、环境参数下限等。

图4.2 智能家居本体库结构图

图4.2详细展示各个子本体之间的交互关系以及所包含的必要属性，并列出每个子本体所派生出来的子类。在智能家居本体库中的Room类、Environment类和ControlDevice类均派生出多个子类，此处之所以如此设计，是考虑这些子类均有其父类的共性，比如数据属性，但是又有各个子类因自身差异所带来的数据属性和类属性。构建智能家居本体库的过程中，根据实际智能家居情景和情景中的物体，来灵活的选择是添加新的派生类还是作为实例存在。

该本体库的构建，借助于本体语言OWL DL来描述，通过使用斯坦福大学发明的本体构建工具Protégé，并依照上面设计的架构图，使用Protégé软件手动添加并定义本体库所需要的类信息，定义类层级关系，添加类的数据属性信息和数据类型属性，添加类与类之间的对象属性，还有必要的标注属性，传递属性，函数型属性等。

### 4.1.4 智能家居本体库完善

经过前文的分析和设计，智能家居本体库已初具形态。为了充实该本体库，使得虚拟的抽象智能家居本体可以和现实世界的人、物、环境构建关联关系，则需要进行七步法的最后一步——创建实例。而要真正做到自适应控制，首先需要向本体库中添加设备和设备服务的相关信息，以让自适应控制有目标可控制；其次，需要用户的个人信息，以记录服务调用数据，存入历史行为库使用；最后，需要添加环境的相关约束信息，以提供用户对设备产生控制的依据。设备信息、服务信息、用户个人信息和环境约束信息，是智能家居本体库进行数据收集、建立完整数据源，来充实本体的过程，也是自适应控制系统在智能家居情景应用下的一个必须过程。这些信息都以一个一个实例的形式添加在本体中。

以上四部分的信息采集，可以分为[48]显示方式和隐性方式，前者表示需要主动参与者直接进行相关操作添加信息，而后者则是由其他已发生操作所产生的数据来间接完成信息的采集过程。根据用户个人信息、环境约束信息、设备信息和服务信息的数据特点，用户个人信息和环境约束信息采用显示方式，也就是用户直接参与的方式进行数据添加，而设备信息和服务信息则采用间接收集的方式。因为用户个人信息为个人隐私方面的信息，而且对于实际环境指标的要求也具有较大的差异性，除自适应控制系统使

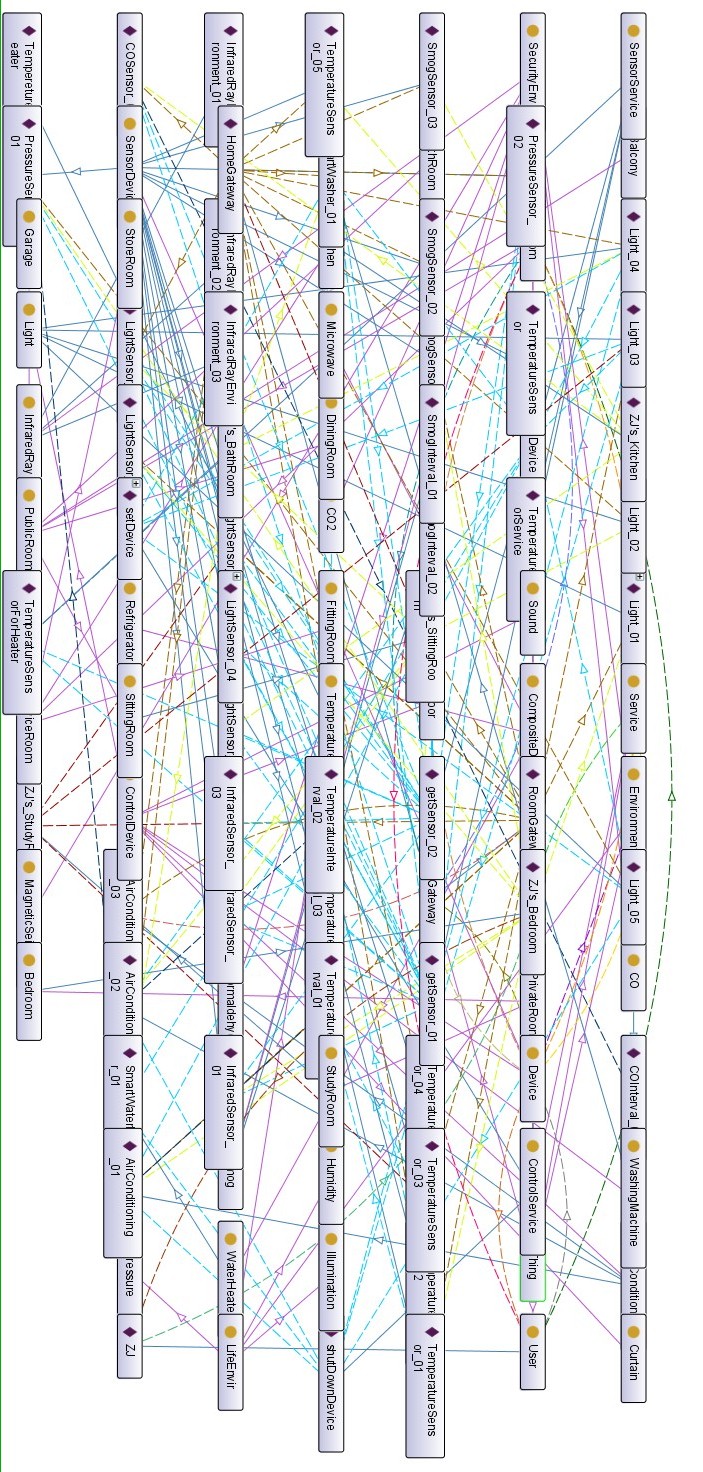
用者之外的人添加都毫无意义。而设备信息则根据设备中间件层进行设备注册和信息录入过程中的数据，半手动半自动完成，服务信息则通过对服务层所封装的服务来自动完成。

图4.3 智能家居本体库网状图

对于用户个人信息，不仅仅局限于对User本体中的数据属性，还包括用户联系方式、生活所在地、工作性质等个人信息，这些信息只需要一次添加；对于环境约束信息，往往和设备相关联，所以如果新添加设备或者服务，则需要用户针对相对应的设备或服务添加该设备所在空间的环境约束信息。该约束信息主要用于设备或服务的自启动、自关闭和报警等需要自适应控制的情景下。对于设备信息，设备中间件层对底层设备进行异构处理的过程中，已经详细记录设备的id号，名称，类型，可选状态，可选动作，协议类型，所属的网关，所属的地理位置等。

