

北京交通大学

本科毕业设计（论文）

智能音箱语音控制系统的设计与实现

**Design and Implementation of Smart Speaker
Voice Control System**

学 院：_____

专 业：_____

学生姓名：_____

学 号：_____

指导教师：_____

北京交通大学

2017 年 5 月

学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

指导教师签名：

签字日期： 年 月 日

签字日期： 年 月 日

中文摘要

摘要：人类的语言交流是人类交互的主要方式，从键盘，鼠标，触屏控制，人机交互的本质始终没有改变，无法完成对话式的交流，而语音交互的出现打破了这一现状，它可以解放双手，甚至解放双眼，给人带来极大的快感，它也将是智能时代的主要交互方式。由于近年来深度学习的突破以及计算能力的提升和数据积累，让语音识别得到了快速的发展，麦克风阵列技术兴起更是实现了远场自由语音交互的需求，国内的科大讯飞、声智科技，思必驰等已经拥有比较成熟的麦克风阵列技术，这种技术为智能音箱的核心技术，为智能提供了解决方案。而音响这样一个电子设备是家庭中除了手机，耳机，家电以外最常见的的设备了，仅国内它就是千亿级别的总产值，需求是刚性的。这样一款设备与语音交互结合产生的智能音箱必定拥有广阔的市场。

该项目来源于我在珠海市魅族科技有限公司北京分公司的实际项目——“gravity 智能音箱”。该产品是一款基于远场语音交互的音箱，它通过语音识别、语意理解、自然语言处理、语音合成等技术实现各种功能，不仅能控制音箱音量、搜索歌曲、歌曲切换，还提供了日常生活中的场景的语音对话功能，如：百科，新闻，天气，时间，闹钟等。在音箱的研发过程中，作者的工作是针对该产品的语音控制系统的设计与实现。主要内容是需求分析、系统设计、编码实现等。本文分析了项目背景与意义，明确了需求跟架构设计，在此基础上，对各个模块提出了实现方案。

目前，音箱通过语音控制系统已经实现了语音与音箱功能的对接以及日常生活中的场景对话。

关键词：智能音箱;语音控制;语音交互

ABSTRACT

ABSTRACT: Human language communication is the main way of human interaction. From the keyboard, mouse, touch screen control, the nature of human-computer interaction has not changed, can not complete the dialogue of the exchange, but the emergence of voice interaction will break the status. It can give people great pleasure by freeing Hands, even freeing eyes. It will also be the main interactive in the intelligence era. In recent years, due to the breakthrough of deep learning and the improvement of computing power and data accumulation, so that voice recognition has been rapid development, the rise of microphone array technology is to achieve a far-field free voice interaction needs, In domestic KEDAXUNFEI, SHENGZHIKEJI, SIBICHI etc already have a more mature microphone array technology, this technology for the smart speaker core technology, provides a solution for the intelligence. The sound of such an electronic device is the family in addition to mobile phones, headphones, home appliances other than the most common equipment, and only it is the domestic level of 100 billion output value, the demand is rigid. Such a combination of equipment and voice generated by the combination of smart speakers must have a broad market.

The project comes from my actual project in Zhuhai City Meizu Technology Co., Ltd. Beijing Branch - "gravity smart speaker". The product is a far-field voice interactive speaker, which through voice recognition, semantic understanding, natural language processing, speech synthesis and other technologies to achieve a variety of functions, not only can control the speaker volume, search songs, song switching, also provides daily The scene of life in the voice dialogue function, such as: Encyclopedia, news, weather, time, alarm clock and so on. In the speaker of the R & D process, the author's work is for the product of the voice control system design and implementation. The main content is demand analysis, system design, coding and so on. This paper analyzes the background and significance of the project, clarifies the demand and architecture design, on this basis, put forward the realization of the various modules.

So far the speaker through the voice control system has achieved the voice and speaker function docking and daily life scenes dialogue.

KEYWORDS: Smart speaker; voice control; voice interaction

目 录

中文摘要.....	I
ABSTRACT.....	II
目 录.....	III
1 引言	1
1.1 论文背景与意义.....	1
1.2 智能音箱应用现状.....	1
1.3 论文主要工作.....	2
1.4 论文组织结构.....	2
2 相关技术	4
2.1 语音识别.....	4
2.2 语音合成.....	4
2.3 自然语言处理.....	5
2.4 ANDROID ACTIVITY.....	5
2.5 JSON 数据	5
2.6 多线程.....	6
2.7 本章小结.....	6
3 需求分析	7
3.1 系统简介.....	7
3.2 系统功能描述.....	7
3.3 功能性需求分析.....	8
3.3.1 系统用例分析	8
3.3.2 音乐类用例分析	10
3.3.3 有声类用例分析	11
3.3.4 天气类用例分析	12
3.3.5 新闻类用例分析	13
3.3.6 闹钟与提醒用例分析	14
3.3.7 百科类用例分析	15
3.3.8 聊天类用例分析	16
3.4 非功能性需求分析.....	16
3.5 本章小结.....	17
4 系统架构与设计	18
4.1 系统总体架构设计.....	18
4.2 系统接口设计	21
4.2.1 process 监听接口	21
4.2.2 UI 交互接口	21

4.2.3 音箱命令执行接口	22
4.3 本章小结	24
5 模块设计与实现	25
5.1 特定场景模块	25
5.1.1 设计描述	25
5.1.2 流程图	26
5.1.3 类图	27
5.2 魅族 NLP 模块	28
5.2.1 新闻类	28
5.3 第三方 NLP 模块	30
5.3.1 音乐类	30
5.3.2 有声类	32
5.3.3 天气类	34
5.3.4 新闻类	37
5.3.5 闹钟与提醒类	38
5.3.6 百科类	41
5.3.7 聊天类	42
5.4 本章小结	44
6 结论	45
参考文献	46
致 谢	47

1 引言

本文阐述的是基于智能音箱的语音控制系统的设计是实现，其来源是我实习中的实际项目——“gravity”智能音箱。语音控制系统是为了实现音箱功能跟语音的对接以及关于日常生活中的场景的语音交互，是该项目的核心系统之一。根据软件工程理论对本项目进行了设计与实现。本章将从论文背景与意义、智能音箱应用现状、论文主要内容、论文组织结构四个方面对项目进行简单介绍。

1.1 论文背景与意义

人类的语言交流是人类交互的主要方式，从键盘，鼠标，触屏控制，人机交互的本质始终没有改变，无法完成对话式的交流，而语音交互的出现打破了这一现状，它可以解放双手，甚至解放双眼，给人带来极大的快感，它也将是智能时代的主要交互方式。由于近年来深度学习的突破以及计算能力的提升和数据积累，让语音识别得到了快速的发展，麦克风阵列技术兴起更是实现了远场自由语音交互的需求，国内的科大讯飞、声智科技，思必驰等已经拥有比较成熟的麦克风阵列技术，这种技术为智能音箱的核心技术，为智能提供了解决方案。而音响这样一个电子设备是家庭中除了手机，耳机，家电以外最常见的的设备了，仅国内它就是千亿级别的总产值，需求是刚性的。这样一款设备与语音交互结合产生的智能音箱必定拥有广阔的市场。

该项目来源于我在珠海市魅族科技有限公司北京分公司的实际项目——“gravity智能音箱”。该产品是一款基于远场语音交互的音箱。它通过语音识别，语意理解，结果处理，语音合成实现各种功能，不仅能对控制音箱音量，搜索歌曲，歌曲切换，还提供了很多日常功能，如：百科，新闻，天气，时间，闹钟。实习的工作是针对该产品的语音控制系统的设计与实现，完成语音交互与音箱实际功能的对接。

智能音箱具有广阔的市场，可以创造巨大的财富，同时它也是未来布局智能家居的入口。语音控制系统是语音交互技术与音箱实际功能的对接，加强语音的交互体验[5]，系统的稳定性直接影响到产品的用户体验以及生命周期的长短，它的业务流程未来甚至可以拓展到其他智能家居上。

1.2 智能音箱应用现状

2015 年智能音箱出现，国外亚马逊 Echo，国内有京东的叮咚音箱，阿里的小飞。智能音箱一开始就被认为是家庭互联网的入口之一[7]，与智能家居进行融合也是其使命之

一，各个巨头出于这方面的考虑都开始抢占这个领域。但是智能音箱还处于起步阶段，有很多困难还待突破。音箱特别是其中的中高端音箱，本来就强调品牌，而且是技术门槛较高的领域，而智能音箱需要应用声学设计、远场拾音、无线技术[14]、语音识别、自然语言处理等众多技术[6]，不仅使技术更为复杂，而且更加依赖音乐内容平台的支持，这些诸多因素都是待创业者解决的难题。用户对智能音箱的要求也远非语音识别所能做到的那样简单，还需要更深入的语音交互才行，而且这种交互应该对接上日常生活中的场景[5]。即使众多科研人员要解决某个特定领域的语音交互都是困难的，更何况如此广泛的领域。

目前智能音箱的主要研究方向是好的语音交互体验，更丰富的内容交互，以及对其他智能硬件产品的控制。

1.3 论文主要工作

结合语音识别，语音合成，自然语言处理，为了很好的实现音箱功能跟语音的对接以及关于日常生活中的场景的语音交互，语音控制系统的设计与实现是非常有必要性，同时也是非常重要的，这也是我在魅族实习的主要内容。

本论文将根据软件工程项目开发流程对语音控制系统从需求分析，架构设计，模块详细设计与实现等各个方面对系统进行论述。作者把语音控制系统根据语音识别后自然语言的处理方式分做三个模块：特定场景模块，魅族 NLP（自然语言处理，Natural Language Processing）模块，第三方 NLP 模块。然后分别从用户角度对其进行了用例分析，之后通过流程图，架构图对系统架构进行了整体性说明。然后从又设计描述，流程图，类图的角度对系统主要功能进行了详细阐述。最后对整个论文进行了总结。为了方便读者阅读本论文，以及对系统核心技术有所了解，作者对相关技术进行了阐述。

1.4 论文组织结构

本论文是以项目“智能音箱语音控制系统”的设计与实现作为研究对象，依据软件工程项目开发的流程进行论述，主要分为六个模块，引言、相关技术、需求分析、系统架构、模块设计与实现、总结。

