

本科毕业论文



论文题目：基于语音识别和面向对象的智能家居系统设计和实现

院 系：软件学院

专 业：软件工程

姓 名：许凯其 学 号：14302010032

指导教师：李敏波 职 称：副教授

单 位：复旦大学

日 期：2017年5月13日

**目录**

[第一章 绪论 5](#_Toc514682109)

[1.1 研究背景和意义 5](#_Toc514682110)

[1.2 国内外研究现状 5](#_Toc514682111)

[1.3 研究目标和内容 6](#_Toc514682112)

[1.4 章节安排 6](#_Toc514682113)

[第二章 相关技术研究综述 8](#_Toc514682114)

[2.1 关键技术概述 8](#_Toc514682115)

[2.2 语音识别技术 8](#_Toc514682116)

[2.3 中文分词技术 9](#_Toc514682117)

[2.4 面向对象技术 10](#_Toc514682118)

[第三章 智能家居的整体需求分析和架构 12](#_Toc514682119)

[3.1 家居环境特征 12](#_Toc514682120)

[3.2 典型的控制应用场景 13](#_Toc514682121)

[3.3 系统的需求分析和相关概念的抽象 14](#_Toc514682122)

[3.3.1 需求分析 14](#_Toc514682123)

[3.3.2 相关概念的抽象和面向对象定义 16](#_Toc514682124)

[3.4 系统整体的流程和架构 18](#_Toc514682125)

[3.4.1 系统整体流程 18](#_Toc514682126)

[3.4.2 系统架构 19](#_Toc514682127)

[第四章 智能家居系统的具体实现和测试 21](#_Toc514682128)

[4.1 语音识别应用的实现 21](#_Toc514682129)

[4.2 规则引擎的实现 22](#_Toc514682130)

[4.3 基本抽象概念的关系及定义 24](#_Toc514682131)

[4.4测试 25](#_Toc514682132)

[4.4.1 基于规则和模式的测试 26](#_Toc514682133)

[4.4.2 主动请求的测试 28](#_Toc514682134)

[第五章 总结与展望 30](#_Toc514682135)

[5.1 总结 30](#_Toc514682136)

[5.2 展望 30](#_Toc514682137)

[参考文献 32](#_Toc514682138)

[致谢 33](#_Toc514682139)

**摘要**

人工智能和计算机技术的发展促进了智能家居的应用和推广，国内外也提出了相关的智能家居解决方案和系统架构，但是缺少在软件层的整体解决方案和设计。为了满足多变的需求和复杂的环境，填补智能家居在软件层研究的缺失，本文基于家居环境的抽象以及百姓日常生活习惯的提炼，结合人工智能中的语音识别技术，使用面向对象技术对家居系统进行抽象，提出了一套智能家居控制架构和智能家居系统流程，并通过编程语言进行系统原型的实现，最终使用相关数据进行测试来检验。

**关键词：**智能家居，语音识别，面向对象，系统原型

**Abstract**

The development of artificial intelligence and computer technology promotes the application and popularization of smart home. It also puts forward relevant solutions and system architecture of intelligent home home and abroad, but it lacks the overall solution and design in the software layer. In order to meet the changeable needs and complex environment and fill in the lack of research in the software layer of intelligent home, this paper, based on the abstraction of home environment and the extraction of common people's daily living habits, combined with the speech recognition technology in artificial intelligence, abstracts the home system by object oriented technology, and puts forward a set of smart home. The control architecture and the smart home system process are implemented, and the prototype is implemented through programming language. Finally, the relevant data are used for testing.

**Key words:** smart home, speech recognition, object oriented, system prototype

# 第一章 绪论

## 1.1 研究背景和意义

随着人工智能、计算机技术的迅猛发展，基于人工智能的应用已经掀起不小的浪潮。结合硬件的不断革新，很多跨时代的产品也因此而生。作为物联网中重要的一环，家居系统也随着时代在不断的发展。经历了两芯线和同轴线完成家庭组网以实现少量的家居控制、到总线和IP技术组网来完成对讲和安防业务，再到集中化的控制系统，最终到当今基于物联网技术实现定制化需求[1]，家居系统正在一步步走向成熟。如今，结合人工智能，已经有了部分“智能化”的家居解决方案的实现，或是具体解决案例或是整体系统设计。但是，百姓生活条件的提高对家居系统提出了更高的要求，安全、舒适、省电、自动、智能等属性缺一不可，这就要求家居系统做出更大的突破来满足百姓的实际生活需求。

根据国家重大专项《无线泛在网络架构研究和总体设计》[2]的研究表明，目前物联网的智能化还在处于初步阶段，家居系统缺少对复杂和多变的环境的灵活应对。于此同时，不同厂商的接口和数据交互以及通信协议的不一致导致不同的家居系统有着巨大的鸿沟，使得行业间存在着技术壁垒，这也影响了家居系统的进一步发展。

所以，如何利用现有的硬件和人工智能以及互联网技术，提出一种可行的智能家居设计方案，满足用户的多变的请求，适应复杂多变的环境，具有一定的意义，也能够为智能家居的发展做出一定的贡献。

## 1.2 国内外研究现状

当前的智能家居的开发更多的是在嵌入式开发或者硬件中的实现，在软件层的研究中也比较关注实际具体场景的应用和设计，很少涉及到具有整体性、普适性的架构设计，可以说在研究上有一定的空白。此外，在语音识别这一方面目前的实现方法还停留在硬件上，大部分以语音识别芯片为主，很少考虑到软件层上的语音识别，用到的语音识别的模型也比较陈旧。

搜索相关学术论文门户网站可以发现，基于面向对象的智能家居的建模、设计和实现极少，可以说在这一方面存着比较大的空白。

针对智能家居的语音识别和面向对象设计，虽有国内外研究者有所涉及，但是更多的是在硬件的层面上，缺少对软件层面的研究，也缺少与软件层的结合，

而软件的环境适应性、跨平台、易维护、易实施等特性，可以说软件层面的深入

研究是未来智能家居的研究方向之一。本文在上述的研究成果上，通过对家居场景的建模和抽象，通过软件设计中的面向对象的方法，结合人工智能中的语音识别，让家居系统具有普适性，能适应多样的环境，满足多样的需求，也为后续的软件层的研究提供一定的思路和借鉴。

## 1.3 研究目标和内容

智能家居未来的大规模应用可预见性，带来了巨大的机遇和挑战。如何从单一的控制到适应多变的需求和复杂的环境，设计出切合用户实际的需求的系统，是至关重要的问题。本文在对国内外家居实际应用场景和相关技术的充分调研基础上，结合用户实际的需求和具体生活场景的抽象，通过面向对象技术来对家居场景建模，设计出普适性的智能家居系统原型并通过相应代码实现，并整合语音识别这一前沿技术来为用户提供便利。

为实现以上需求和系统设计，本文的具体研究目标和内容如下：

一、提出智能家居的整体架构和系统流程

通过分析生活中常见的家居场景以及家居的部署，以及对百姓日常生活习惯的详细调研，对整体需求进行详细分析后，提出一套贴切、符合生活习惯的智能家居系统架构和流程，并对各个层级进行划分。

二、设计并实现基于面向对象的家居系统模型

利用面向对象的方法，对当前的常见家居场景进行充分的调研，并对经典家居场景进行抽象、建模，并且通过代码进行系统原型的实现。

三、实现语音识别功能

通过比较分析主流的软件层面的语音识别API或者接口，综合开发难度、性能，选出适合家居系统的API或接口，通过其提供的说明文档，将其接入智能家居系统原型，使系统原型具备语音识别的功能。