添加实例后，最终得到的智能家居本体库网状图如图4.3所示。在本文的智能家居本体库中，添加两个网关实例“HomeGateway”和“RoomGateway”、一个房间实例“ZJ’s\_Bedroom”，而“RoomGateway”则在房间“ZJ’s\_Bedroom”中；该网关下有“TemperatureSensor\_01”、“LightSensor\_01”、“Light\_01”、“AirConditioning\_01”、“PressureSensor\_01”、“SmogSensor\_01”等设备实例，其中相关的环境实例有“TemperatureInterval\_01”和“SmogInterval\_01”。

## 4.2 基于规则推理的自适应控制

在前一节中，分析了智能家居本体库的关键元素，阐述本体库构建的过程，并展示添加现实实例信息后所呈现的完整本体库网状图。在错综复杂的关系网中，根据分析交互层处理后的用户指令，使用规则库中的规则，从关系网中推理出符合用户需求的目标服务。在该节中，将详细讨论分析如何基于Jena设计规则，使用规则，并基于本体推理来实现到智能家居中的初步自适应控制。

### 4.2.1 规则推理场景需求分析

本体库由类、概念属性、数据类型属性、数据、实例等元素信息构成，类与类之间通过概念属性构建关系，实例与数据之间通过数据类型属性构建关联，而类与实例之间则通过子类关系产生联系。这些关联关系，如果用自然语言来表述，则是一句完整的描述语言，比如“Light\_01是Light的实例”，那么根据语法结构抽取其中的重要信息，可以得出“Light\_01”、“Light”、“实例”等信息。在本体中，将元素与元素间关联关系，转化为的三元关系，如，这样的关系被称为三元组[49]，其中subject的类型属性被称为定义域，object的类型属性被成为值域。整个本体库中的所有元素和元素之间关联关系，通过构建多个三元组，则可以形成一个错综复杂的关系图，该关系图的边为有向边，即得到一个有向关系图。

添加实例的智能家居本体库，其中有详细的类定义，有类的实例信息，通过读取该本体库的信息，可以获取若干显而易见的三元组关系。但是当处理来自交互层的请求时，比如语言指令“在卧室我可以控制哪些设备？”通过语义解析和本体资源转换后得到如下详细指令：“ZJ’s\_Bedroom”、“User\_controlDevice\_Device”，但是该关系“User\_controlDevice\_Device”并不在原本体库的关系内容中，而是由已知关系衍生出来的新关系，此时，则需要添加一些新的规则来应对变化的需求；除此之外，原本体库中构建的类关系和添加的实例信息，存在一些显而易见的特性，比如三元组表示“UserA 居住在RoomX”，对于三元组关系，表示“RoomX 属于UserA”，这两个三元组均成立，且livesIn和belongTo的定义域和值域发生了改变，则认为livesIn和belongTo这两个类属性是互逆属性。类似于逆属性这一类特性属性，在OWL DL描述语言的约束中还有很多，这些属性的使用和声明，很大程度减轻了手工构建本体库中的工作量，而且可以帮助推理出更丰富的三元组内容。

除了通过添加规则的方式来增加新属性和推理潜在三元组信息外，还可以通过分析实际生活情景信息，确立符合事实的情景规则链，来推导最终期望发生的事实，以此形成规则，比如，厨房的CO感应器可以监测厨房的CO浓度，当监测到CO感应器所探测的浓度值超出阈值时，则向用户预警。除了CO感应器，智能家居中的烟雾感应器、温度感应器等设备，在感知环境后，都可以做出一定的响应行为，以完成用户所期望的情景控制。对于这一类设备的情景应用，则可以添加情景规则的方式，根据感知情景信息，触发相应的动作。除了预警类的情景，用户还可能因为自身家居状态的不同，有一些对应的个性需求，比如用户处于睡眠状态中时，其房间的灯处于关闭状态，窗帘也合上，空调打开并调节到26℃；用户工作日下班回到家中，期望客厅的灯打开，空调打开等。这些用户所处的情景模式或者行为状态，往往与多个设备之间有联动的关系。对于以上列举的环境预警和用户状态情景，是智能家居中典型的应用情景，均可以通过自定义规则的添加，结合添加属性规则的本体库完整信息，来推理出符合智能家居用户和情景需求的服务，以达到自适应控制目的。

该小节中列出智能家居自适应控制所需要的两种类型的规则，一类是丰富本体库内容的属性规则，包括特性属性的声明和新属性的添加；另一类是通过分析情景需求，根据事实(facts)链来构建自定义规则，当情景匹配时，则触发对应的事件。本节中，也主要针对这两类规则进行分析研究和实现。

### 4.2.2 规则库定义

逻辑是本体知识表达的基础，为了更好的利用规则，推理出更丰富的知识，文中所提出的规则，语法遵守Jena，逻辑遵循谓词逻辑[50]（predicate logic）。在谓词逻辑中，规则的形式如下所示：

其中的PreR1， … ，PreRn 以及NewR均为原子公式 (atomic formula)，→ 的左半部分是该规则的前提(premise)，→ 的右半部分则是规则的结论(conclusion），当前提中的每一个原子公式都为真时，则认为规则的结论为真。而对于每一个原子公式，都是一个三元组关系，即。在三元组中，subject和object可以是变量（variable），也可以是常量（constant），而link可以是谓词（predicate）或者函数符号（function symbol），当link为函数符号时，三元组也可以表示成。例如智能家居本体库中的三元组，其中U为变量，25为常量，User\_hasAge为函数符号，则该三元组可表示为；而对于三元组中，Type则为谓词，表明变量U是User类的实例。下文中规则的声明和定义则都来源于以上谓词逻辑和三元组的知识。

对于属性规则的定义，首先分析本体中属性的种类和内容。类属性，即概念属性，在特性上有传递性、对称性、非对称性、自反和反自反等特性属性，还可以添加类属性相关的定义域和值域；数据类型属性，可以为其添加全称限制、存在限制、值限制、基数限制、数据类型限制等限制内容。为此，根据以上规则定义，来定义属性规则。

【定义1】：L表示link集合，即谓词和函数符号集合，V表示subject和object的集合，即变量和常量集合，对于接下来的定义有，。

【定义2】：如果，则表示关系l具有传递性，则该属性规则定义一个传递性规则。

例如，在本体库中Room类是PrivateRoom的父类，PrivateRoom是Bedroom的父类，那么基于父子类的传递性，有Bedroom是Room的子类。该定义中的传递性关系常发生在父子层次关系中，以及部分-整体关系中。

【定义3】：如果，则表示关系l1和关系l2具有互逆关系。

例如，对于任意一个设备D和房间R建立Device\_locatedIn\_Room关系，即（D, Device\_locatedIn\_Room, R），那么就一定有（R, Room\_hasDevice\_Device, D），此时Device\_locatedIn\_Room和Room\_hasDevice\_Device这两个属性就是互逆关系。