第一章：引言。分为四个小节，前两个小节对课题的研究背景与意义、智能音箱应用现状进行简述，表明了本课题的研究背景和需要解决的问题；后两个小节说明了论文主要内容和论文的组织结构。

第二章：相关技术概述。分为七个小节，前六个小节分别介绍了整个项目用到的关

键技术。最后一个小节为本章内容的小结。

第三章：需求分析。分作五个小节，第一个小节对系统作了简介，第二节是介绍系统的模块划分，把系统根据自然语言处理方式分作三大模块，第三节是对系统功能性需求进行论述，通过用例图分析了系统的主要功能，第四节是对系统的非功能性需求进行论述，从五个方面考虑了该需求。最后一个小节为本章内容的小结。

第四章：系统架构。分作三个小节，第一小节对系统整体架构进行说明，通过流程图，架构图进行了阐述。第二小节对系统主要接口进行了论述，讲解了模块与模块间是如何交互的。最后一个小节为本章内容的小结。

第五章：模块设计与实现。分作四小节，前三个小节对三大模块进行了详细论述，每一个模块都从设计描述，流程图，类图进行了分析阐述。最后一个小节为本章内容的小结。

第六章：总结，对论文撰写的总结和对所有支持、帮助我完成这个项目的人表示感谢，以及参考文献和英文文献翻译。

2 相关技术

“gravity”智能音箱的软件端是基于 Android 平台实现的，语音控制系统的软件端的重要组成部分。在项目的设计实现过程中，用到的关键技术有：语音识别、语音合成、自然语言处理、Android activity、JSON 数据、多线程。

2.1 语音识别

语音识别(ASR, Automatic Speech Recognition)是让机器通过识别和理解把语音信号转变为相应的文或命令，也就是让机器听懂人类的语音。目前主流的语音识别技术是基于统计模式识别的基本理论，这种语音识别系统由信号处理包括统计信号处理和特征提取、声学模型、发音字典、语言模型、解码器等几个基本模块一起构成。信号处理和特征提取模块的主要任务是从输入信号中提取特征，供声学模型处理，同时，它一般也包括了一些信号处理技术，用来尽可能降低信道、环境噪声、说话人等因素对特征造成的影响。声学模型，目前多采用基于隐马尔科夫模型的建模方法。发音字典包含系统所能处理的词汇集及其发音，实际提供了声学模型建模单元与语言模型建模单元间的映射。语言模型对系统所针对的语言进行建模，目前各种系统普遍采用的还是基于统计的 N 元文法及其变体。解码器通常是在给定了根据语法、字典对马尔科夫模型进行连接后的搜索的网络后，在所有可能的搜索路径中选择一条或多条最优路径作为识别结果。

2.2 语音合成

语音合成指将文字信息转化为可听的声音信息，它涉及声学、语言学、数字信号处理、计算机科学等多个学科技术，是中文信息处理领域的一项前沿技术。通过计算机语音合成则可以在任何时候将任意文本转换成具有高自然度的语音，从而让机器像人一样开口说话。

本项目使用的是百度语音合成技术。百度语音合成客户端 Android 版 SDK 是一种面向 Android 移动设备的语音合成解决方案，已支持 SDK 内部直接播放合成语音和从 SDK 获取语音数据，并支持男女声、语速、音调、音量、音频码率等设置。

SpeechSynthesizer，主要用来控制合成进程：设置参数，开始，结束，取消等。

getInstance() 方法获取语音合成引擎实例。setContext() 方法设置语音合成实例的上下文环。setSpeechSynthesizerListener() 方法设置语音合成监听器，回调合成和播

放进度以及错误信息。initTts()方法初始化合成引擎，可以指定使用 online 在线，或者 mix 离线混合引擎。mix 混合引擎会在 online 在线不能用的情况下自动使用 offline 离线引擎。batchSpeak()方法批量合成并播放文本文件，一次可传入一个 list，这个 list 会按顺序合成播放。onError()方法出错时的回调函数。

2.3 自然语言处理

自然语言处理(NLP, Natural Language Processing)就是用计算机来处理、理解以及运用人类语言[9]，它属于人工智能的一个分支，是一门语言学与计算机科学的交叉学科。简单自然语言处理主要是对文本进行分词、关键字提取等，复杂自然语言处理包括词义消歧、语法分析、语义理解、上下文关系等。自然语言在词法、句法及语义三个层次上充满了大量的歧义，自然语言处理的关键在于歧义消除。现在自然语言处理都要靠大规模语料库和统计机器学习方法，它涉及与语言处理相关的数据挖掘、机器学习、知识获取、知识工程、人工智能研究、语言学研究等。自然语言处理具有广泛的应用前景，特别是在信息时代，如：机器翻译、手写体和印刷体字符识别、语音识别及文语转换、情感分类、信息检索、问答系统等。

2.4 Android Activity

Activity 是用户接口程序，它给用户提供了一个交互式的接口功能，是 android 应用程序的基本功能单元[1]。Activity 本身是没有界面的，它创建了一个窗口，开发人员可以使用 setContentView()接口把 UI 放到 Activity 创建的窗口上，当 Activity 指向全屏窗口时，可以用方式实现，如：作为漂浮窗口，或者嵌入到其他 Activity。Activity 是单独的，用于处理用户操作。几乎所有的 Activity 都要和用户打交道，用户可以用来交互为了完成某项任务[11]，如：拨号、拍照、发送 email、看地图。创建一个新的 Activity 必须继承 Activity 类，从一个 Activity 跳转到另一个 Activity，可以使用 startActivity()方法。一个 Activity 从启动到停止要经过各种状态的转变，它被通知这种状态转变通过 Activity 的生命周期回调函数[10]。有许多回调函数一个 Activity 可能会收到，源于它自己的状态变化，无论系统创建它、停止它、恢复它、摧毁它，并且每个回调提供你完成适合这个状态的指定任务的条件。

2.5 JSON 数据

JSON 是一种轻量级的数据交换格式，采用完全独立于语言的文本格式，这些特性

使 JSON 成为理想的数据交换语言，易于人阅读和编写，同时也易于机器解析和生成。JSON 有两种表示结构，对象和数组。对象结构以 “{” 大括号开始，以 “}” 大括号结束。中间部分由 0 或多个以 “,” 分隔的 “key(关键字)/value(值)” 对构成，键值对之间运用 “,”（逗号）分隔。数组结构以 “[” 开始，“]” 结束。中间由 0 或多个以 “,” 分隔的值列表组成。GSON 是 Google 开发的 Java API，用于转换 Java 对象和 JSON 对象，是本项目所使用的对象转换器。gson 提供 toJson() 方法可以将对象转换成 JSON 字符串，想要把 JSON 字符串转换成 Java 对象可以使用其提供的 fromJson() 方法来实现。

2.6 多线程

线程是进程中独立运行的子任务，多线程就意味着多任务，可以在同一时间内运行多种不同的任务，极大的提高了 CPU 的利用率[2]。本项目的编码语言为 Java，在 Java 中，实现多线程编程主要的两种方式：一是继承 Thread 类，另一种是实现 Runnable 接口。多线程就涉及到线程安全与非线程安全，线程安全的情况下多线程对同一个对象实例中变量的并发访问不会产生脏读，而非线程安全就会产生脏读。对于非线程安全的情况下可以使用 synchronized 进行同步处理，也可以使用 volatile 解决变量在多个线程之间的可见性。对于并发量的控制可以使用 semaphore, semaphore 控制了当前的访问个数，提供同步机制，控制同时访问的个数。有时线程需要优先执行，这时就需要对线程进行优先级划分，设置线程的优先级可以使用 setPriority() 方法, 使用 sleep 可以在毫秒级别内控制线程的休眠。

2.7 本章小结

本章分别对项目中使用到的 6 大关键技术：语音识别、语音合成、自然语言处理、Android activity、JSON 数据、多线程依次进行了介绍。

3 需求分析

语音控制系统作为智能音箱的核心组成部分，其功能的设计与实现对于音箱用户的粘性影响重大。需求分析[13]通过对功能及系统行为建模分析，将用户需求精确化、完全化为系统的设计与实现定义了明确要求。本章将从系统简介、系统模块划分、功能性需求分析、非功能性需求分析四个方面进行论述。

3.1 系统简介

语音控制系统是为了控制语音交互流程，完成对音箱功能的对接以及提供用户有的一些关于日常生活中场景的语音交互[8]。语音控制系统根据语音识别后自然语言的处理方式分做三个模块：特定场景模块，魅族 NLP 模块，第三方 NLP 模块。特定场景模块：该模块主要处理一些音箱基本功能和增加用户友好的功能，让音箱更加人性化，基本功能如：上一曲、下一曲、音量调整，增加用户友好的功能如：闭嘴、我回来了。魅族 NLP 模块：该模块主要是通过关键字的内容识别、上下文分析、然后提供更好的服务，这里面主要是关于新闻的内容，这是魅族自己编辑、筛选过的内容。第三方 NLP 模块：该模板包含百科、诗歌、闹钟、有声娱乐等各种日常生活中的交互场景。除了以上提到的主要功能外，当然其中还贯穿着语音唤醒、语音识别、语音打断的全程开关的控制，以及日志输出，错误处理等[4]。

3.2 系统功能描述

语音控制系统根据自然语言处理方式分做三个模块：特定场景模块，魅族 NLP 模块，第三方 NLP 模块。特定场景模块包括：歌曲切换、音量调节、收藏、播放暂停、回家出门。魅族 NLP 模块包括：新闻。第三方 NLP 模块包括：百科、歌曲、闹钟、天气、新闻、聊天。如图 3-1 所示：