## 1.4 章节安排

本文通过对家居实际应用场景的调研，提出一套具有普适性的家居系统原型，并通过面向对象的方法对其做了软件层的实现，同时整合了人工智能中的语音识别，为该系统做了智能化提供一定的思路。本文的文章结构如下：

第一章：绪论。介绍本文的研究背景和意义，阐述国内外的研究现状，并且提出研究内容和目标，介绍本文的章节结构。

第二章：相关技术详细阐述。对研究应用到的面向对象技术、语音识别技术、

以及语音识别处理后的所需中文分词处理技术进行调研，并对面向对象技术、语

音识别技术、中文分词技术进行详细的阐述。

第三章：智能家居的需求分析和整体流程架构。研究当前常见家居场景和百姓的生活习惯，对其进行提炼和需求分析，对其活动特征进行抽象和整合，提出智能家居系统的整体架构，并且对架构中各个部分进行功能和关系说明。

第四章：智能家居系统的具体实现和测试。进一步分析家居场景和实际需求，对架构中的各部分进行实现和详细说明，并通过多组常见的家居环境的模拟数据，对系统原型进行测试。

第五章：总结与展望。对本研究中所做工作的总结，并对智能家居的未来做出展望。

# 第二章 相关技术研究综述

## 2.1 关键技术概述

本文主要研究家居系统在软件层的抽象和模型构建以及相应的实现，并且通过所设计的模型解决多变的环境和复杂的请求等需求，最终实现智能化。在研究中用到的技术主要有三部分，语音识别技术、中文分词处理技术、面向对象技术。

在实现智能化的方面，本文用到了语音识别技术，通过识别用户的语音，将用户的语音转换成相应的文字，再通过中文分词处理技术进行所得汉字切分，转换成系统可以理解的需求，并作出相应的操作。

在实现普适化的方面，本文用到了面向对象技术，通过面向对象技术抽象具体事务的能力以及其在软件层的大规模应用，对家居系统进行软件层的需求分析，建模，最终通过代码的方式实现，并用相关的可视的实验数据进行检测。

## 2.2 语音识别技术

语音识别技术指将语言词汇内容转化成计算机可以理解的文本或者二进制编码。

20世纪90年代后，随着处理器、存储器硬件水平的提高，语音识别从实验室走了出来，进入到大众日常生活，并随着人工智能的发展不断突破，在生活中应用越来越广[3]，在语音拨号、语音导航、语音检索等领域给人们生活带来了极大的便利[4]。该项技术有如下适合本课题的特性：

一、用户的便利性：

传统的软件层或者App层面上的设计都是基于用户输入或者点击，通过判断用户的输入，结合预测模型来判断用户的意图。语音识别则跳过了这一步，由用户只需要对着话筒说话即可，由计算机来完成语音到用户输入的转化。对于行动不便或者不习惯使用移动设备的老人、残疾人，基于语音识别技术的系统能为这些人群提供更加接近自然的交流方式和更为友好的用户界面[5]。

二、识别的准确性：

近年的神经网络的发展给人工智能带来了新的活力，也给语音识别带来了新的突破，目前的各大厂商所宣布的语音识别准确性均在90%以上，复杂的神经网络模型使得语音识别功能的更加强大，发音不够标准不能识别、声音模糊等历史遗留的问题也得到了解决。

三、开发的方便性：

当前语音识别不再严重依赖硬件芯片，由于软件开发框架和web技术的发

展，很多厂商或是提供了API服务，或是提供了相应的接口，来帮助开发者接入语音识别的功能，可以说语音识别的开发不再和硬件芯片时代一样具有很大的行业门槛，只需要会一种主流的语言，即可完成系统的语音识别接入。

目前语音识别技术接入主要分两种。第一种是在线API的请求，通过申请相应的key和license，将输入的语音打包成一定的数据格式（例如json）对厂商进行http请求，厂商返回相应的结果。第二种是通过离线的包（例如jar包），调用包中的接口以及包中所自带的模型，将输入的语音通过接口转化成厂商规定的格式，再调用厂商的接口，来实现对语音的识别。

两种请求各有优缺点，第一种在线API的请求需要通畅的网络，一旦网络无法畅通或者网络有阻塞，可能会出现识别不准确或者无法识别的情况，但是第一种的开发时间成本比较低，厂商通常有相应的demo的实现，将其进行封装后，替换成开发者的key和license即可，并且在线API的请求中，厂商的模型是实时保持更新的，如果网络通畅，准确性高于第二种。第二种开发方式由于是离线的包，所以不需要相应的网络，可以说对环境的依赖不是这么的大，可以在极端条件下运行，但是离线的包也限制了开发者，由于当前语音识别设计到深度神经网络，通常一个准确性较高的模型需要比较大的空间，同时由于离线模型不具有实时同步更新性，所以准确度和新的词汇适应性不如第一种。

考虑到家居系统中网络的普遍和当前Wi-Fi技术的流行，可以说家居系统的网络通畅性有一定的保障，再综合软件的轻便性，所以在技术选型上本文选择了一种方式，通过在线API的请求来实现语音的识别。

## 2.3 中文分词技术

本文主要用到了自然语言处理技术中的中文分词技术，中文分词技术指的是将汉字序列切分成一个个单独的词。

目前日常生活中，中文分词技术应用广泛，汉字的智能输入、中外文对译、中文校对、汉字简繁体转换、自动分类、关键词检索等方面都有涉及[6]。中文分词技术有以下几个适合本课题的特性：

一、消除歧义

通过中文分词技术能消除输入文本中的歧义使得计算机能够更好理解的用户的输入，例如“开窗帘”这一条输入请求，如果简单的按照关键词匹配，可能计算机会匹配到“开窗”这一动作，通过中文分词，将这三个字切分成“开”、

“窗帘”这两部分，再结合相应的预测模型，就能消除歧义，避免系统的错误操作。

二、关键词匹配

通过中文分词后，将一段完整的文本切分成几部分，能够匹配相应的关键词，如温度、湿度，对其中的关键词进行匹配，做出相应的操作。对比直接匹配关键词，切分后的关键词匹配准确性更高。

三、分词的准确性

目前中文分词已经初具规模，而且本文中的家居常用语音输入并不涉及到专业词汇，均为生活用语，可以说经过大规模文本训练的中文分词模型具有比较可信的准确性。

四、开发的方便性

目前各大中文分词的开发者或者厂商均提供了相应的离线的包或者API接口来供开发者实现，并且提供了多种多样的语言选择，结合详细的开发文档，实际开发的门槛并不高。

目前中文分词技术和语音识别技术一样，也是两种接入方式，第一种是API接口，第二种是通过离线的包接入。两种方法的优缺点和语音识别技术两种方法类似，有区别的是中文分词离线的包通常在需求不是很高的情况下比较小，只需要一个比较小的词典和一个预训练好的模型即可，本文中的家居常用文本输入的内容十分常见，均为生活中常用词，不存在生僻词汇和专业术语，对词典的要求比较低，对模型的复杂性要求也比较低，加上国内目前在线API的中文分词技术商用性不高，大部分处于初步发展的阶段，综合考虑后选择了导入的离线的包，调用包中的接口，结合对应的简化版词典来进行实现。

## 2.4 面向对象技术

面向对象（Object Oriented）是对现实世界抽象和理解的方法，是一种具有普适性的软件开发方法。面向对象编程实现了软件系统的三个目标：重用性、灵活性。在面向对象分析的过程中，强调了领域内发现和描述对象或者概念，面向对象设计中强调了软件对象和它们如何相互协作来实现需求。结合本课题研究方向，面向对象有如下优点：