【定义4】：如果，则表示关系l1和关系l2可以产生新的衍生关系l3。

例如，智能家居本体库中已经存在的关系有User\_livesIn\_Room和Room\_hasDevice\_Device，而此时交互层请求的是User和Device的类属性关系，但是在原本构建本体库的过程中并没有这两者的直接关系，为此，可以通过现有的类属性衍生出新的类属性关系，来丰富交互层资源映射库的内容。比如处理请求“在卧室我可以控制哪些设备？”时，可以衍生类属性关系User\_controlDevice\_Device，来表示在房间中的用户可以控制的设备，即，通过添加新的类属性关系，可以在不改变原本体库的情况，添加原本并未考虑却实际需要的关系，依次减少系统的耦合性。

除了以上定义的经典特性属性规则外，在本体中常见的还有对称属性、函数式属性等特性属性规则。以下是本文中智能家居本体库推理过程中用到的部分属性规则，其中包含互逆属性、衍生属性、定义域、值域等属性规则，规则前提都满足时，则可以推出规则结论，得到更丰富的三元组关系。其中rule1-rule6定义的是两个关系的互逆性，而rule7-rule9定义的是定义域、值域和子属性的规则，rule10定义的是衍生关系的规则，最后一个规则定义的是传递性规则。

|  |
| --- |
| @prefix sh: <http://www.semanticweb.org/zj/ontologies/2015/5/SmartHome#>.  @include <RDFS>.  @include <OWL>.  [rule1: (?x sh:Device\_belongTo\_Gateway ?y) ->  (?y sh:Gateway\_hasDevice\_Device ?x)]  [rule2: (?x sh:Device\_hasService\_Service ?y) ->  (?y sh:Service\_relatedToDevice\_Device ?x)]  [rule3: (?x sh：Device\_locatedIn\_Room ?y) ->  (?y sh:Room\_hasDevice\_Device ?x)]  [rule4: (?x sh:Device\_occupiedBy\_User ?y) ->  (?y sh:User\_hasExclusiveDevice\_Device ?x)]  [rule5: (?x sh:Room\_belongTo\_User ?y) ->  (?y sh:User\_livesIn\_Room ?x)]  [rule6: (?x sh:Gateway\_locatedIn\_Room ?y) ->  (?y sh:Room\_hasGateway\_Gateway ?x)]  [rule7: (?x ?p ?y) , (?p sh:domain ?c) -> (?x sh:type ?c)]  [rule8: (?x ?p ?y) , (?p sh:range ?c) -> (?y sh:type ?c)]  [rule9: (?a ?p ?b), (?p sh:subPropertyOf ?q) -> (?a ?q ?b)]  [rule10: (?x sh:User\_livesIn\_Room ?y),  (?y sh:Room\_hasDevice\_Device ?z)->  (?x sh:User\_controlDevice\_Device ?z)]  [rule11: (?x ?p ?y), (?y ?p ?z), (?p sh:type sh:TransitiveProperty) -> (?x ?p ?z)] |

在Jena的规则推理机中，为了丰富规则推理的形式和内容，还存在原语（primitive）。每个原语可以被用于规则，作为原子公式存在于规则前提或者规则结论或两者兼而有之。原语中可以绑定变量和常量，也可以是谓词关系。以下列举本文中认为比较重要且实际需要的原语，如表4.2所示。

表4.2 Jena内置原语及其释义

|  |  |
| --- | --- |
| 原语 | 含义 |
| equal(?x,?y) notEqual(?x,?y) | 测试x和y是否相等 |
| lessThan(?x,?y) le(?x,?y)  greaterThan(?x,?y) ge(?x,?y) | 测试x是否<,<=,>或者>=y,其中x，y的类型必须一样 |
| now(?x) | 返回当前时间，形式为xsd:dateTime |
| makeTemp(?x) | 创建新的类/数据类型属性 |
| isDType(?x,?t) notDType(?x,?t) | 确定x是不是t的类型 |
| hide(p) | 隐藏谓词为p的三元组信息 |
| listContains(?t,?x)  listNotContains(?t,?x) | 确定t中是不是包含元素x |

以上列出的原语，对于构建情景匹配的自定义规则具有重大的意义，这些原语的使用不仅可以根据实际需求灵活的增加本体库中的节点信息，还可以通过赋值和比较等方式控制复杂多变的逻辑。在本文中，主要定义两类自定义规则。第一类是根据现实情景中环境参数来定义预警或联动操作的规则，第二类是根据用户所处状态或者环境模式信息来定义服务规则。

【定义5】：规则，表示前提事实为真，则结论为真。

例如，设定房间的温度阈值为10℃~36℃，当房间的温度感应器感应到房间的温度超出该阈值范围的时候，则开启空调，并调至为默认的26℃。此时，将该信息转化为定义的模式，则是(房间，有温度值，x)，(房间，设定温度下限，y)，小于(x,y) → 开启(空调，26℃)。

【定义6】：规则，表示前提为真，则触发后续的多个结论。

例如，当用户处于睡眠状态时，则希望有一个良好的睡眠环境，如房间灯关闭，窗帘合上，空调调至适宜温度。此时，则可以得出该规则模式，(用户，状态，睡觉) → 关闭( 灯 ),关闭( 窗帘 ),开启(空调，26℃ )。

根据以上规则的定义，在该智能家居本体库的规则库中，也根据场景需求添加自定义规则，如下样例所示，rule12表示当某房间有环境参数相关的设备，同时该房间有环境参数阈值，那么就通过获取该设备感知的环境值，并比较阈值范围，如果超出阈值，则报警；rule13表示当床上的压力传感器感知到压力达到一定的值时，则判定用户处于睡眠状态；而rule14则表示用户处于睡眠状态时，则将房间的等关闭，且设置空调。

|  |
| --- |
| @prefix sh: <http://www.semanticweb.org/zj/ontologies/2015/5/SmartHome#>.  @include <RDFS>.  @include <OWL>.  [rule12: (?x sh:Room\_hasLimitEnvironment\_Environment ?y),  (?y sh:Environment\_hasUpperLimit ?i),  (?x sh:Room\_hasDevice\_Device ?z),  (?z sh:Device\_relatedToEnvironment\_Environment ?y),  (?z sh:Device\_hasAction get),  (?z Device\_doService getSensor\_02),  (?z sh:Device\_hasValue ?e),  greaterThan(?e, ?i) -> (?z, sh:warning true)]  [rule13: (PressureSensor Device\_doService getSensor\_02),  (PressureSensor Device\_hasValue ?i),greateThan(?i,100) ->  (ZJ User\_hasState Sleeping)]  [rule14: (ZJ User\_hasState Sleeping) -> makeTemp(Sleeping),  (Light\_01 Device\_doService shutDownService),  (AirConditioning\_01 Device\_doService setService)] |