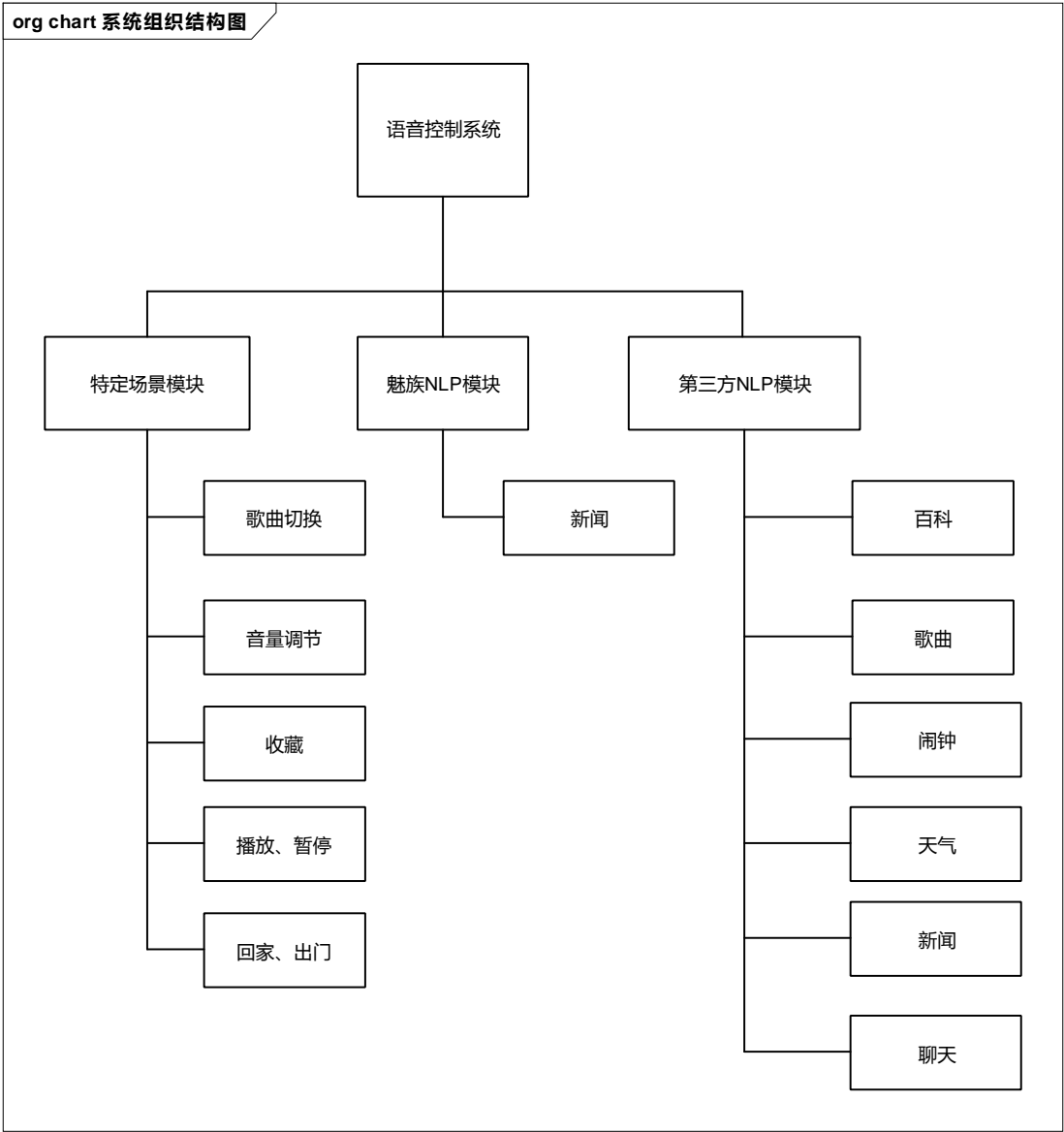


图 3-1 系统组织结构图

3.3 功能性需求分析

下面对介绍语音系统功能性需求分析,主要系统层面以及音乐类节目、有声类节目、天气、新闻、闹钟与提醒、百科类、聊天类等七大功能方面对系统进行分析。

3.3.1 系统用例分析

下面从用户角度对语音控制系统的进行分析,使用用例图对系统功能进行展示,根据系统模块划分分为特定场景模块、魅族 NLP 模块、第三方 NLP 模块三个模块,用例分

析如下所示。

首先是特定场景模块，该模块包括对歌曲切换、音量调节、收藏、播放暂停、出门回家等的语音控制，用例图如图 3-2 所示：

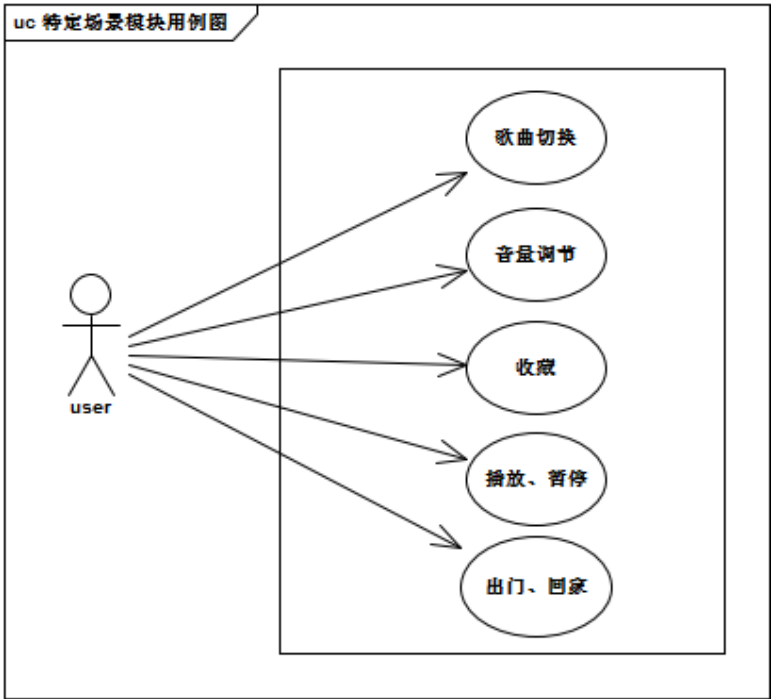


图 3-2 特定场景模块用例图

接下来是魅族 NLP 模块，该模块是针对包含新闻文本进行 NLP 以调用魅族自己编辑过的新闻，用例图如图 3-3 所示。

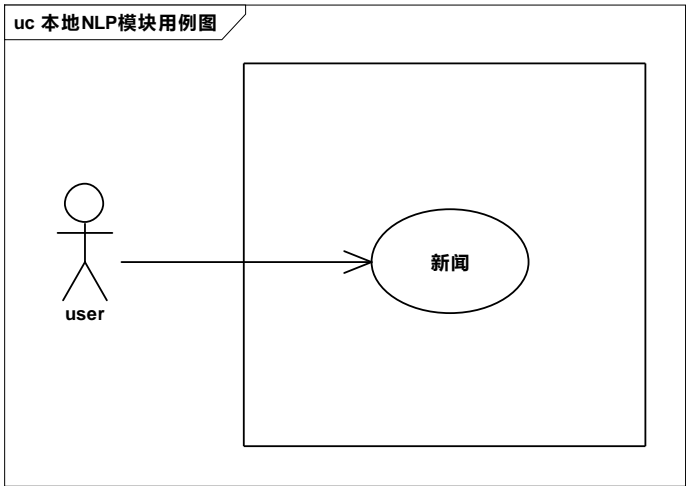


图 3-3 魅族 NLP 模块用例图

接下来是第三方 NLP 模块，该模块包括百科，歌曲、闹钟，天气，新闻，聊天类，用例图如图 3-4 所示：

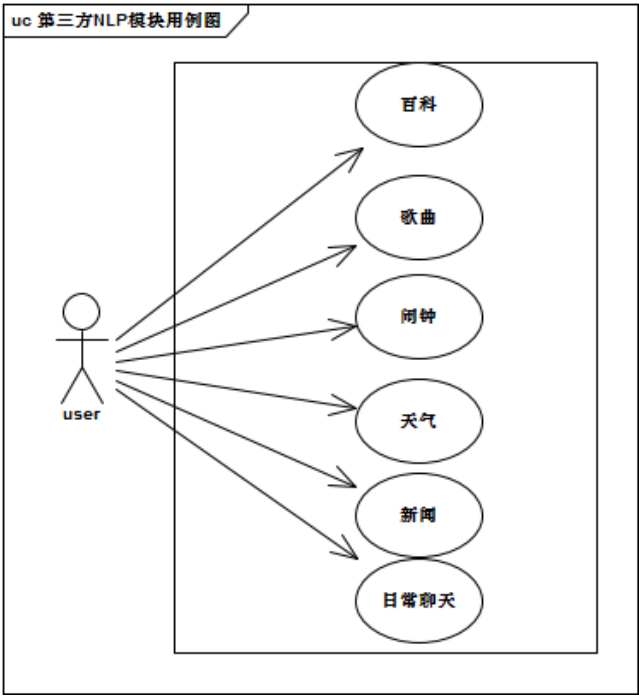


图 3-4 第三方 NLP 模块用例图

3.3.2 音乐类用例分析

音乐类节目的主要功能包括：按格式点播、按歌曲点播，按组合点播、歌手名+歌曲名点播、混合点播、分类点播、歌词点播、随机点播。用例图如图 3-5 所示：

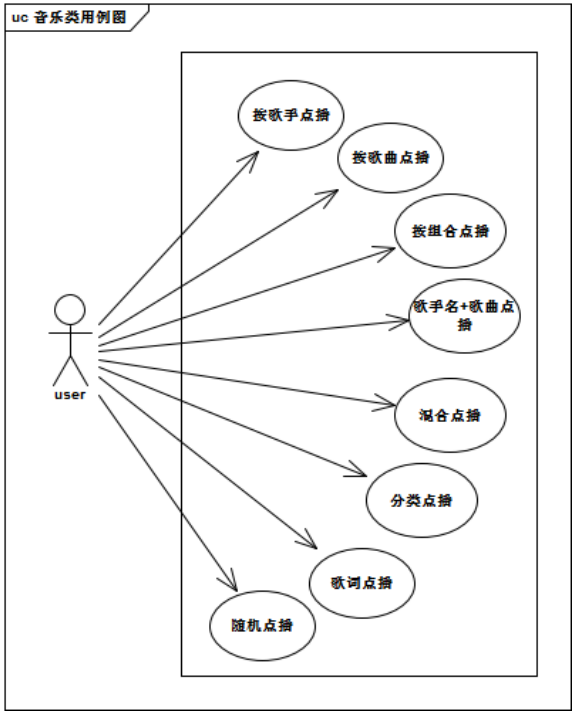


图 3-5 音乐类用例图

点播格式如表 3-1 所示：

表 3-1 音乐类点播格式表

功能名称	结果返回方式	例句
按歌手点播	返回按该歌手名查询得到的结果中，热度排行前 20 的歌曲	我想听周杰伦的歌
按歌曲点播	返回按该歌曲名查询得到的结果中，热度排行前 20 的歌曲	我想听稻香
按组合点播	返回以合唱者名称综合查询得到的结果中，热度排行前 20 的歌曲	我想听周杰伦和温岚合唱的歌
歌手名+歌曲点播	返回按意图搜索的结果里，热度最高的 1 首歌	我想听周杰伦的稻香
混合点播	返回按混合条件分别搜索后，能多重命中的歌曲	我想听周杰伦忧伤风格的歌
分类点歌	返回按关键词查询到的结果中，热度排行前 20 的歌曲	我想听咖啡厅音乐
歌词点歌	先按歌词查询对应歌曲，再按歌曲名称查询曲库并返回排行前 20 的歌曲	歌词里的“我竟悲伤得不能自己”的歌
随机点播	从热门歌曲曲库中，随机返回 20 首歌	随便来首歌