1. 面向对象技术的特性

抽象，封装，继承，多态是面向技术的四大特性，对于家居系统，研究过程中需要对家居系统进行抽象，例如冰箱这一个对象，不需要考虑冰箱买入的时间等无关因素，而应该关注冰箱当前的温度，当前的状态（开或者关）这些关键的属性，抽象能够使研究者更加关注事物与当前系统有关的部分，抛弃无关部分。

封装这一特性对家居系统设计也有比较重要的意义。封装将对象的数据保护起来，对象只能通过受保护的接口访问其他对象，避免了数据泄露，保证了数据的安全性。通过家居对象的封装，对应了相应的数据访问接口，防止了数据的泄露，也给后续开发实践者规范了后续开发的流程。通过继承这一特性，家居系统可以定义一些基础类（下一章详细阐述），并且由子类进行继承，并加入子类特有的方法，使得整体的结构更加清晰，可重用，也可以为后续的开发提供了借鉴。

二、面向对象的语言多样性

当前市面上有很多成熟的面向对象语言，如Java、C++、C#，对于家居系统软件层的语言实现，可以说有很多的选择。

三、面向对象的广泛性和可扩展性

当前面向对象的概念和应用已经超越了单纯的软件开发和程序设计，诸如数据库设计、分布式系统、人工智能、应用平台等方面都有面向对象的实现和应用，用面向对象设计家居系统，使得家居系统原型沿着面向对象的思想可以添加更多的元素，比如分布式系统等前沿技术。

四、家居系统本身的特性所决定

家居系统有比较多的文字表述说明需求，需要处理比较多的人机交互操作，很少涉及到复杂的数学计算和算法，而面向对象特点正是擅长抽象这些需求，将语言逻辑转换成可执行的程序。

当前比较典型的面向对象语言有Java、C++、Smalltalk、C#、EIFFEL等，Java以其平台无关性、详细的内置类库、安全性与健壮性、诸多的成熟框架支持等优点占据了使用率第一，并且考虑到后续研究会拓展到Android上，以及需要对Web应用开发的支持，所以本次开发所选用的编程语言为Java。

# 第三章 智能家居的整体需求分析和架构

调研国内外调研家居系统设计后，本章结合常见家居、生活场景和家居的布局的研究，对几种典型的家居场景进行深入分析，并且提炼实际需求，结合面向对象的思想，整合语音识别，提出一套可行的智能家居整体设计架构，并对系统的各个模块功能进行划分。

## 3.1 家居环境特征

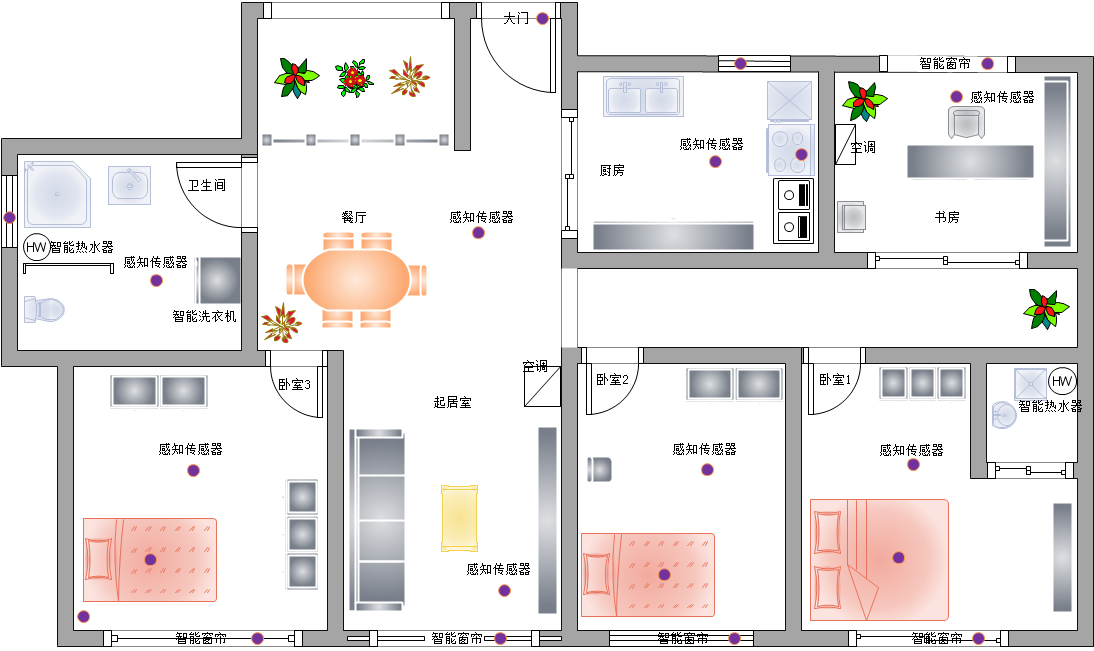
自改革开放以来，我国的家居环境有了极大的改善，根据国家统计局在2017年7月6日的公布数据显示[7]，2016年全国城镇居民人均住房建筑面积为36.6平方米，相较于改革开放初期的6.7平方米，实现了近546%的跨越式发展，并且住房环境和住房的质量也有着很大的提升。

除了住房的面积的变化，住房的功能也随着经济提升和生活条件改变也产生了一定的变化。改革开放初期，住房基本以卧室为主，很少有其他功能的房间，居民的卫生间和洗漱间以共享的方式为主。随着改革开放的进行，经济条件的提升和生活条件的改变带来了新的需求，娱乐、休闲、学习、工作、盥洗等需求也对住房结构产生了巨大的影响。书房、厨房、独立卫浴的卫生间，甚至家庭影院等新时代的功能的房间也随之产生。根据我国于2012年生效的国家标准《住宅设计规范》显示[8]，住宅应按照套型设计，每套住宅应设卧室、起居室（厅）、厨房和卫生间等基本功能空间，并对各种套型的面积进行了严格的规定。此外，《住宅设计规范》还对住宅设计提出了以人为本和节能的需求，这对家居系统的设计也提出了一定的要求。

在基本家居场景环境中，不同的房间有着不同的功能，卧室主要负责睡眠和休息的功能，厨房负责烹饪的功能，卫生间负责盥洗的功能，书房负责学习的功能，客厅负责接待客人和日常休闲的功能，可以说各个房间的功能包含了居民生活日常的方方面面。

传感器的技术迅猛发展给家居系统带了新的需求和机遇，通过传感器的控制和信号传递，家居系统的中的各个房间不在独立，可以相互协作来完成用户多方面的需求和适应复杂多变的环境。比如厨房中的烟雾传感器，通过烟雾传感器可以对厨房中的烟雾进行监控，并且在烟雾超标且用户不在时，对用户的其他的房间的家居进行关闭或者安全模式的操作，并通知用户，使整个家居系统处于警报状态。再如湿度传感器，可以对房间的湿度进行监控，并且通过空调和窗户协调

工作来调节房间的湿度。一个典型的家居环境，如果配上相应的传感器，加上相应的网关，可以使得家居环境更加整体，并且各个家居不在独立，能协调进行功能以自我适应环境和响应不同用户的请求。

如图3.1所示，该家居系统中有三个卧室，一个厨房，一个书房，一个餐图3.1 典型的智能家居系统平面图[9]

厅，一个起居室，一个卫生间，每个房间配有感知传感器。在该系统中，家居具有远程控制功能，比如用户可以远程控制智能热水器，进行打开或调节温度的操作，同时也可以通过感知传感器来获取房间当前的温度等状态。