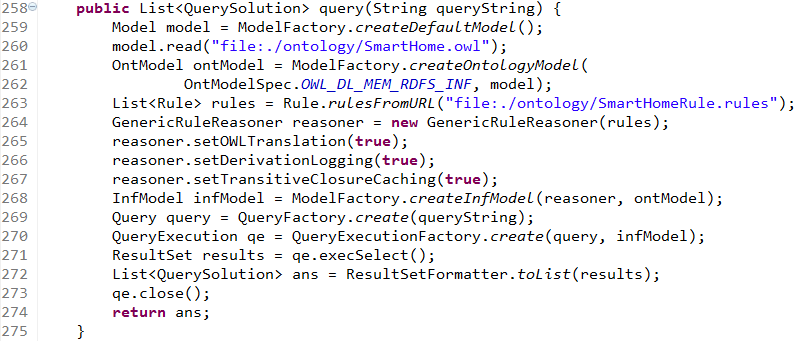
规则库中的属性规则和自定义规则，可以直接通过文件的方式手动添加和修改，相比于修改实际的本体文件和逻辑代码，有明显的优势。为此，本小节遵循Jena规则语法，详细定义智能家居自适应系统中所应用的规则，为之后处理请求到服务的推理流程提供良好的基础。

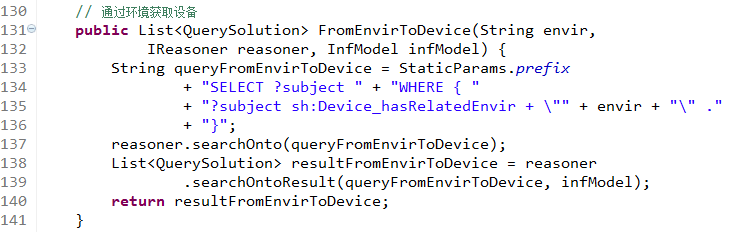
### 4.2.3 推理流程设计与实现

在前文中，对智能家居本体库的构建过程进行详细的描述，并说明各个子本体之间的类属性和每个子本体的数据类型属性，并依据现实情况在本体库中添加相应的实例信息；定义规则库中两类规则——属性规则和自定义规则，并结合本体库做样例说明。接下来，将根据建立成功的智能家居本体库和规则库，阐述如何完成智能家居系统中的自适应控制过程。

交互层获取用户指令后，需要将指令进行语义拆分、解析，并和本体资源比对映射，转换为智能家居本体库能够理解的资源信息，转化并提交的参数包括本体层面的用户位置、指定设备、环境参数、执行方式、执行时间、执行参数等内容。应用层获取这些输入信息后，首先根据用户位置、指定设备、环境参数等信息推理出用户所处空间中相关设备；在3.4小节中已经详细说明，设备的最终Id号为由厂家编码、网关编码和设备编码决定，而此时获取的设备，仅仅包含其中的设备编码，为此，需根据注册的设备信息找到相关的网关标识和网关协议，以得到完整的设备标识；通过交互层提供的执行方式，得到用户指令中所期望的设备执行方式，如果没有明确信息，则默认为“get”；获取设备信息以及执行方式后，则根据指令中的时间信息确认是立马执行设备服务，还是指定时间执行，最终调用服务，控制设备。详细执行模式如图4.4所示。

图4.4 推理模式图

根据空间、设备、环境参数、动作、时间等参数的完整程度，推理出相关设备、调用服务的过程主要有四种模式，分别是“11000”，“11001”，“01101”，“01111”，其中“0”代表该参数不完整，“1”表示该参数完整，每一位按照顺序分别对应着环境参数、空间、设备、时间、动作。“11000”表示有明确的环境参数和空间信息，没有明确的设备，时间和动作，则认为是获取当前位置下的环境值，比如获取当前房间温度，后面三者则默认为“now”和“get”。“11001”表示有明确的环境参数、空间信息和动作，没有明确的设备、时间，则认为获取或者改变当前环境下的状态，比如改变当前温度到26℃，则时间默认为“now”，动作为“set”，则推理可以改变温度的可设置设备。“01101”表示有明确的空间信息、设备、动作，没有明确的环境参数、时间，则根据空间的特定设备和动作，改变其状态，比如设置房间的空调到26℃，则默认时间为“now”。“01111”表示有明确的空间、设备、动作。时间，没有明确的环境参数，则根据指定设备调用对应服务，且根据时间，在一定时间后或者现在调用服务，比如1小时候把房间的灯打开，环境参数默认为设备相关的环境参数。

通过进一步封装和使用Jena提供的本体操作接口，可以读取和修改OWL本体库文件，并读取和解析规则库文件，初始化推理机，获取推理模型。样例代码如上所示。对于指定的推理，通过SPARQL语句，定义需要查找的源头和条件，则可以推理出需要的目标。示例代码如下所示。

通过对Jena接口的灵活使用，结合可进行本体查询的SPARQL，可自由地根据用户指令的各个维度的请求，结合规则库中各项规则，向底层的设备中间件发送设备的控制指令，以达到控制目的。

对于自定义规则，主要体现周边环境变化时所带来的目标设备-服务变化，或者多个设备服务的发生引发其他目标设备-服务的发生。所以，当规则前提中所关联的环境因素变化时，或者规则前提中相关设备-服务发生变化，导致规则前提为真时，需要严格执行规则结论中相关的设备-服务，否则，自适应控制系统的自适应性无从谈起。为此，规则库中的自定义规则，则需要被反复验证，以确认是否需要执行相关设备-服务。在本文的实现过程中，采用开源作业调度框架Quartz,将每条自定义规则与一个定时子线程进行绑定，注册成一个单独的子任务，每个子任务有触发器和监听器来保证规则在指定时间内被反复执行，在该子线程运行时，通过截取规则前提中的每个子公式，验证真伪，最终确定规则中的前提是否为真，如果为真，则解析规则结论，查询并推理结论中对应服务。除了自定义规则，用户发出的延迟处理请求也需要进行任务的注册，并进入线程池中，等待指定的执行时间。

# 第五章 潜在规则挖掘

在前面的章节中，阐述了智能家居场景下自适应控制系统的架构和分层设计，并详细描述智能家居本体建模和基于规则的自适应控制过程。家居场景下的用户通过控制系统的使用，将产生日常操作数据，这些数据记录是用户日常行为数据，包括在何时控制什么设备，调用什么服务。而这些数据的深入分析，将有助于进一步形成潜在规则，促进该系统的自适应程度。在本章中，将通过原生系统场景下获得的用户行为-服务数据，基于关联规则挖掘算法，寻找潜在规则，为该自适应控制系统的设计提供一些思路。

在本文中，用户数据的形式主要是系统在什么时候调用什么服务，这些数据包含用户的行为特点和生活习惯。通过对用户日常数据的处理和分析，可以初步知道服务与服务之间的关联关系，以及时间与服务调用的关联关系。接下来，首先阐述实验用到的算法和思路，然后描述实验所使用的数据集和预处理的过程，最后将从两个方面进行数据挖掘，并分析实验结果，以确认该研究思路的可行性。