3.3.3 有声类用例分析

有声类节目的主要功能包括：按节目类型点播、按作家点播、按节目名称点播、按作家+节目名称点播。用例图如图 3-6 所示：

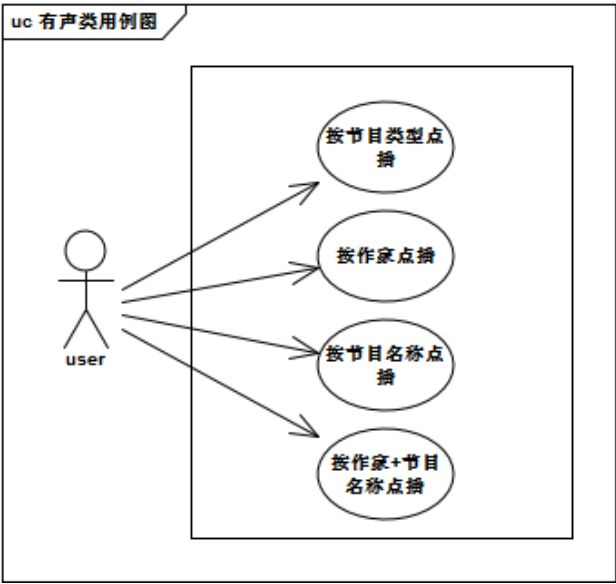


图 3-6 有声类用例图

点播格式如表 3-2 所示：

表 3-2 有声类点播格式表

功能名称	结果返回方式	例句
按节目类型点播	返回按节目类型查询得到的结果中，热度排行前 20 的节目	来段相声
按作家点播	返回按作家名称查询得到的结果中，热度排行前 20 的节目	我想听郭德纲的相声
按节目名称点播	返回按节目名称查询得到的结果中，热度排行前 20 的节目	我想听报菜名
按作家+节目名称点播	返回按作家名称和节目名称综合查询的结果中，能匹配的 20 歌节目	我想听郭德纲的报菜名

3.3.4 天气类用例分析

天气类的主要功能包括：按城市查询天气、按日期查询天气、按天气查询天气、按日期+城市查询、按天气+日期查询天气、按日期+城市+天气查询天气、询问后连续查询。用例如图 3-7 所示：

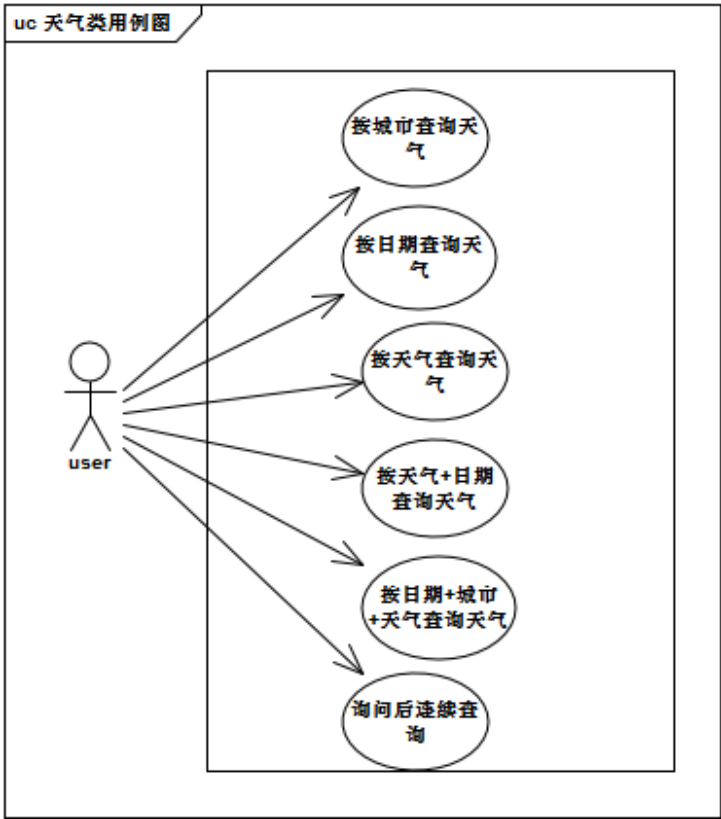


图 3-7 天气类用例图

表达方法如表 3-3 所示：

表 3-3 天气类表达方式表

功能名称	例句	返回内容
按城市查询天气	深圳天气	深圳明天 xx℃-xx℃，x 风 x 级，雾霾指数 xx
按日期查询天气	明天天气	深圳明天 xx℃-xx℃，x 风 x 级，雾霾指数 xx
按天气查询天气	下雪了吗？	没有
按日期+城市查询	明天上海天气	上海明天 xx℃-xx℃，x 风 x 级，雾霾指数 xx
按天气+日期查询天气	明天下雪吗	明天不下雪
按日期+城市+天气查询天气	明天深圳会下雨吗？	明天深圳不会下雨 明天深圳会下雨
询问后连续查询	那北京呢？	按上一条查询方式