## 3.2 典型的控制应用场景

在上述3.1的家居场景中，智能家居场景主要包括房间中的控制对象和感知对象（比如传感器），现在假设家居场景中有如下房间：一个卧室、一个卫生间、一个书房、一个厨房、一个客厅，这些房间包含感知设备，如温度传感器、光照传感器、湿度传感器、烟雾传感器等，基于上述环境的感知设备，并结合实际日常场景中的概括提炼，现提出如下三种典型的场景。

基于用户主动请求的场景：正值大寒，用户告知家居系统1小时以后下班，打开客厅的空调，设置成热风模式并且温度设置为28℃，并打开热水器。家居系统受到用户的请求后，对卫生间的热水器打开加热模式。用户1小时到家后，客厅空调已经提前打开，并且客厅温度舒适，用户也可以第一时间洗澡。

基于被动触发的场景：用户离家10分钟后，系统检测到家中无人状态，切换到安全模式，此时厨房内的烟雾传感器检测到烟雾超标，系统会对用户进行提醒，并且进行报警的操作，同时关闭所有的家居，打开窗户进行通风，并且拨打物业的电话进行危险的排查。

基于数据挖掘后历史行为的场景：系统通过对用户日常行为的挖掘发现，周一到周五用户通常7点半起床，8点上班，5点下班，9点上床，10点睡觉，双休日用户喜欢午休，午休时间为1-5点，那么卧室的灯周一到周五会在7点半打开，8点自动关闭，9点打开，10点自动关闭，而卧室的窗帘则在周末1-5点处于关闭状态。（由于缺少大量的历史模拟数据，本文中家居系统原型并未对此种场景进行实现）

## 3.3 系统的需求分析和相关概念的抽象

### 3.3.1 需求分析

结合3.2的实际场景，并考虑实际应用场景，基于语音识别的家居系统原型应该满足如下需求：

功能性需求：

一、用户主动请求的需求

用户通过语音输入，系统对用户的语音进行模式或者动作的匹配，获悉用户

的意图，并对具体房间的具体家居进行具体的操作，结合中文的实际语境含义进行提炼，有以下两种实际情况：

用户告知房间，告知家居名称，并告知相应的具体操作：

对指定房间中相应的家居直接执行指定的操作即可，并做相应条件异常性检测。比如用户请求系统“打开卫生间热水器，设置温度为50℃”，家居系统应直接对卫生间热水器进行打开操作，并且设置温度为50℃。

用户未告知房间，告知家居，告知相应具体操作：

对用户的位置进行检测，如果用户所在房间中有该家居，则对该家居执行相应的操作，比如用户所在地点为卧室，用户请求系统“打开空调”，家居系统检测到用户所在地位卧室，并且卧室中有空调，此时应对卧室的空调进行打开操作。如果用户的所在位置无该家居，并且家居系统中该家居唯一，系统应对该家居进行操作。比如家居系统中只有厨房有一个微波炉，用户在卧室请求系统“微波炉打开，加热，温度设置为200℃”，此时系统得知用户所在位置为卧室，卧室中没有微波炉，并且系统中只有一个微波炉，位置在厨房，用户的意图明显为对厨房

的微波炉进行操作，系统对厨房的微波炉进行打开，并且做加热的动作，温度设置为200℃。这里检测到相应的家居后，做具体操作前，需要对家居的状态进行检测，如果家居的状态未打开，需要对家居进行打开的操作在进行后续操作。

二、被动触发场景的检测，匹配相应的规则或模式

通过系统自带或者用户自己设计的规则和模式，系统通过规则和模式来控制相应家居的状态。比如用户设置了室内温度大于28℃，开启的所在房间的空调，并设置温度为24℃，空调设置为冷风的模式，系统会对用户所在的房间进行时刻监听，一旦温度超过用户所设置温度，如果房间中有空调，立刻开启空调并对空调的温度进行相应的设置。

三、相应优先级的需求

对于用户的主动请求的需求，优先高于被动触发场景，比如用户设置下午1-5点打开窗帘的规则，但是某天用户下午休息在家，用户主动请求关闭窗帘，此时系统应该正确判断相应需求的优先性，做出关闭窗帘而不是打开窗帘的操作。

但是系统中，存在着一种情况处于最高优先级，就是安全模式，一旦安全模式被触发，其他的操作都应无效，直到安全模式解除。例如某一天发生了火灾，烟雾传感器检测到房间内的烟雾高于安全值，触发安全模式，此时系统应对家居做出关闭状态，通知用户，而不是房间温度很高开启了空调，导致空调着火加剧了火灾。总结就是系统中应有以下优先级：安全模式>主动请求>被动规则或模式匹配。

四、规则和模式的自定义

用户可以对系统中的规则的参数进行自定义，也可以对模式进行各个参数的

设置，来满足定制化的需求，比如小王家和小李家的空调开启温度不相同，小王家为28℃，小李家为30℃，这些家居的规则参数用户应有自定义的能力，来满足不同用户的多样化需求，而不应是固定死板。此外，除了参数的可以自定义，规则和模式也需要支持自定义，即针对同一家居，不同的客户也有不同的需求，比如小王家要求夏天空调家中有人就时刻开启，小李家则根据温度开启，系统需要满足多变的需求，对多变的规则能做出正确的操作。

五、家居系统实物自定义

用户可以通过配置文件对自己的家居系统的房间、家居、传感对象进行自定义，系统通过配置文件来获知用户定义的房间、家居、传感对象来对家居系统做初始化操作。

非功能性需求：

一、系统的普适性

家居系统原型应支持多个家居系统，而不是单独一个家居系统，需要具有扩展性。

二、查询和管理

家居系统原型应支持查询和管理或者能够具有支持查询和管理的设计支持，比如对于同一小区同一居民楼的家居系统，它们的编号应具有一定的相似度或者一定的标识，对于家居系统的电器应该有编号来方便查询。

三、可移植性

家居系统原型应支持多平台迁移，相关的技术能支持多种语言实现。

四、可交互性

后续的应用需要能够和用户及开发者以及管理人员进行友好的交互。

### 3.3.2 相关概念的抽象和面向对象定义

#### 3.3.2.1 家居

家居是房间的组成部分，家居的状态和属性设置影响了家居环境，也决定了用户的生活体验。

在家居系统中的家居受用户控制，也受系统控制，可以将家居看成控制对象。无论是空调、冰箱、窗帘，还是热水器这些家居都有共有特性，就是它们都有状态（开或者关），并且由于是智能家居系统，需要适应用户基于时间的请求，系统的家居应该都有内置硬件计时器的支持，即家居中自带显示当前时间。考虑到，系统中需求优先级的判断，如安全模式远大于其他模式或者命令，所以家居中当前控制信号或者当前控制指令来进行标识，防止误控制。最后，考虑后续的查询和管理，家居应该有唯一的标识来区分。

所以，通过面向对象的方法，传感对象类应包括下列属性：状态、信号级别、时间、编号。

#### 3.3.2.2 传感对象

传感对象是房间的组成部分，能够直接获知房间的状态，如温度、湿度、烟雾浓度，规则或者模式的触发第一步都要检测传感对象所获取到的房间状态，可以说传感对象提供了获取房间状态的入口。

在家居系统中的传感对象大多为各种传感器，比如烟雾传感器、温度传感器、湿度传感器，这些传感对象都有状态这个共有特性，因此可以抽象传感对象基础类，基础类中有状态（state），而其他子类（比如烟雾传感器）可以通过继承父类，并且添加相应的属性来进行扩展。另外，和控制对象相同，考虑到后续的管