## 5.1 关联规则概念与定义

关联规则是数据挖掘中的经典问题之一，针对该问题也产生了多种高效的算法。在本文中，考虑到数据规模和数据模式，主要采用关联规则的经典算法—Apriori算法。该算法应用的典型例子是购物篮分析，该过程通过发现顾客放入他们“购物篮”中的商品之间的关联，来分析顾客的购物习惯[51]。其中，购物篮中包含多个商品，每个商品是购物篮中的项，购物篮则为商品所构成的项集。假定，所有商品组成一个集合，每种商品是否被购买用一个布尔变量来表示，那么用一个布尔向量则可以表示该购买状态，通过分析布尔向量，则可以反应不同商品被同时购买的购买模式，这些模式可以用关联规则（association rule）的方式来表示。

针对以上问题，定义为所有项的集合，类似于商场中的所有商品，而事务T则是由I中的若干项组成的项集，有；定义事务集合,表示数据集中所有的数据，这些数据由一项一项的事务T组成，每个事务T都有唯一的标识信息，来区别每一条事务数据。现有X,Y两个项（项集），且在数据集（事务集合）中的多个事务中包含该两项（项集）。那么定义的蕴涵式为从事务集合中挖掘出的X与Y的关联规则，且有,,,,，该规则表示如果X出现在某个事务中，那么Y就有一定的概率也出现在事务中。

对于关联规则的度量，则通过支持度（support）和置信度（confidence）来分别表示所发现规则的有用性和确定性。对于包含X,Y项集的关联规则，支持度表示在事务集合中，包含的比例，比如啤酒尿布的支持度为2.5%，则说明所分析的所有事务中，2.5%的事务里含有啤酒和尿布；置信度则表示在事务集合中，包含X的事务同时也有Y出现的比例，比如啤酒尿布的置信度为70%，则说明购买啤酒的客户中70%也同时购买尿布。以上定义用公式表示如下所示：

根据关联规则挖掘中对频繁项集的定义，最终挖掘的关联规则如果满足最小支持度和最小置信度，则称为强规则。

Apriori算法则是根据以上定义，首先通过查找满足最小支持度的项，从频繁度为1的项集，找到频繁度为2的项集，依此类推，直到找不到更大的频繁k项集，如此形成的集合则为大于最小支持度的所有频繁项集；接下来找出这些频繁项集中大于最小置信度的频繁项集，生成最终的强关联规则。本文主要遵从该算法的思想来进行接下来针对智能家居场景下用户数据的挖掘，通过对数据的预处理，和内容划分，再在各子数据集中采用Apriori算法来挖掘服务与服务之间以及服务发生时间与服务之间的关联关系。

## 5.2 数据内容与预处理

智能家居场景下的用户数据，主要记录用户在何时调用过何种服务。在第四章中，本体库设计的六个维度都有各自的属性信息，比如name， id等，而最终目标服务的调用，则采用每个维度的部分关键属性信息予以记录，即“用户在什么时候调用了哪个设备的什么服务，调用服务的参数是什么，最终服务的执行结果如何”，此时被用到的维度属性信息分别是：Userid，Sensorid，Sensorname,date,time,Serviceid,Servicename,Serviceparameter,Result。在自适应控制系统应用层的历史行为库中，则根据以下格式对用户数据进行保存：

但是考虑到实验的完备性和现有实验环境的局限性，所采用的数据为麻省理工大学某实验组所采集的实验数据[52][53]，该实验通过60个传感器和控制器，包括灯、窗户、门、电器、水管等设备，记录用户在16天中的家居行为数据，数据内容和数据格式与本文实验设想大体一致，为此，本实验主要依托该数据进行关联规则挖掘，以验证本文架构设想的可行性。

该数据主要包含两份数据，记录日期分别是2003-03-27至2003-04-11和2003-04-19至2003-05-04。每份数据中包含三份文件，Activities.csv、activities\_data.csv和sensor.csv，第一个和第三个文件记录着行为和设备与其对应编号的映射关系，而activities\_data.csv则是以时间流的方式记录用户的行为数据。数据格式为：

该数据首先记录一个行为，并详细记录该行为的开始时间和结束时间，接下来的四行，则分别记录该行为事件下的各个设备相关的子事件，包含设备的id、设备名称、设备服务调用开始时间和结束时间。

该实验的目标是想通过用户的日常行为服务调用数据，挖掘出这些数据中潜在的关联规则，包括服务与服务之间调用的关系，和服务与调用时间的关系。如果想通过关联规则挖掘的方式寻找这种联系，则所处理的数据中，服务必须出现在一个事务中。但是对于该数据内容，无法直接进行挖掘，需要根据实验目标进行数据的预处理，并剔除其中的无效和干扰数据。为此，本文采用如下的预处理方式。

首先，对于没有发生过或者发生次数特别小的Sensor不予考虑，将相关数据从数据集中剔除；其次，在剩余的数据中，通过解析数据文件，将所有的数据拆分为“ID, date, start, end, sensorid, sensor, tag”的形式，并保存在MySQL数据库，获得的数据内容如图5.1所示，共7000余条数据。

图5.1 数据样例

以上数据依旧没有办法满足挖掘条件，为此，结合智能家居场景和用户家居行为特性，以半个小时为一个时间单元区间，将所有的数据根据服务开始时间，划分到对应的时间区间中，即将00：00：00至23：59：59划分为48等分；考虑到如此划分可能弱化相邻区间服务的关联度，所以错开15分钟，即将00：15：00至23：44：59分成前47等分，再加上剩余时间中的服务，同样获得48等分。根据以上的划分方法，将服务映射到各自时间区间中，以获取最终的数据data1，该数据用以挖掘服务与服务之间的关联规则，即哪些服务的调用和另外一些服务关联性较大，该部分的挖掘并不考虑服务关联在时间跨度上的影响。

遵循之上数据划分的方式，同样将数据根据时间区间的方式进行划分，与之前不同的是，将时间区间调整为20分钟，即每20分钟为一个区间，将16天中同一区间的数据进行整合，获取若干数据文件。考虑到用户可能处于夜间的睡眠状态，或者长期外出导致时间区间内数据的缺失，为此，对于某一个时间区间内，数据量不满足一定量的将予以剔除。如此，最终可以获取最多58个文件data2-x（1-58），每个文件记录着16天中对应时间区间内的服务调用情况。该数据用以挖掘时间区间与服务之间的关联关系。