3.3.5 新闻类用例分析

新闻类的主要功能包括：按新闻类查询、按事物查询、按时间查询。用例图如图 3-8 所示：

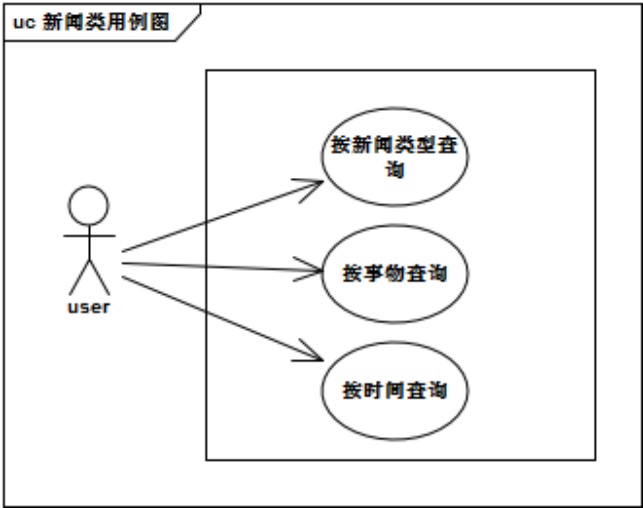


图 3-8 新闻类用例图

表达方法如表 3-4 所示：

表 3-4 新闻类表达方式表

功能名称	结果返回方式	例句
按新闻类型查询	播报该类型新闻	我想听国内新闻

表 3-4(续) 新闻类表达方式表

功能名称	结果返回方式	例句
按事物查询	播报根据该事物查询到的新闻	王宝强最近有什么新闻
按时间查询	播报两日内热门新闻	最近有什么新鲜事儿？

3.3.6 闹钟与提醒用例分析

闹钟与提醒的主要功能包括：按日期设定闹钟、按时间设定闹钟、按循环方式定闹钟、删除已设闹钟、更改已设定闹钟参数、设定提醒、更改提醒、删除已设定提醒。

用例图如图 3-9 所示：

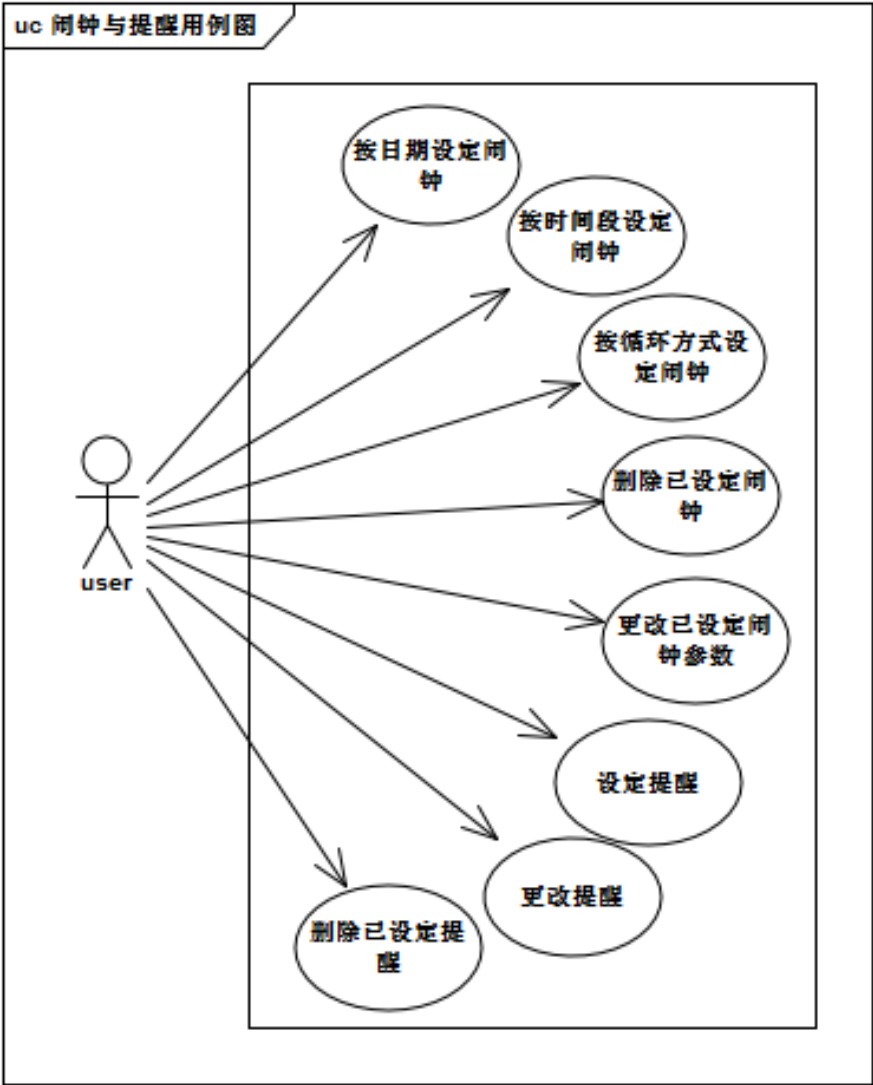


图 3-9 闹钟与提醒用例图

表达方式如表 3-5 所示：

表 3-5 闹钟与提醒表达方式表

功能名称	例句	应返回的参数	本项需处理的不同行为
按日期设定闹钟	定明天早上 8 点的闹钟	新建、明天、早上 8 点	
按时间段设定闹钟	定半小时后的闹钟	新建、今天、半小时后的时间点	
按循环方式定闹钟	定工作日早上 8 点的闹钟	新建、工作日、早上 8 点	工作日、每天、每周 X（每周 X 可有多 个）
删除已设定闹钟	取消工作日早上 8 点的闹钟 删除我的所有闹钟	删除、工作日、早上 8 点 删除、所有	删除
更改已设定闹钟参数	把工作日早上 8 点的闹钟后延半小时	更改、工作日早上 8 点的闹钟、延后半小时的时间点	延后、提前
设定提醒	十分钟后提醒我洗澡	十分钟、洗澡	时间、行为
更改提醒	把提醒后延 10 分钟	后延、十分钟	时间、修改行为
删除已设定提醒	不需要提醒我去洗澡了	删除、洗澡、提醒	

3.3.7 百科类用例分析

百科类为按名称查询返回的结果，用例图如图 3-10 所示。

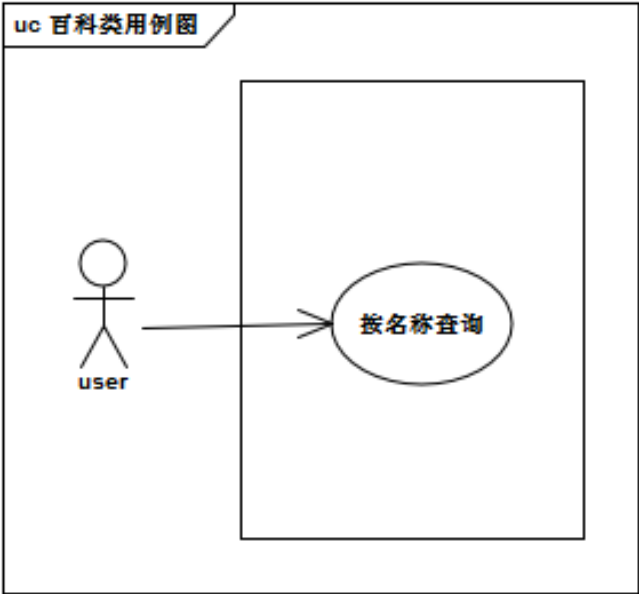


图 3-10 百科类用例图

表达方式如表 3-6 所示：

表 3-6 百科类表达方式表

功能名称	结果返回方式	例句
按名称查询	返回按关键词查询到的百度百科结果	谁是王菲

3.3.8 聊天类用例分析

聊天类为随意说一句话，当所有分类都处理不了的时候，返回聊天结果。用例图如图 3-11 所示。

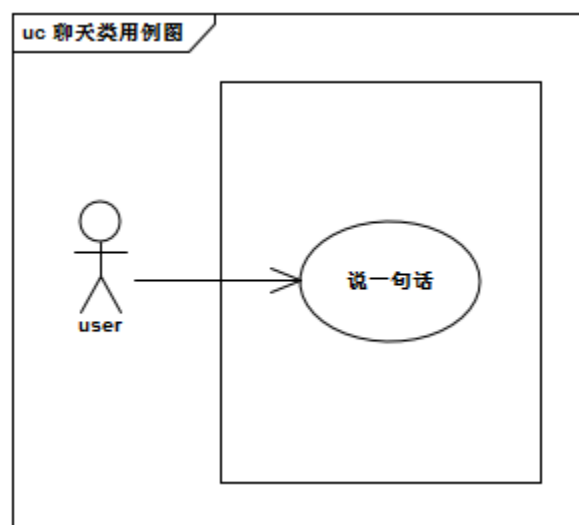


图 3-11 聊天类用例图

3.4 非功能性需求分析

系统除了考虑功能性需求外，还应该考虑非功能性需求。当非功能性需求被很好的满足，系统的功能将更能够被用户接受，同时也将更有利于系统未来的升级优化、维护，这往往能觉得整个项目的成败。下面从五个方面考虑非功能性需求。

系统实用性:系统除了有针对音箱控制的功能外，还有可以对日常工作可以进行辅助的功能。例如通过语音设置闹钟、询问天气、听新闻、百科等。同时还可以进行日常语音交互。在进行广泛的语音交互的过程中，语音控制系统要能够控制每一次语音交互的完整性，以及能够打断语音交互，和命令式退出语音交互。

系统可靠性:在语音控制系统运行过程中不应该发生系统故障，在语音交互过程中造成系统崩溃时，应能够进行系统恢复，并且不影响其他语音交互功能。

系统效率:在进行语音交互的时候，状态切换都是控制在秒的级别。唤醒后三秒内无输入给出提示，五秒内无输入再次给出提示，如果十秒内一直没有语音进行交互则退出

语音交互；当一次语音交互结束后，如果五秒内没有再次进行语音交互则退出语音交互。

系统可拓展性:当语音识别，自然语言处理的技术进一步提升后，语音交互将能够实现更多的功能，为了接入更多的语音交互功能，系统的可拓展性是必须的[3]。同时伴随着智能家居，物联网的进一步发展，在以后针对家居与家居的相连，公司进行功能调整的时候，都需要系统有良好的可拓展性。

用户友好性:语音交互是人与机器的交互，相比于人与人进行语言交流，机器如果不能及时反馈，用户体验将会非常不好。所以当系统出现错误，或者无法识别问题以进行语音交互的时候，系统都设置了特殊情况发生下的特定语音，以及特定交互 UI。同时根据测试结果和用户体验设置体验良好的语音交互流程。

3.5 本章小结

本章首先简单介绍了系统，然后对系统进行了模块划分，系统总体划分为三个大模块，大模块下面分别划分出了一些小模块。接下来分别从功能性需求、非功能性需求两个方面对系统进行了分析。

4 系统架构与设计

在上一章的基础上，读者已经对系统的功能比较了解了。本章将结合所有功能对语音控制系统进行系统架构与设计，下面从系统总体架构设计、系统接口设计两个方面进行论述。系统总体架构设计主要分析构成系统的模块及其关系，系统接口设计主要介绍模块与模块间是如何交互的。

4.1 系统总体架构设计

系统的整体流程是先由用户录入语音，然后由思必驰进行语音处理，把语音转文字并做初步理解，识别出里面的部分业务，如：语音无法理解、识别错误。然后根据处理完后的文本，拦截特定场景下的业务，处理后让音箱去执行相关命令。如果超出这个范围，这把文本交给 NLP 服务方，这么 NLP 服务方进行处理分为魅族 NLP 服务方和第三方 NLP 服务方，然后识别出其中的音乐，百科，有声，天气，聊天机器人，及其他业务，处理后让音箱去执行相关命令，至此完成一轮语音对话，如果用户要进行持续语音对话，就再次进行语音输入，完成第二轮对话，这样进行多轮循环。流程图如图 4-1 所示：