理和控制，每个传感器应有唯一的标识。

因此，控制对象应包括这些属性：状态、编号。

#### 3.3.2.3 房间

在家居系统中，房间是家居系统的子单元，不同房间承担着不同的功能，同时房间也可看成容器，房间中有控制对象和传感对象，控制对象影响着房间的各项指标，而传感对象监控这房间的各项指标。

不同房间虽有不同的功能，但是也有相同的特性，每个房间都有自己的名称，为了支持用户自定义的功能，房间名称不局限于卧室等常见词汇，可以用户进行自己定义。此外，房间的基本属性，面积，编号这些基本共有属性也不可或缺。另外，为了表示房间是容器，包含传感对象和控制对象，房间类中应有控制对象列表和传感对象列表。

因此，房间包含以下属性：控制对象列表、传感对象列表。

#### 3.3.2.4 用户

用户是家居系统的体验者，也是家居系统的控制者，同时也是规则和模式参数的设置者。

系统中基于用户请求控制家居需要用户的位置，所以用户位置这一个特性不可或缺，同时考虑到后续控制权限的问题（比如孩子不能控制热水器温度，不能控制微波炉），需要添加用户所能控制的控制对象列表。此外，前文中提到的多种状态，比如用户睡眠，工作，学习等常规的状态也需要进行抽象，最后还需要抽象出用户的名称和用户的年龄等常见的属性。

由此可得用户需包含下列属性：姓名、年龄、所在位置、用户状态、控制的传感对象编号列表。

#### 3.3.2.5规则

在智能家居系统中，规则起到了连接传感对象和控制对象的作用，系统通过监听传感对象的传感属性，来检测是否触发规则，触发规则后，规则告诉系统应该做哪些操作。

不同的规则有以下相同特性，具备一定触发条件，有相关的控制对象，及后续对该控制对象的操作，并且需要考虑相应操作的优先级。例如“温度大于28℃，

开启空调”和“下午1-5点打开窗帘”，这两条简单的规则，它们都有触发条件（温度的条件和时间的条件），都有控制对象（空调和窗帘），都有操作（打开），并且如果对同一电器进行操作，优先级

所以，规则抽象后，具有以下概念：触发条件、控制对象、具体的操作、规则的优先级。

#### 3.3.2.6 模式

多变的需求以及需求间的相似性，产生了模式，模式和规则功能相似，也起到了连接传感对象和控制对象的作用，并告知系统进行何种操作。

对比规则，模式更加复杂多变，模式可以看成是多条规则的组合，也可以看成是不同的触发条件，形成了不同的响应。规则多为一对一的关系，而模式可以看成多对多的关系，即多个触发条件，对应多个相应。例如休闲模式，它的触发条件可以是用户的请求，也可以系统的检测，相应也随着多变，如果用户在客厅，则打开电视机和音响，如果用户在书房，则打开电脑。

因此，模式抽象后，具有以下概念：触发条件、控制对象、具体的操作、模式的优先级。

#### 3.3.2.7 家居系统

家居系统是容器，包含了人和物，物指的是生活环境，具体来说就是一个个不同功能的房间，人指的是在其中生活的用户。

在完整智能家居系统中，需要对不同的家居系统进行区分和管理，因此每一个家居系统都应有自己的编号，来进行标识。

由此，家居系统包括以下属性：人物列表、房间列表、编号。

## 3.4 系统整体的流程和架构

### 3.4.1 系统整体流程

在3.2的需求分析中可知，家居系统整体上共有两个流程，第一种是基于用户主动请求的（通过语音输入）的控制对象操作，第二种是系统对规则和模式进行匹配检测，模式或规则触发后对控制对象进行操作，再结合前文提到的语音识别技术和中文分词技术，提出图3.1所示的系统整体流程。

首先，系统进行初始化（包括导入的房间，人物，相应规则参数等），然后系统启动，此后系统在每一时刻会对音频文件进行检测输入，如果某一时刻有音频文件输入，系统进行语音识别，将音频文件转换成文本，再通过中文分词，将文本文件切分成一个个字段，通过字段的解析的模型来判断用户的意图，并对具体的家居进行具体的操作。此外，系统也时刻检测模式规则，一旦模式或规则条件触发，系统则对具体家居进行相应的操作。

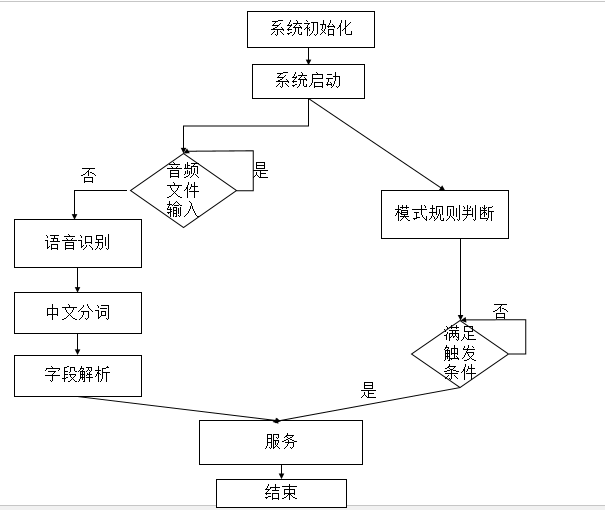


图3.1 系统流程图

### 3.4.2 系统架构

基于家居生活环境的特征，结合实际需求，本文提出图3.2的系统架构，如图3.2所示，系统总体架构主要分三个层次，数据层、应用层、概念层。

数据层：该层主要是数据的读入和存储，具体展开来说是家居环境中人和物、房间信息的存储和读取，同时也定义了存储的空间。在本系统中，仅仅由于是系统原型仅仅对系统做了简单的读入和存储，后续数据层需要考虑数据的可视化管理以及相关数据库如Mysql等成熟数据库技术的应用。

应用层：基于用户的输入或者通过规则引起实现对环境的规则、模式的匹配。该层通过加载家居模型，通过语音识别应用或者规则引擎，产生对具体家居的具体操作（服务），来改变环境，提高用户的舒适度。应用层中还应有开发者或者用户定义的规则和模式库来为整个系统指定规则，指导整个系统如何进行工作。

概念层：定义系统中的实物概念，如用户、房间、空间、控制对象、传感对象，通过继承实现多种子类，同时通过工厂模式为数据层提供系统载入的接口。

以上各层关系如下：数据层利用概念层的工厂模式相关类来创建对象，并且对数据做初始化，应用层加载数据层初始化完成后的家居空间，在用户输入时或者匹配环境时，通过调用概念层各个子类中get方法来获取所需的属性（如通过get获取传感器获取的环境参数值），并执行传感对象提供的相应动作方法（如heating等）来做相应的动作或者执行传感对象提供的相应的set方法对相应的属性进行设置。