为此，基于关联规则挖掘的自适应控制系统，通过以下两步来进行：

1. 通过预处理的data1数据，挖掘服务与服务之间的关联规则。
2. 通过预处理的data2系列数据，挖掘时间区间与服务之间的关联规则。

## 5.3 实验结果与分析

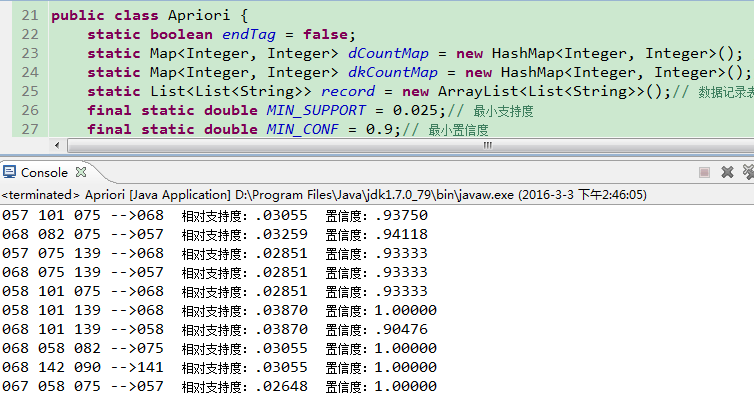
通过对Apriori算法的实现，并设置最小支持度为0.025，最小置信度为0.9之后，批量对预处理后的数据内容进行运算，可以获取多条关联规则挖掘结果。实验运行效果如图5.2所示，在挖掘结果中，包括“A🡪B”、“A B🡪C”、“A B C🡪D”等多种模式的关联规则。

图5.2 关联规则挖掘结果示例

关联规则中三个字符的Sensorid，通过数据集中的sensor.csv的映射关系翻译，即可以获取每条关联规则对应的含义。通过映射关系翻译后，发现所挖掘出来的部分规则都包含在一定的行为中，并因为用户的个人生活习惯而存在一定的关联性。而有些关联规则并没有实际的意义，但是从所挖掘的关联规则看，有些看似毫无关联的服务之间，却隐藏着用户的潜在需求。表5.1是通过处理data1的数据内容，所挖掘出来的部分关联规则。表5.2则是通过处理data2系列文件的数据内容，所挖掘出来的部分关联规则。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | 关联规则 | 支持度 | 置信度 | 意义 |
| 1 | 137 🡪 105 | 0.02851 | 1.00000 | 冰箱 🡪 灯光 |
| 2 | 126 🡪 091 | 0.02851 | 1.00000 | 冰箱功能1 🡪 冰箱功能2 |
| 3 | 088 081 🡪 068 | 0.03259 | 1.00000 | 冷水 衣橱 🡪 热水 |
| 4 | 088 142 🡪 141 | 0.02851 | 0.93333 | 冷水 洗碗机 🡪 厨房门 |
| 5 | 088 090 🡪 141 | 0.03259 | 1.00000 | 冷水 烘干机 🡪 厨房门 |
| 6 | 055 072 🡪 084 | 0.04073 | 0.90909 | 厨房柜子 柜子🡪 抽屉 |
| 7 | 137 143 070 🡪 080 | 0.02851 | 0.93333 | 冰冻室 微波炉 洗碟机🡪柜子 |
| 8 | 080 055 091 🡪 084 | 0.05092 | 0.92593 | 柜子 冰箱🡪抽屉 |
| 9 | 067 057 130 🡪 068 | 0.02851 | 1.00000 | 在卫生间中，柜子，门🡪热水 |
| 10 | 053 072 070 🡪 084 | 0.03055 | 0.93750 | 柜子 洗碟机🡪抽屉 |
| 11 | 088 068 058 082🡪075 | 0.02851 | 1.00000 | 在卫生间，冷/热水，药柜子，书房抽屉🡪用抽屉 |
| 12 | 080 055 137 091🡪084 | 0.03259 | 0.94118 | 在厨房，开柜子，用冰箱🡪用抽屉 |
| 13 | 068 067 057 101🡪058 | 0.03870 | 0.90476 | 在卫生间中，开灯，接冷水🡪药柜 |
| 14 | 088 057 058 082 075🡪068 | 0.02648 | 0.92857 | 在卫生间中，开灯，使用冷/热水，开小柜子，书房中开抽屉 |
| 15 | 088 067 057 058 101🡪068 | 0.03055 | 1.00000 | 在卫生间中，开灯，使用冷/热水，开小柜子 |

表5.1 服务-服务关联规则挖掘结果示例

从表5.1可以看出，有些关联规则体现服务调用之间的关联关系，比如1、2、4、5、7、8、10；而类似3这样的关联规则，在最终的挖掘结果中，则多达12条，全部是冷水服务和热水服务，以及一些其他服务的关联关系，比如排气扇、灯光、衣橱等；这些规则的置信度极高，说明冷水与热水被用户间歇使用的习惯就如此；比如9、15，是更为复杂的关联规则，体现用户在卫生间中进行洗漱活动所进行的一系列事件，体现的也是用户的生活习惯；对于以上部分规则，其实际意义完全一样，比如8和12；对于像6、11、14，这样的规则并没有特别符合逻辑的释义。从服务-服务关联规则的结果中可以发现，挖掘出的部分关联规则并没有严格的逻辑含义，有些关联规则则是显而易见的逻辑，而挖掘结果中仍然可以发现未知的服务调用之间的关系，比如1、2、4、5等。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | 时间区间 | 项集 | 支持度 | 意义 |
| 1 | 11：40：00-11：59：59 | 126,129,141 | 0.64 | 进入厨房使用火炉和冰箱 |
| 2 | 12：20：00-12：39：59 | 104 | 0.53 | 开走廊的灯 |
| 3 | 13：20：00-13：39：59 | 140 | 0.87 | 开/关大门 |
| 4 | 16：40：00-16：59：59 | 140,107,097 | 0.83 | 开/关门，开起居室灯 |
| 5 | 17：00：00-17：19：59 | 141,051,052 | 0.52 | 进入厨房后，使用厨房用品 |

表5.2 时间区间-服务关联规则挖掘结果示例

从时间区间与服务的挖掘结果看，可以发现一些明显的频繁项集，在一定的时间区间中，都有一些常发生且有现实意义的服务，如表5.2中结果所示。在中午和傍晚时，用户可能倾向于回家，进入厨房进行一些活动，从这些结果中也大略知道用户的上班时间和起居时间，以及一些生活模式。在挖掘结果中，也存在一些支持度高但是毫无意义的结果。

从以上的关联规则结果中，通过筛选，选取其中有实际意义但是并没有被发现的有效规则，并根据第四章中智能家居规则库规则定义，添加新的属性内容，构造完整可用的规则。例如表5.1中的第1条潜在规则“137🡪105”，通过比对sensor.csv映射文件，则是Kitchen\_Freezer🡪Kitchen\_Light。依照规则库定义5，可以获取初步规则rule:(Freezer137 Device\_doService setDevice)🡪(Light105 Device\_doService startDevice)；表5.2中的第4条潜在规则，通过映射转换后，规则为时间区间（16:40-16:59）🡪Foyer\_door,LivingRoom\_Light1,LivingRoom\_Light2，依据规则库定义6，可以转换为初步规则rule:now(?x),le(?x,16:59:59),ge(?x,16:40:00) 🡪(door140 Device\_doService startDevice),(Light107 Device\_doService startDevice),(Light097 Device\_doService startDevice)。