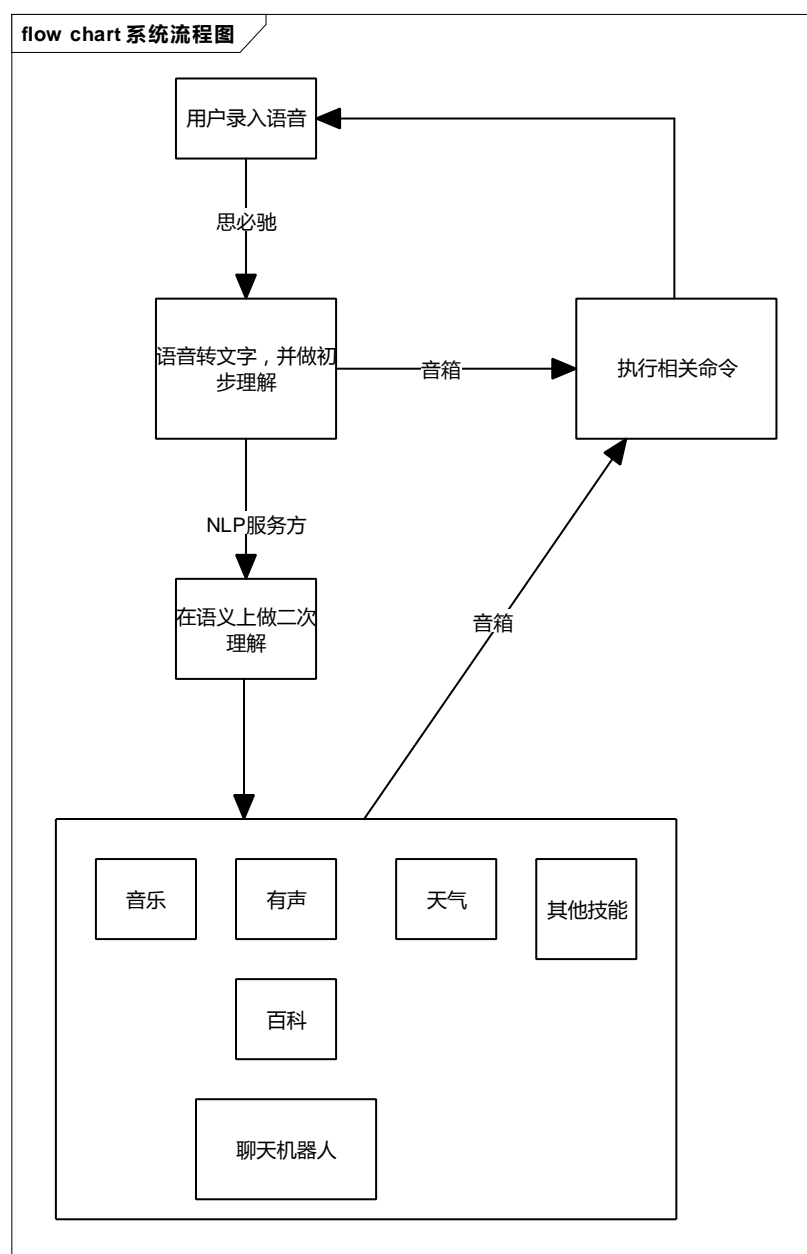


图 4-1 系统流程图

系统的实现由语音输入模块（唤醒、语音）、文本处理模块（特定场景模块、魅族 NLP 模块、第三方 NLP 模块）、业务逻辑处理模块（process）、交互处理模块（Android UI，语音合成）、命令下发模块五个模块组成。用户先进行语音唤醒，成功后音箱进入语音识别状态，当用户输入语音后进行语音识别，把识别后的文字进行文本处理，文本处理分作三块：特定场景模块、魅族 NLP 模块、第三方 NLP 模块。由这三块依次处理。处理完后把文本下发给 process 进行业务处理，处理完后再根据业务显示 Android UI 交互页面，同时给出提示音，或者语音合成播报动态内容。然后在根据具体业务把命令下发让音箱执行相应功能。具体架构设计如图 4-2 所示：

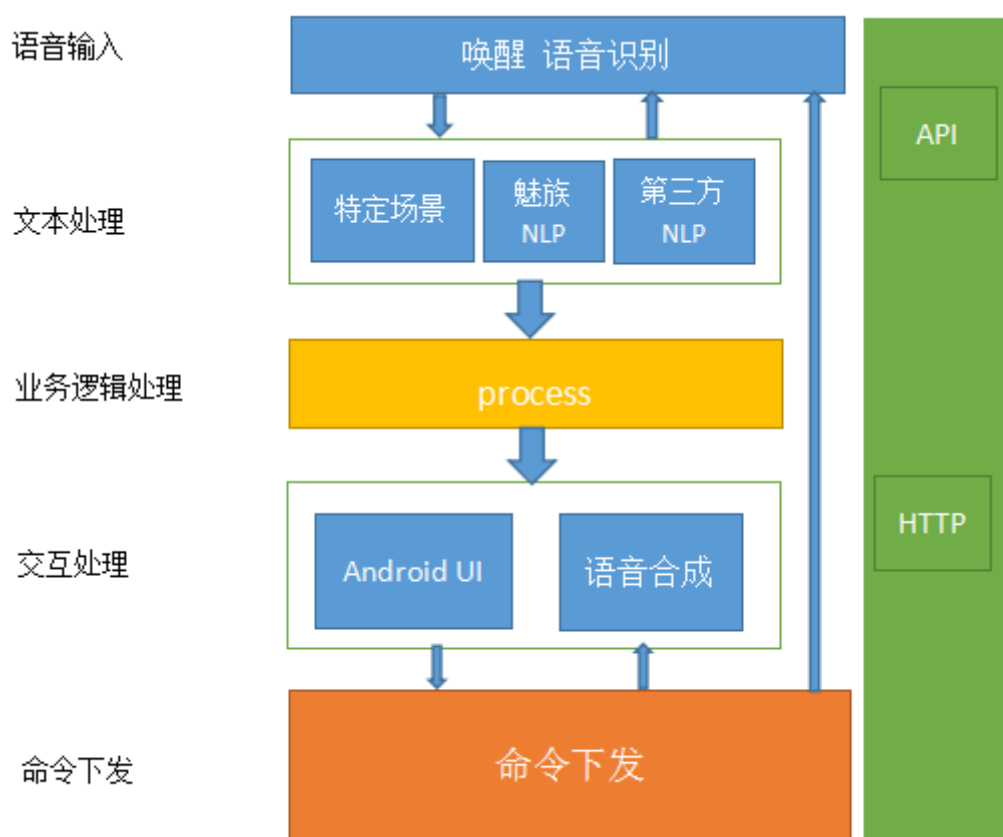


图 4-2 系统架构图

语音输入模块：该模块唤醒，语音识别两个子模块。唤醒指通过个某个特定的关键字让音箱从录音状态切换到语音识别状态。语音识别指在用户录入语音后，把语音转文本，然后把得到的结果 JSON 数据和状态码（0：成功；70904：等待语音录入超时；70914：识别错误；70910：网络超时）交给下个模块处理，同时在处理过程中关闭语音唤醒和语音识别。

文本处理模块：该模块从上一个模块传递过来的数据中提取出用户输入 input 部分数据，依次让特定场景，魅族 NLP，第三方 NLP 进行处理，只要有一个模块能够处理就停止传递，并返回处理后的状态值。

业务逻辑处理模块：对上一个模块传递下来的实际业务进行处理，如音乐要处理资源无效、资源有效、播完完成、播放打断等业务逻辑。在业务逻辑处理的同时，交互处理模块，命令下发模块也开始执行。

交互处理模块：根据不同的业务传递过来的类型值，显示不同的 UI 交互及语言提示。

命令下发模块：更加业务逻辑模块处理后的具体业务下发相应的命令，根据不同情况，在命令下发后可能调用交互处理模块，如：收藏、音量调节，会在命令操作成功后在进行交互处理；在命令下发后可能是对唤醒，语音识别的开启、关闭，如音乐播放，会把语音识别关闭，语音唤醒开启。

4.2 系统接口设计

这里主要介绍控制 process 监听接口、UI 交互和音箱命令执行的接口，process 监听接口是针对 process 的处理状态控制唤醒，以及语音识别的开关，控制 UI 交互的接口是针对用户语音交互过程中 UI 状态以及提示音的切换，音箱命令执行的接口是针对音箱功能的控制。

4.2.1 process 监听接口

接口设计如图 4-2 所示：

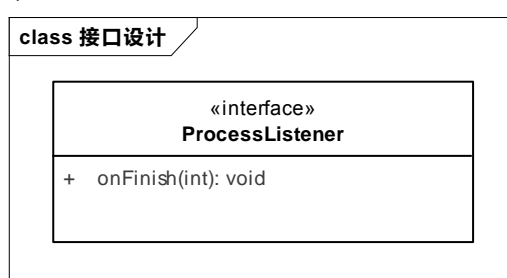


图 4-3 process 接口设计图

方法参数的值如表 4-1 所示：

表 4-1 process 接口方法参数表

Code 值	描述
-1	关闭交互界面，语音识别
0	关闭语音唤醒，语音识别继续
2	结束语音控制
3	切换音箱状态到工作者状态
4	切换音箱状态播放新闻

4.2.2 UI 交互接口

接口设计如图 4-2 所示：

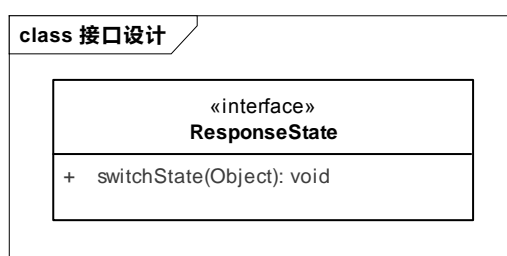


图 4-2 UI 交互接口设计图

方法参数值如表 4-2 所示：

表 4-2 UI 交互接口方法参数表

Param 值	描述
ResponseStatus.RESPONSE_STATUS_WAKE_UP	唤醒
ResponseStatus.RESPONSE_STATUS_WAKE_UP_AGAIN	再次唤醒
ResponseStatus.RESPONSE_STATUS_END_COMMUNICATE	结束对话
ResponseStatus.RESPONSE_STATUS_WAKEUP_CANNOT_UNDERSTAND	无法理解
ResponseStatus.RESPONSE_STATUS_ERROR_HANDLING	错误处理
ResponseStatus.RESPONSE_STATUS_PLAYING	播放
ResponseStatus.RESPONSE_STATUS_HUMAN_SERVE	人口客服
ResponseStatus.RESPONSE_STATUS_TIMING	提醒
ResponseStatus.RESPONSE_STATUS_ARM_CLOCK	闹钟
ResponseStatus.RESPONSE_STATUS_COMMUNICATE_END_WAITING_INPUT	对话结束 等待再次 输入
ResponseStatus.RESPONSE_STATUS_BREAK,	打断
ResponseStatus.RESPONSE_STATUS_ASR_INPUT	识别结果
ResponseStatus.RESPONSE_STATUS_TEST,	测试
ResponseStatus.RESPONSE_STATUS_COLLECTION	收藏
ResponseStatus.RESPONSE_STATUS_SHUTDOWN	关闭
ResponseStatus.RESPONSE_STATUS_NETWORK_TIMEOUT	网络超时
ResponseStatus.RESPONSE_STATUS_CENTER_CONTROL_UNPROCESSED	中控
ResponseStatus.RESPONSE_STATUS_CAN_NOT_HANDLE_COMMAND	命令无法 处理
ResponseStatus.RESPONSE_STATUS_NETWORK_USELESS	网络不可 用