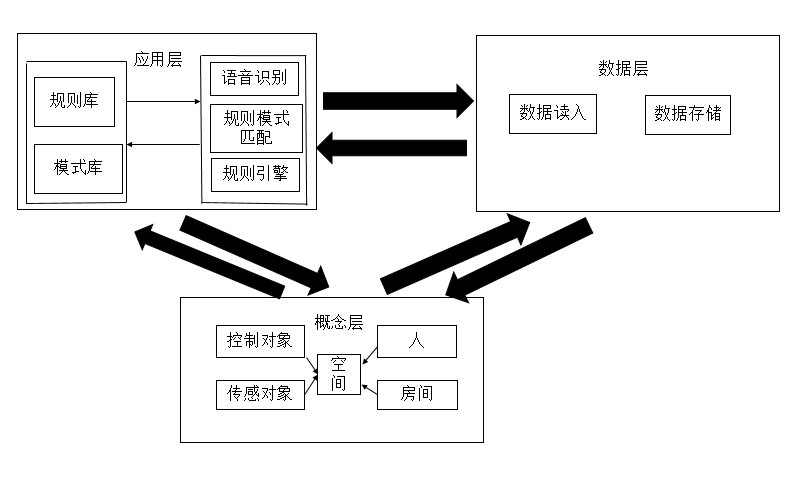


图4.2系统架构图

# 第四章 智能家居系统的具体实现和测试

前文中，对智能家居系统的流程和整体架构进行了定义，本章则着重描述重点功能的具体设计和实现。本项目基于jdk1.8，开发编译器为eclipse，整体代码目录结构如下图4.1。

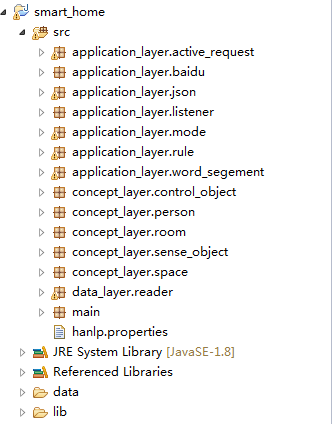


图4.1 系统代码目录层次图

## 4.1 语音识别应用的实现

在系统流程中，定义了语音识别的流程，首先系统获取用户的输入音频，在获取用户的音频后，调用语音识别的API，将语音打包成相应的格式发送给服务提供商，获取服务提供商返回的文本，通过中文分词将

语音识别：调用了百度语音的API（application\_layer.baidu包），使用Http请求将用户输入的wav格式语音转换成对应的json（application\_layer.json包），并且解析出json中的中文文本。

中文分词：导入了Hanlp（哈工大nlp）的jar包，使用Hanlp的标准分词对文本进行切分（application\_layer.word\_segement包），将文本转换成String数组。

字段解析：在active\_request包中的active\_request.java中对用户主动请求进行切分，将其切分成地点、控制对象、动作（开或者关）、设置属性、设置

属性值、其他动作（如加热、保温）后，由具体某个相关的电器请求处理类进行具体的操作。具体方法如下：

air\_conditioning\_request T=**new** air\_conditioning\_request();

T.control(rList.get(i), actionOpenOrClose, attrNum, attr, otherAction);

## 4.2 规则引擎的实现

由于开发语言选择的是Java，在本次系统原型规则引擎实现中，选用了基于Java的规则引擎工具Drools。

Drools是一种基于Java的开源规则引擎，适用于多变复杂的业务逻辑，可将复杂的规则从硬编码中解放出来，以规则脚本的形式存在文件中，使得规则的变更可以不需要修改代码。[10]

在application\_layer.rule中，不同的家居对应了不同的规则类，规则类首先加载该家居系统中对于该电器的定制化配置规则文件，遍历房间中的控制对象列表，如果房间中的控制对象列表是对应的家居，那么调用房间中传感器的get方法来获取相应的属性，并将对象传入规则引擎，利用规则引擎drools对规则按照优先级进行自动的执行和解析，实现了规则的自定义和个性化定制。以air\_conditioning\_rule为例，规则文件如图4.3所示，定义了两种规则，温度大于28℃开启空调，设置成冷风，并且温度设置为24℃，温度小于10℃打开空调，并且设置成热风，温度设为24℃，其中salience表示规则的优先级，when表示条件，then中表示结果。

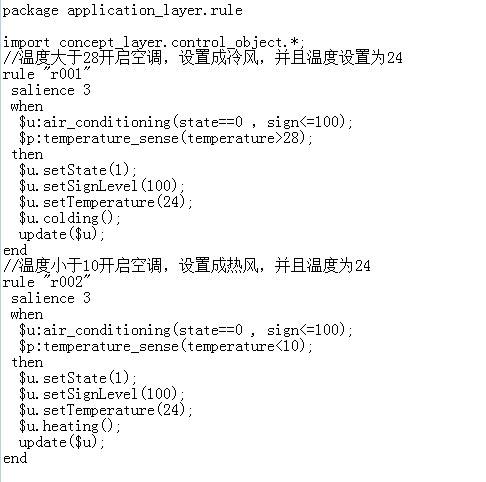


图4.2 air\_conditioning\_rule.drl

模式：在application.mode中对不同的模式结合规则引擎进行了不同的触发条件的定义，目前系统共定义并且实现了四种基本的模式，睡眠模式、节能模式、家中无人模式、安全模式。

睡眠模式检测用户的状态或者用户的主动请求，如果用户正处在睡眠状态或用户通过语音表达想睡眠的意图，会对用户房间中开启的家居进行睡眠模式的设置，如果空调打开，会将空调的模式调整成睡眠模式，如果灯打开，会将灯关闭，同时关闭已经打开的窗帘。睡眠模式尽量保证用户所处房间的家居处于安静工作的状态以及保证房间的光照降到最低。

节能模式检测用户的请求，如果用户主动请求系统节能，会对用户所在房间正在运行的家居进行节能模式的设置。节能模式尽量保证用户所使用的家居处于省电状态，减少不必要的浪费。

家中无人模式检测所有用户所在位置，如果所有用户均不在家，会对除了冰箱等必须开启的其他家居进行关闭，以达到节能和安全的目的。

安全模式通过获取烟雾传感器提供的烟雾浓度值，来判断房间的烟雾是否超过警戒值，如果超过警戒值，会对家居系统中所有的房间的所有家居进行关闭的

操作，并且做出通知用户和小区物业的动作，并且设置所有电器当前信号值为最高，直到用户解除警报。

## 4.3 基本抽象概念的关系及定义

控制对象：在concept\_layer.control\_object对各个控制对象进行了抽象，定义了control\_object.java这个基础类，也由上文面向对象抽象后的属性设置了对应的get，set方法（图4.3），并且对常用的家居（窗帘、空气净化器、空调、冰箱、微波炉、灯、热水器）进行了面向对象层面上的继承和对应属性方法扩展，同时也创建了对应的工厂类（control\_object\_factory.java）。

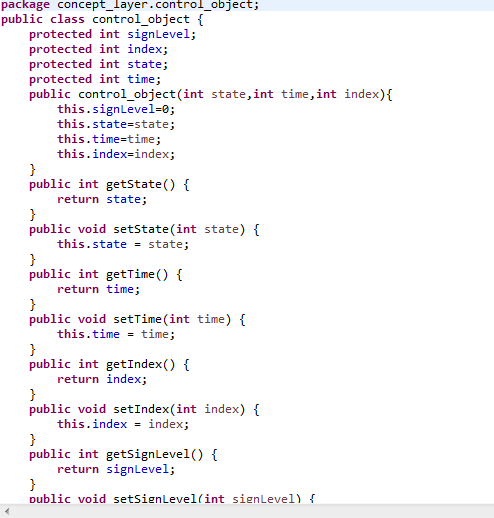


图4.3 control\_object.java的具体实现

传感对象：在concept\_layer.sense\_object包中对传感对象进行了抽象，定义sense\_object.java这一基础类，也根据上文分析后面向对象抽象后的属性设置了对应的get，set方法，也对常用传感器（温度传感器、湿度传感器、烟雾传感器、光照传感器），也创建了sense\_object.factory.java来实现工厂模式。

房间：在concept\_layer.room.room中实现了对房间的抽象，根据上文面向对象的分析结果，定义了房间的面积、编号、控制对象列表、传感对象列表、名称这些属性。