# 第六章 总结与展望

## 6.1 总结

物联网的发展促进了智能家居领域的应用推广，国内外也提出了各种智能家居和智能设备的方案和架构。但是随着人们对舒适、便捷和智能化生活的追求越来越高，迫切需要家居场景中的智能设备可以智能地获取和传输情景信息，并进一步实现自动控制和调节。为此，需要更智能化的控制系统，以智能设备为载体，以服务为中心，考虑用户需求和行为特点，更好地为用户服务。然而，目前市场上流通的应用，更多的专注于设备控制本身，没有结合情景信息来考虑设备和服务资源的整合，也没有充分考虑用户行为方式，为此本文从设备服务资源整合出发，基于用户行为规则和提炼习惯关联规则，提出智能家居应用场景下的基于本体的自适应控制框架，并在该框架下研究如何整合情景下的设备与服务，提出基于规则推理的自适应控制方案，并基于用户行为数据提炼规则。

在研究了国内外解决设备异构性的框架基础上，通过研究和分析国内家居情景活动特征和家居场景布局方式，设计一套智能家居场景下的自适应控制系统架构，包括物理设备层、设备中间件层、服务层、应用层和交互层，详细阐述各个层级的模块功能和交互逻辑，并提出本文的侧重研究部分——应用层，并明确应用层中主体模块和相应功能。

在应用层中为了实现现实物理设备的自适应控制，需要充分考虑现实生活情景信息。为此，通过采用七步法的本体建模方法，参考指导原则，分析国内家居情景特征和布局特点，从用户、空间、设备、服务、网关和环境等多个维度描述智能家居场景，抽象其中的关键概念和属性信息，基于现实实体构建智能家居本体模型，并通过手动添加实例的方式完善智能家居本体库。

为了实现系统的自适应控制程度，在构建的智能家居本体模型上，本文采用基于规则的推理引擎工具Jena。设计了一套自适应控制规则的定义方法，并根据家居场景添加属性规则和自定义规则。利用规则库中规则，设计并实现从用户需求到目标服务的自适应逻辑推理流程，阐述了自适应的运行机制。在本体模型中，不仅可以推理出用户的直接需求，还可以通过还基于现实基础添加自定义规则，将环境信息和设备服务融合起来，以实现基本的自适应控制能力。考虑到属性规则和自定义规则的局限性，提出通过关联规则挖掘Apriori算法来挖掘用户历史行为数据中的情景-设备-服务关联规则的思路，进一步充实规则库，提升系统的自适应控制能力。

## 6.2 展望

虽然国内外研究者都对相关课题做了很多的研究和实现工作，但是想要更好的实现市场用户的需求，还有很多的问题亟须解决。本文的研究从三个研究点提供一些思路，但是因为水平有限，研究工作中难免有不足之处，需要在以后的工作中进一步完善和细化。本文还可以从以下几点做进一步的研究和探讨：

1）本文仅从本体模型构建上考虑，但是在真实的应用场景中，设备数量多，对情景环境的实时监测数据量巨大，如何更好的构建本体模型和使用本体库，使整个系统可以良好运行，及时响应用户请求，还需深入研究和实验。

2）在基于用户历史控制数据的设备-服务关联规则挖掘中，仅仅采用经典的挖掘算法来进行实验，但是因为数据有限，挖掘结果并不是很丰富，所以可考虑采集更全面的数据，并使用或者设计更有效且高效的算法来挖掘潜在规则。

3）现有框架和系统仅仅针对单个用户，那么考虑多用户下的智能家居自适应控制系统处理，如何考虑多用户协同控制设备-服务，并在容错性良好的情况下，响应各用户请求，在日后的系统应用中，值得深究。

4）整个系统的设计以及研究内容的设计与实现，都是从需求和功能上出发，并没有充分考虑其中可能存在的各种安全问题，所以，为了提升系统可靠性，用户数据传输的安全性也是值得继续研究的内容。