4.2.3 音箱命令执行接口

接口设计如图 4-3 所示：



图 4-3 音箱命令执行接口设计图

音箱控制部分协议如表 4-3 所示：

表 4-3 音箱控制协议表

action	说明	请求方式	携带的参数
Info	获取音箱的设备信息	Get	无
Status	获取音箱的播放状态	Get	无
SetPlayMode	设置播放模式（乱序，随机，顺序）	Post	{ playMode:Int }
Pause	请求音箱暂停播放	Get	无
Resume	请求音箱继续播放	Get	无
Prev	请求音箱播放下一首歌曲	Get	无
SetVolume	设置音箱音量	Post	{ currentVolume:Int }
GetVolume	获取音箱音量	Get	无
Seek	跳转到当前播放的曲目的某个时刻	Post	{ position:Int }
SetAlarm	设置闹钟	Post	{ "alarmList": [{ "time":String "mode":Int "dayList":[int] "playList":PlayList "volume":Int "enable":Boolean },] }
GetAlarm	获取音箱端所有闹钟列表	Get	无
DeleteCurrentSong	取消歌曲收藏	Post	{ track:Track }
Suspend	定时关机	Post	{ time:int }

4.3 本章小结

本章的内容为系统架构与设计，主要讲述了系统总体架构设计、系统接口设计。架构设计讲述了模块间的关系和逻辑，接口设计讲述了模块的交互方式。本章内容是建立在第三章需求分析的基础上，同时为下一章模块设计与实现做好了铺垫。

5 模块设计与实现

本章对每个模块的详细设计进行分析，从不同的角度对详细设计进行分析。每一个详细设计都由设计描述和 UML [12] 图组成，UML 图包括流程图、类图。设计描述，通过功能分析简单描述该实现方式；流程图，从用户角度分析功能的实现流程；类图，从类与类之间的调用关系及接口设计方面分析具体实现方式。

5.1 特定场景模块

5.1.1 设计描述

特定场景模块主要用于对音箱功能的控制，以及增加用户友好性，该模块包括：歌曲切换、音量调节、收藏、播放暂停、回家出门等技能。特定场景在关键字在系统启动的时候先加载进来。在有的场景下文本就等于关键字时触发特定场景，如暂停关键字：静音、暂停、停止播放、暂停播放、闭嘴，结束对话关键字：再见、拜拜、没事儿了、没有了、没了、你退下吧。在有的场景下包含特定场景关键字时触发特定场景，如频道切换关键字：切、换、播放，同时还必须包含“频道”字符。

关键字设置如表 5-1 所示：

表 5-1 特定场景关键字表

类型	内容
Pause	静音，暂停，停止播放，暂停播放，闭嘴
Scenarios_LeaveHome	上班去了，上班去啦，上班了，出门了，出门啦，我上班去了，我上班去啦，我上班了，我上班啦，我出去了，我出去啦
Scenarios_GetHome	回家了，回家啦，到家了，回来了，我回家了，我回家啦，我到家了，我到家啦，我回来了，我下班啦，下班了，下班啦
Music_Play	我要听，我想听，播放，给我播放，为我播放，搜索
Close	关上，关闭，关机，休眠
Next	下一首，下一集，下一回，下首歌，切歌，换首歌听，再来一首，换首歌，换一个，换一首
Prev	上一首，上一曲，上一个，上一集，上一回
Next_Ch	下个频道，下一频道，下一个频道
Prev_Ch	上一频道，上个频道，上一个频道
Volume_Up	声音大点，大点声，大声点，声音大一点，大声一点

表 5-1(续) 特定场景关键字表

类型	内容
Volume_Down	声音小点，声音小一点，小一点，小声点，小声一点
End_Dialog	再见，拜拜，没事儿了，没有了，没了，你退下吧
Collection_Song	收藏，收藏这首歌，我喜欢这首歌，我收藏这首歌，喜欢这首歌
Collection_Song_Cancel	取消收藏，取消收藏这首歌，不收藏这首歌了
Collection_Album	收藏专辑，收藏这个专辑，帮我收藏这个专辑
Collection_Album_Cancel	取消收藏专辑，取消收藏这个专辑，不收藏这个专辑

5.1.2 流程图

当用户语音输入后，进行语音识别，对转换成的文本后进行关键字过滤。当特定场景触发后则把显示相应的交互 UI 和提示音，并且把命令下发给音箱，执行相应的音箱功能。具体流程图如图 5-1 所示：

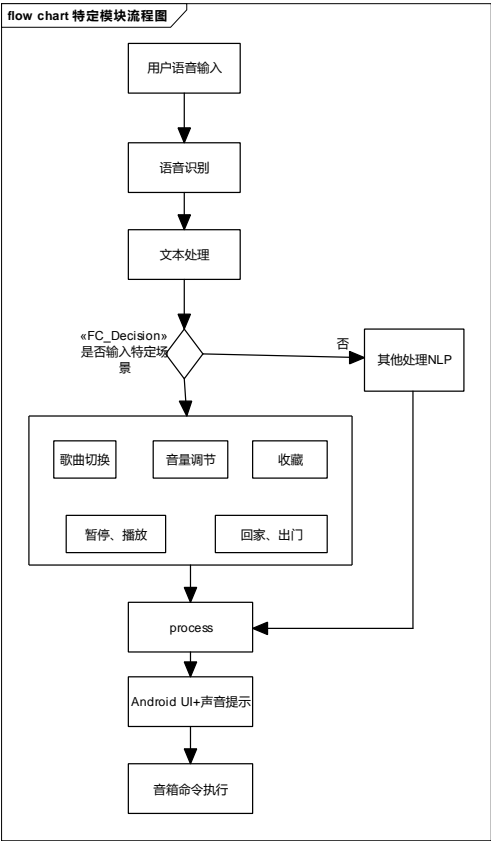


图 5-1 特定场景流程图

5.1.3 类图

下面是特定场景模块的核心类图。

SpeechWakeupEngineOnWakeupListenerImpl 类：监听语音识别，并取得结果，然后对结果进行前期处理，以找出当前用户说的一句话属于哪个模块。

KeyWordsFetcher 类：用来加载特定场景的关键字

KeyWords 类：关键字抽象类，这个类定义了判断用户语音转文本后是否等于特定场景关键字的 equalsKeyWord() 方法，以及是否包含特定场景关键字的 containsKeyWord() 方法。该类定义的抽象方法 doHandle() 用于具体的特定场景继承 KeyWords 类后自己去实现关于特定场景的一些业务，如：命令下发、显示 UI 交互。

ActionManager 类：用于控制一些 action 使用，如：语音合成播放文本、语音识别的开启关闭、语音唤醒的开启关闭、UI 交互。

ResponseState 接口：状态响应接口，该接口定义了状态切换 switchState() 方法。具体的业务由实现该接口的状态类去实现。具体类图如图 5-2 所示：

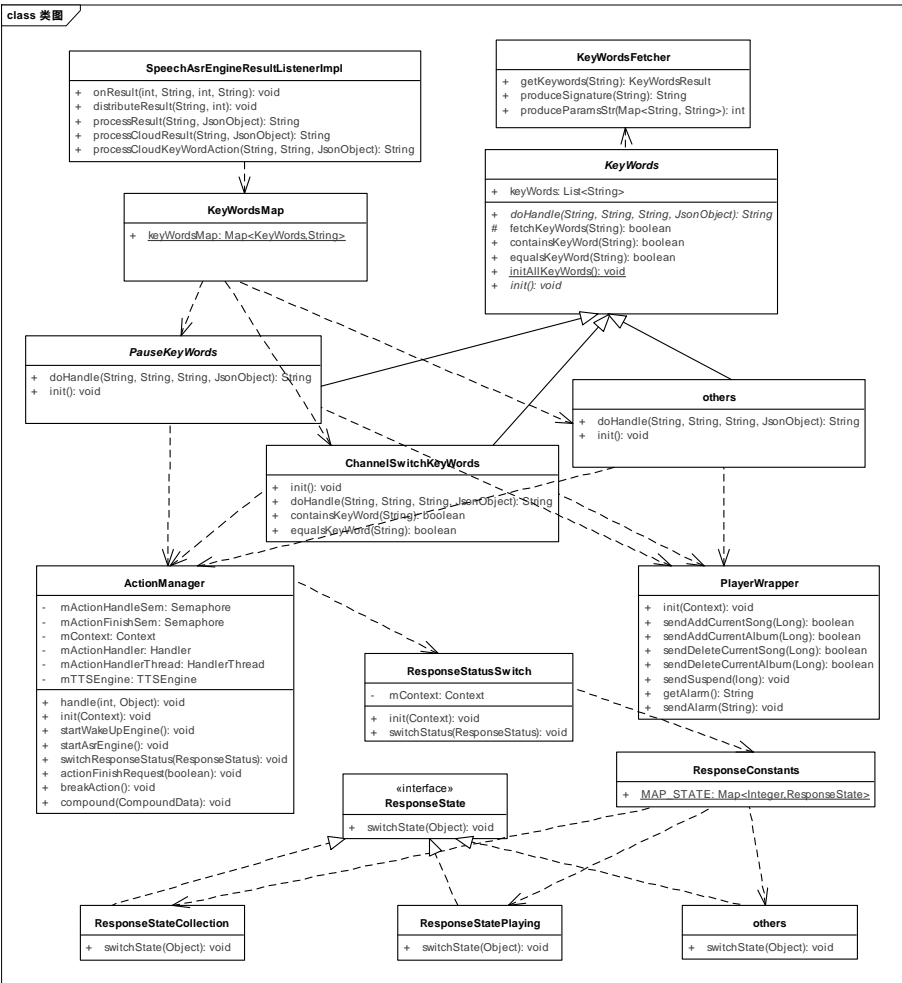


图 5-2 特定场景类图

5.2 魅族 NLP 模块

5.2.1 新闻类

5.2.1.1 设计描述

该模块的新闻内容包括：军事，财经，体育，娱乐，八卦，科技，国外，国内，社会，热点，热门，魅族新闻。这些新闻内容为魅族编辑过的内容，根据 NLP 后进行搜索，搜索到直接播放，没有则交由其他功能进行处理。

5.2.1.2 流程图

用户输入的语音转文本后，判断是否属于特定场景，如果不属于再判断是否包含“新闻”关键字，如果是则对该条文本进行魅族自己定义的 NLP，否则交由其他 NLP。魅族 NLP 主要是通过模板方法去掉新闻查询表达方式里的动词，如：听、播、来一段等，同时去掉结尾的“的新闻|地新闻”字符，然后在分组提取出里面的关键字，得到 NLP 结果，NLP 结果包含频道、关键字、收录关键字以及原问题，然后根据结果去查询魅族编辑过的新闻内容。如果查询到新闻，把查询的所有新闻下发给音箱进行语音播报并显示交互 UI。具体流程如图 5-3 所示：