空间：在concept\_layer.space.space中实现了对空间（家居系统的抽象），根据前文面向对象的分析结构，定义了空间中的房间列表、人物列表、编号这些属性。

这些基本的概念定义和关系可以用如下UML图（图4.4）来表示：

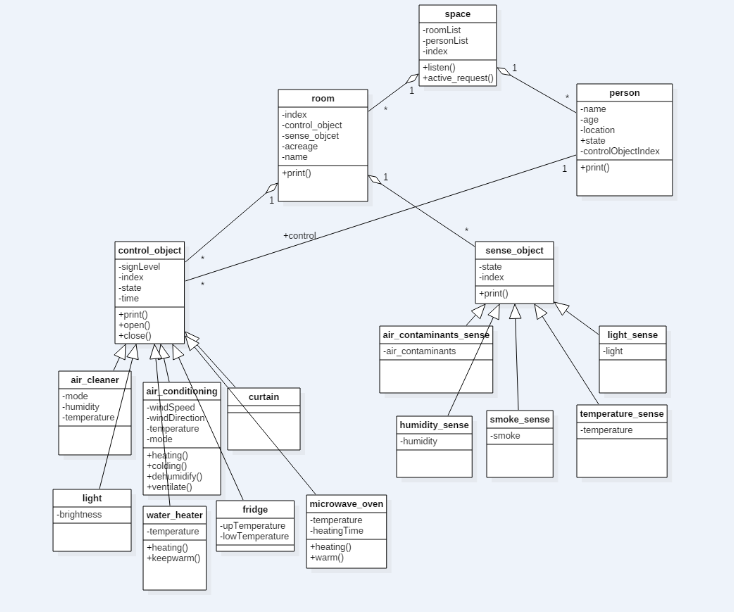


图4.4 抽象概念UML图

## 4.4测试

上文中提出智能家居的整体架构和系统流程，并且对其各个模块进行了详细的分析和设计以及代码上的实现。本章则基于常见的生活场景和需求，对家居系统原型进行测试和验证。

### 4.4.1 基于规则和模式的测试

#### 4.4.1.1 基于规则的测试

常见场景描述：正值盛夏，卧室内的温度30℃，小明在卧室内感到燥热，此时，空调自动打开，并且自动设置成冷风模式，降低了室内温度，使小明感到舒适。

相关规则：当房间中有人且温度大于28℃时，开启空调，并设置冷风模式，空调温度设置为24℃。

相关数据：初始化后调用space中print方法，对房间和人物进行输出结果如图4.5。

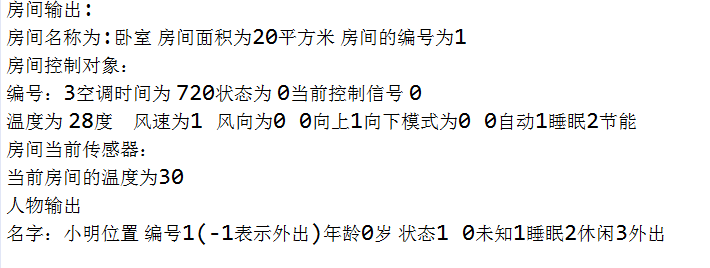


图4.5 基于规则的测试（输入数据可视化结果）

测试结果：调用监听方法后，并对space中的房间和人物进行输出结果如图4.6。

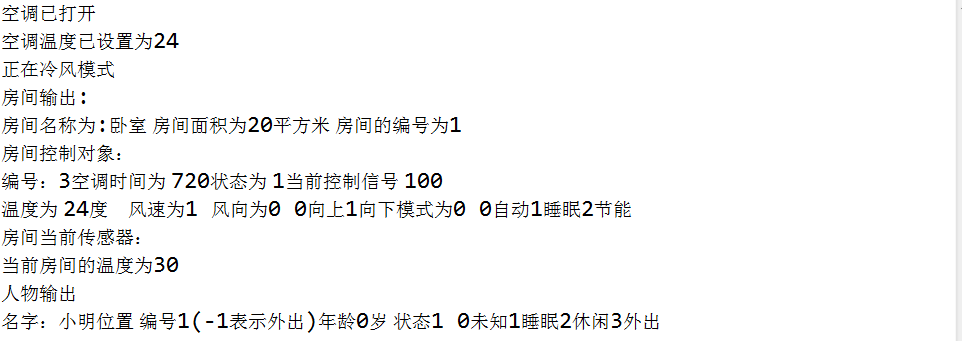


图4.6 基于规则的测试（输出结果可视化结果）

可以发现，空调的状态发生了改变，由0变成1，即由关闭状态变成了开启状态，同时空调的设置为24℃，正在冷风模式，并且当前的控制信号也发生了改变。

#### 4.4.4.2基于模式的测试

常见场景描述：主人小明和小红中午12点出门，家中的空调忘记了关闭，智能家居系统发现家中无人，启动了无人模式，从安全和节能的角度上关闭了空调。

相关数据：初始化后调用space中print方法，对房间和人物进行输出结果如图4.7。

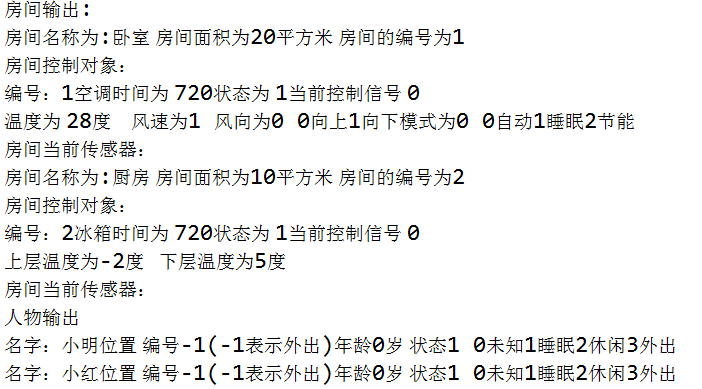


图4.7 基于模式的测试（输入数据可视化结果）

测试结果：调用监听方法后，并对space中的房间和人物进行可视化输出，

结果如图4.8。对比图4.7可以发现，空调已经关闭，但是冰箱并没有关闭，并且空调的控制信号发生了改变。

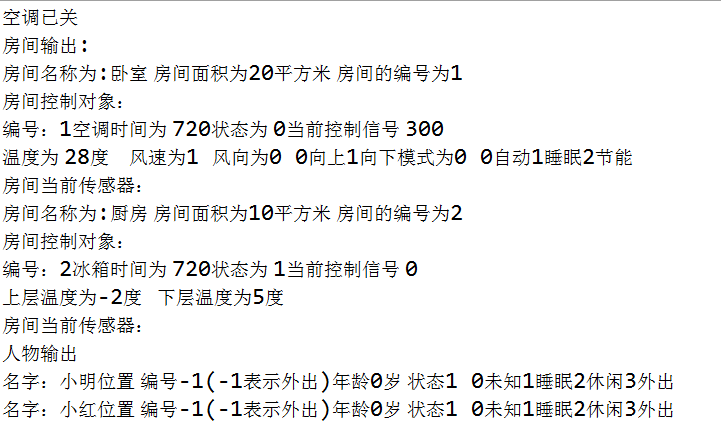


图4.8 基于模式的测试（输出结果可视化结果）

### 4.4.2 主动请求的测试

常见场景描述：正值盛夏，小明马上下班，家中无人，想要家里的空调进行提前制冷，对系统输入内容为“打开客厅空调，制冷，温度设置为28度”的音频文件。系统接收到小明的请求后将其通过语音识别，中文分词，字段解析获知了小明的意图，打开了空调，并温度设置为28℃。