# 

# 参考文献

1. Xu Heyuan. A Study on Ubiquitous Network and Internet of Things. ZTE COMMUNICATIONS,2010,16(Z1),pp.13-16
2. Suryadevara N K, Mukhopadhyay S C. Smart Home Related Research[M]//Smart Homes. Springer International Publishing, 2015: 11-51.
3. Joseph B, Jonathan T, Anne H. Flexible IoT Middleware for Integration of Things and Applications. Web Services, 2014. IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2014: 968-975.
4. Perera C, Zaslavsky A, Christen P, et al. Context-aware sensor search, selection and ranking model for internet of things middleware. Mobile Data Management (MDM), 2013 IEEE 14th International Conference on. IEEE, 2013, 1: 314-322.
5. SUN J Z, SAUVOLA J. Towards a conceptual model for context-aware adaptive services, parallel and distributed computing, applications and technologies,2003. PDCA. Proceedings of the Fourth InternationalConference,27-29，Aug.1-3,2003.
6. Strang T, Linnhoff-Popien C. A context modeling survey[C]//Workshop Proceedings. 2004.
7. Llinas J, Waltz E. Multi-sensor Data Fusion. Norwood, MA: Artech House,1990.
8. D L.Hall, J.Linas. Handbook of Multi-sensor Data Fusion. Boca Raton, FL, USA
9. Smirnov A V, Kashevnik A M, Ponomarev A V, et al. Ontology-based organization of interactions between services in the smart space for hybrid system control[J]. Scientific and Technical Information Processing, 2015, 42(5): 367-374.
10. Coen M, Phillips B, Warshawsky N, et al. Meeting the computational needs of intelligent environments: The metaglue system[C]//Proceedings of MANSE’99. 1999: 201-212.
11. Brumitt B, Meyers B, Krumm J, et al. Easyliving: Technologies for intelligent environments[C]//Handheld and ubiquitous computing. Springer Berlin Heidelberg, 2000: 12-29.
12. Bonino D, Corno F. Dogont-ontology modeling for intelligent domotic environments[M]. Springer Berlin Heidelberg, 2008.
13. Wang X, Zhang D, Dong J, Chin C and Hettiarachchi, S.R. Semantic Space: An Infrastructure for Smart Space. IEEE Pervasive Computing, Vol.3, No.2,2004.
14. 聂风华, 钟晓流, 宋述强. 智慧教室: 概念特征, 系统模型与建设案例[J]. 现代教育技术, 2013, 7: 5-8.
15. kishore Ramakrishnan A, Preuveneers D, Berbers Y. Enabling self-learning in dynamic and open IoT environments. Procedia Computer Science, 2014, 32: 207-214.
16. Adomavicius G, Tuzhilin A. Context-aware recommender systems[M]//Recommender systems handbook. Springer US, 2011: 217-253.
17. Srikant R, Agrawal R. Mining generalized association rules[M]. IBM Research Division, 1995.
18. Mobasher B, Cooley R, Srivastava J. Automatic personalization based on web usage mining[J]. Communications of the ACM, 2000, 43(8): 142-151.
19. Chen Z. Mining individual behavior pattern based on significant locations and spatial trajectories[C]//Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on. IEEE, 2012: 540-541.
20. Berkane M L, Seinturier L, Boufaida M. Using variability modelling and design patterns for self-adaptive system engineering: application to smart-home[J]. International Journal of Web Engineering and Technology, 2015, 10(1): 65-93.
21. Giaretta P, Guarino N. Ontologies and knowledge bases towards a terminological clarification[J]. Towards very large knowledge bases: knowledge building & knowledge sharing, 1995, 25: 32.
22. Gómez-Pérez A, Benjamins R. Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods[C]. IJCAI and the Scandinavian AI Societies. CEUR Workshop Proceedings, 1999.
23. 刘红阁, 郑丽萍, 张少方. 本体论的研究和应用现状[J]. 信息技术快报, 2005, 3 (1): 1-12.
24. Miller E. An introduction to the resource description framework[J]. Bulletin of the American Society for Information Science and Technology, 1998, 25(1): 15-19.
25. Brickley D, Guha R V. {RDF vocabulary description language 1.0: RDF schema}[J]. 2004.
26. McGuinness D L, Van Harmelen F. OWL web ontology language overview[J]. W3C recommendation, 2004, 10(10): 2004.
27. Bechhofer S. OWL: Web ontology language[M]//Encyclopedia of Database Systems. Springer US, 2009: 2008-2009.
28. Erl T. Service-oriented architecture: a field guide to integrating XML and web services[M]. Prentice Hall PTR, 2004.
29. Fielding R T, Taylor R N. Principled design of the modern Web architecture[J]. ACM Transactions on Internet Technology (TOIT), 2002, 2(2): 115-150.
30. Guinard D. A web of things application architecture-Integrating the real-world into the web[D]. ETH Zurich, 2011.
31. Rule-based expert systems[M]. Reading, MA: Addison-Wesley, 1984.
32. McBride B, Boothby D, Dollin C. An Introduction to RDF and the Jena RDF API[J]. Retrieved August, 2004, 1: 2007.
33. Dickinson I. Jena ontology api[J]. On the WWW, at http://jena. sourceforge. net/ontology/index. html, 2009.
34. Reynolds D. Jena 2 inference support[J]. Online manual at http://jena. sourceforge. net/inference/index. html, 2004.
35. Carroll J J, Dickinson I, Dollin C, et al. Jena: implementing the semantic web recommendations[C]//Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters. ACM, 2004: 74-83.
36. Han J, Kamber M, Pei J. Data mining: concepts and techniques[M]. Elsevier, 2011.
37. Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases[J]. ACM SIGMOD Record, 1993, 22(2): 207-216.
38. Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[C]//ACM Sigmod Record. ACM, 2000, 29(2): 1-12.
39. <http://www.chinanews.com/estate/2010/12-29/2755873.shtml>
40. <http://www.mohurd.gov.cn/gsgg/gg/jsbgg/201110/t20111026_206878.html>
41. 龚 军.燃气自动阻断器:中国,CN200520094926.8[P].2006-3-29.
42. 刘燕玲.物联网感知信息模型与处理的研究[D].复旦大学,2015.
43. 梁佳.基于M2M的物联网中间件的设计与实现[D].复旦大学,2015.
44. 王怡,鄂旭.基于物联网无线传感的智能家居研究[J].计算机技术与发展,2015,(2):234-237.DOI:10.3969/j.issn.1673-629X.2015.02.053
45. 张囡囡. 面向语义网的领域本体半自动构建方法的研究[D]. 大连：大连海事大学，2008
46. 李勇,张志刚.领域本体构建方法研究[J].计算机工程与科学,2008,30(5):129-131.DOI:10.3969/j.issn.1007-130X.2008.05.039.
47. Gruber TR. Towards Principles for the Design of Ontologies Used for Knowledge Sharing[J]. International Journal of Human Computer Studies, 1995, 43: 907-928.
48. Daoud M, Lynda T, Mohand B. A Personalized Graph-Based Document Ranking Model Using a Semantic User Profile[J]. User Modeling, Adaptation, and Personalization, Springer Berlin Heidelberg, 2010: 171-182.
49. Klyne G, Carroll J J. Resource description framework (RDF): Concepts and abstract syntax[J]. 2006.
50. Van Emden M H, Kowalski R A. The semantics of predicate logic as a programming language[J]. Journal of the ACM (JACM), 1976, 23(4): 733-742.
51. (美)韩家炜, (美)坎伯(Kamber, M. ),等. 数据挖掘 : 概念与技术, 第3版[M]. 机械工业出版社, 2012.
52. E. Munguia Tapia. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. Pervasive 2004,Vienna, Austria.
53. E. Munguia Tapia. Activity Recognition in the Home Setting Using Simple and Ubiquitous Sensors. S.M. Thesis, Massachusetts Institute of Technology, 2003.

# 致谢

从本科阶段到研究生阶段，这已经是我在复旦大学的第七个年头。随着毕业论文的完成，研究生阶段的生活也要接近尾声，在此，感谢这七年以来，帮助过我的老师和同学！

感谢我的研究生导师李敏波老师。在研究生期间，您在研究能力和工程能力上对我的耐心培养，让我在研究生期间成长不少，也将会让我受益终生；而且您在平时工作中表现的强大的责任心和精益求精的做事态度，让我深受感染，非常值得我学习。

感谢实验室的同学们，在研究生期间的学习和探讨，也感谢在毕业论文编写过程中提供的启发和思路。同时，特别感谢刘燕玲、梁佳等同学，正因为大家向着同一目标努力，才让工作量繁杂的研究课题能够有条不紊的进行下去，感谢你们在物理设备层和设备中间件层的工作，以及为我提供的规范化服务，和开拓性的开发建议。

感谢我的家人，多年来的支持和鼓励，无私的付出和关爱，让我无后顾之忧，专心学业，自由地做我感兴趣的事情。

感谢我的男朋友，多年以来的陪伴，让我们互帮互助，共同进步。

感谢复旦大学软件学院，为这七年来的学习和生活所提供的良好条件。

最后，衷心地感谢在百忙之中评阅论文和参加答辩的各位专家、老师！

曾君

2016年3月31日

**复旦大学**

**学位论文独创性声明**

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

作者签名： 日期：

**复旦大学**

**学位论文使用授权声明**

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

作者签名： 导师签名： 日期：