相关数据：初始化后调用space中print方法，对房间和人物进行输出结果如图4.9。

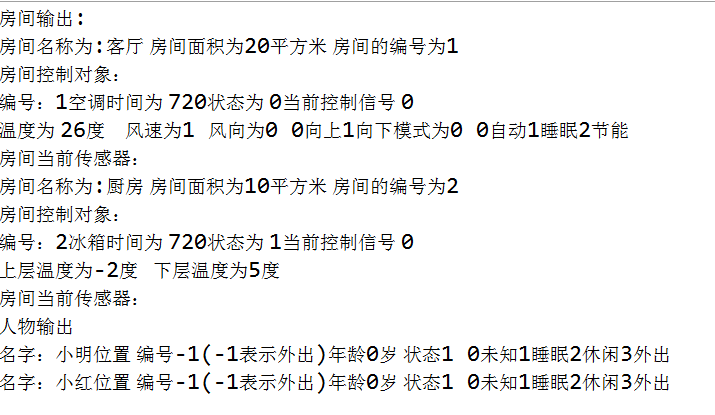


图4.9 主动请求测试（输入数据可视化结果）

测试结果：输入内容为“打开客厅空调，制冷，温度设置为28度”后，并定义请求者为小明后，调用系统主动请求的处理方法后，对系统当前状态进行输出，结果如图4.10。对比4.9，可发现空调的状态发生了改变，从关闭到打开，并且空调的温度属性也根据小明的需求发生了改变，从26℃变为了28℃，同时也设置为冷风模式。

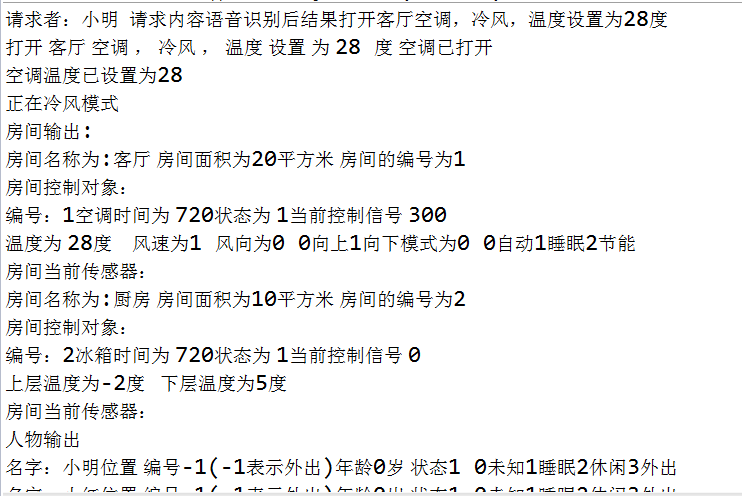


图4.10 主动请求测试（输出数据可视化结果）

# 第五章 总结与展望

## 5.1 总结

作为物联网中重要的一环，家居系统也随着时代的发展和科技的进步不断的革新，越来越多的家居系统的案例和架构不断涌现。但是人们对家居的环境舒适度和智能化程度的要求也不断提高，迫切的需要相关成熟的智能家居解决方案或者智能家居应用面世。然而，市面上对智能家居的研究大多停留在硬件或通信协议上，很少涉及软件层面上，并没有考虑到实际生活场景以及对生活场景的抽象。虽有一定数量的解决方案，但是缺少普适的、整体的解决模型。为此，本文从软件层面出发，通过面向对象的思想，对家居系统进行软件层的抽象，并且结合当前应用较广的人工智能中的语音识别技术，对家居系统进行智能化，提出整体解决流程和系统架构，以适应多变的环境和用户复杂的需求。

通过概括归纳日常生活场景以及常见家居布局，结合语音识别技术，定义智能家居系统的流程，设计一套智能家居系统原型，包括概念层、数据层、应用层，详细说明各层之间的关系和流程，并对其进行代码上的实现。最终，以常见的生活典型场景，对系统原型进行了进一步的验证。

## 5.2 展望

虽然本文提出了一套切实可行的智能家居系统，从软件层面上弥补了一定的研究空白，但是如何应用于实际生活，提高用户的体验度，仍有很多问题需要结局。本文还有以下问题可以进行扩展和探讨：

1）：本文的规则虽使用了规则引擎，满足了不同用户对规则的需求，同时也优化了规则的处理流程，但是规则的编写仍需要开发人员获取用户需求后，结合系统的整体架构和概念类的的定义，进行编码，如何将用户的需求自动的转换成符合条件的、可执行的代码或者规则文件，进一步提升用户的体验，降低开发的人力成本，需要后续的探索。

2）：基于用户历史行为的数据挖掘可以指导家居系统，并提升用户的体验，增加系统的智能化，如何对用户的历史行为进行数据挖掘，并且总结成可执行的规则，添加到本系统中，有待后续的研究和实验。

3）：本文的语音识别中的字段解析并不能理解特别复杂的请求，如何处理用户更加复杂的请求，在日后的系统研究中，有待开发者和设计者的探究。

4）：本文虽然对家居系统的各个组成结构进行了编号的划分，但是并没有给出对家居系统中的各个组件进行数据管理和维护的方案，如何管理庞杂的家居系统中的各个复杂的元素，需要后续数据库的支持，也是可以研究的内容。

# 参考文献

1. 朱敏玲，李宁 智能家居发展现状及未来浅析 电视技术 2015，39（4）
2. Xu Heyuan A Study on Ubiquitous Network and Internet of Things ZTE COMMUNICATIONS,2010,16(Z1),pp.13-16
3. 马畅俊 语音识别技术在智能家居中的应用现状浅析 《科技传播》 2018.1下
4. 黄贤立，罗冬梅 基于讯飞语音技术的智能家居APP控制 福建电脑 2016年第8期
5. 王富中，黄文浩 基于语音识别技术的智能控制系统设计 [J].自动化与仪表，2006,21（4）：8-10
6. 余战秋 中文分词研究及其应用初探 《电脑知识与技术》 2004.32
7. <http://finance.sina.com.cn/money/future/indu/2017-07-07/doc-ifyhvyie0468192.shtml>
8. <https://baike.baidu.com/item/住宅设计规范/8915001>
9. 曾君 基于本体的自适应控制在智能家居中的研究 复旦大学 硕士学位论文
10. <http://www.360doc.com/content/15/1116/16/16002580_513612029.shtml>

# 致谢

随着本科论文的完成，我的四年大学生涯也随之接近尾声，感谢在这四年中帮助过我的同学和老师。

在本文的完成过程中，李敏波老师帮助了我很多，不但提供给我以思维的启发，并且督促我的进度，对我的工作精益求精，他的负责、耐心、专业值得我学习，也给我今后的生活学习树立了榜样，在此感谢李敏波老师。

除了李老师的帮助，实验室的吴宇学长帮助了我很多。吴宇学长和我反复讨论系统的架构，规范开发流程，深入技术细节，在技术上给了我很多的启发和帮助，在此谢谢吴宇学长。

此外，身边的同学和辅导员在生活中也帮助我，关心我，让我能够以更好的状态投入学术的研究上，谢谢你们。

感谢我的父母，多年的关心和付出，使我能够专心自己的学业，完成研究工作。

感谢复旦大学软件学院，提供了优质的生活学习平台，无私的培养了我。

最后，感谢百忙中对我的论文进行评阅、答辩的导师。