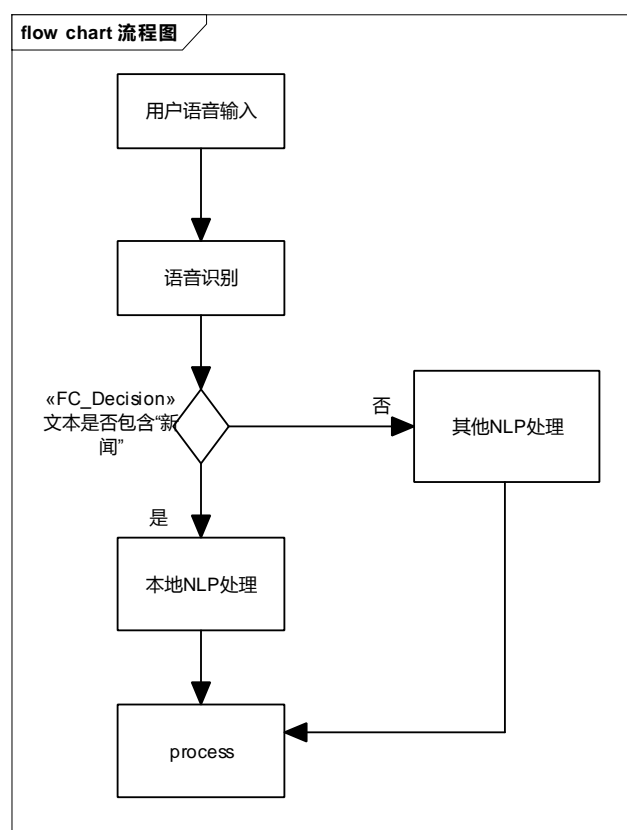


图 5-3 魅族 NLP——新闻类流程图

5.2.1.3 类图

下面特定场景模块新闻类的核心类图。

SpeechAsrEngineResultListenerImpl 类：其中的 onResult() 方法用来处理语音识别后的结果，distributeResult() 方法用于分发结果。processCloudResult() 方法获取相应模块的处理结果。

MZNLPQuery 类：用于处理魅族 NLP 模块。

NLPQuery 类：用于获取 NLP 结果，结果以 MZResult 对象的形式返回。

NewsAnnouncer 类：用于处理 NLP 结果，把每个 News 对象中的内容，标题，URL 提取出来，放进 Content 中，然后把 list<data>下发给音箱进行语音播报，并改变 UI 状态。

SpeechManager 类：用于控制 UI 状态个改变，如：关闭、启动、维持不变。具体类图如图 5-4 所示：

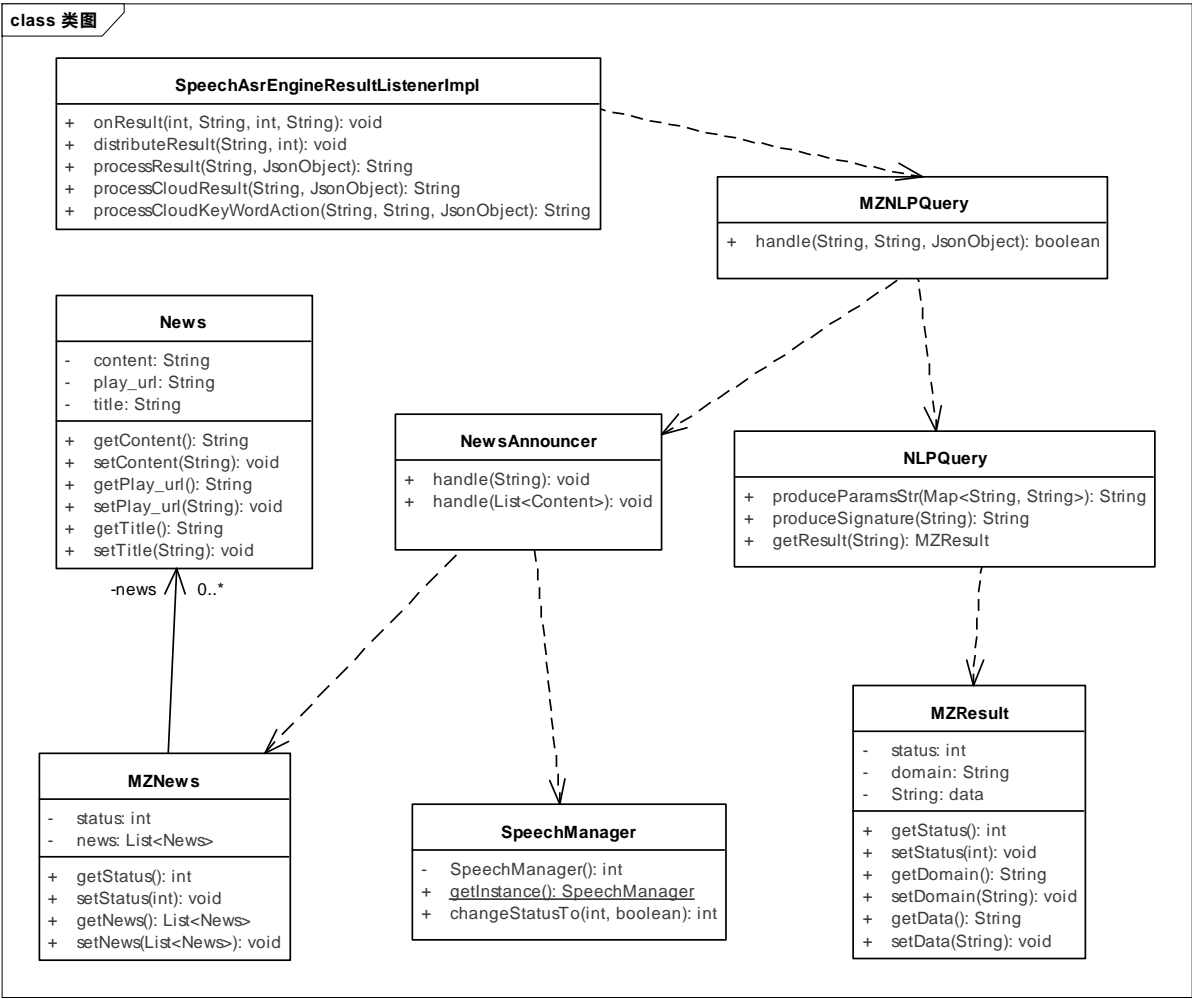


图 5-4 魅族 NLP——新闻类类图

5.3 第三方 NLP 模块

5.3.1 音乐类

5.3.1.1 设计描述

音乐类作为音箱的最主要功能，当用户语音中包含音乐类意图时，需要详细分析属哪一类点播方式，处理后实现相应的功能。同时在需要考虑各种非正常情况的备用方案，如资源不可用，识别错误，搜索不到资源等。

5.3.1.2 流程图

NLP 处理方根据句义识别用户是否有点播意图，意图是否可直接执行：如果用户录入的内容，视为命中了定义的音乐类点播格式，且程序在检索后发现有效果，则视为用户的这条意图可以直接执行；如果用户录入的内容，不能够准确地命中音乐类点播格式，或命中了点播格式，但搜索结果无可用内容，则需要做下一层判断。

若意图可直接执行，则由 NLP 服务方返回相应结果到魅族，魅族音箱播放器按结果中的歌曲 ID 拉取播放地址信息进行播放。

若意图不能执行，则对用户意图做分词，拆分出能够命中相关音乐类词库的关键词，有相应关键词，则以对话方式引导用户接受推荐，用户返回肯定意图，则返回推荐结果到音箱；用户不接受推荐，则结束对话，由音箱询问用户是否有新要求。

关键词：关键词指的是对于音乐类需求典型关键词的词库，如：歌手名词库、歌曲名词库、风格名称词库等。具体流程图如图 5-5 所示：

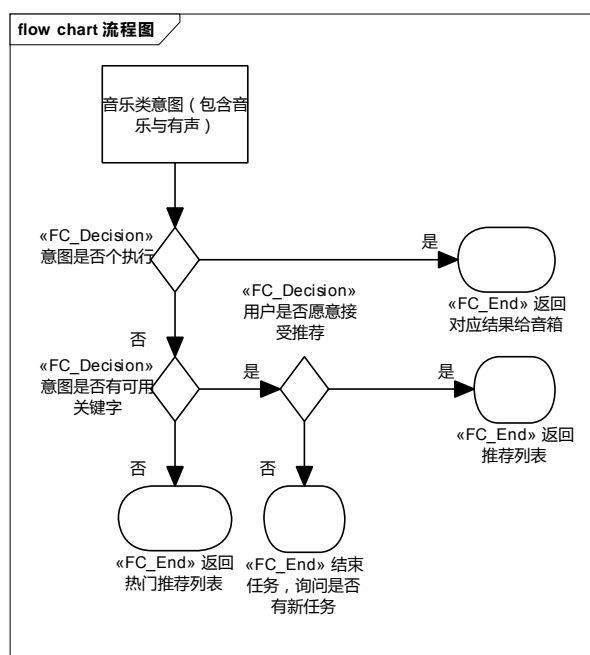


图 5-5 音乐类流程图

5.3.1.3 类图

下面是音乐类核心类图。

RuyiAi 类：用于获取第三方 NLP 结果，sendMessage() 方法用于把用户语音识别后的文本发送给第三方，然后把得到 JSON 字符串序列化为 RuyiIntentsResult 对象。

HttpHelper 类：通过 HTTP 获取资源。其中 sendGet() 方法用于 get 方式的请求，sendPost() 方法用于 post 方式的请求。

RuyiIntentsResult 类：第三方 NLP 返回的 JSON 结果的序列化对象。分作四层，从 RuyiIntentsResult 到 Result 到 Intent 到 Output。

RuyiMusicResult 类：用于封装从 RuyiIntentsResult 类的对象中提取出关于音乐的信息。

ProcessListener 接口：用于当第 NLP 结果处理完后回调对语音唤醒、语音识别的控制。具体类图如图 5-6 所示：

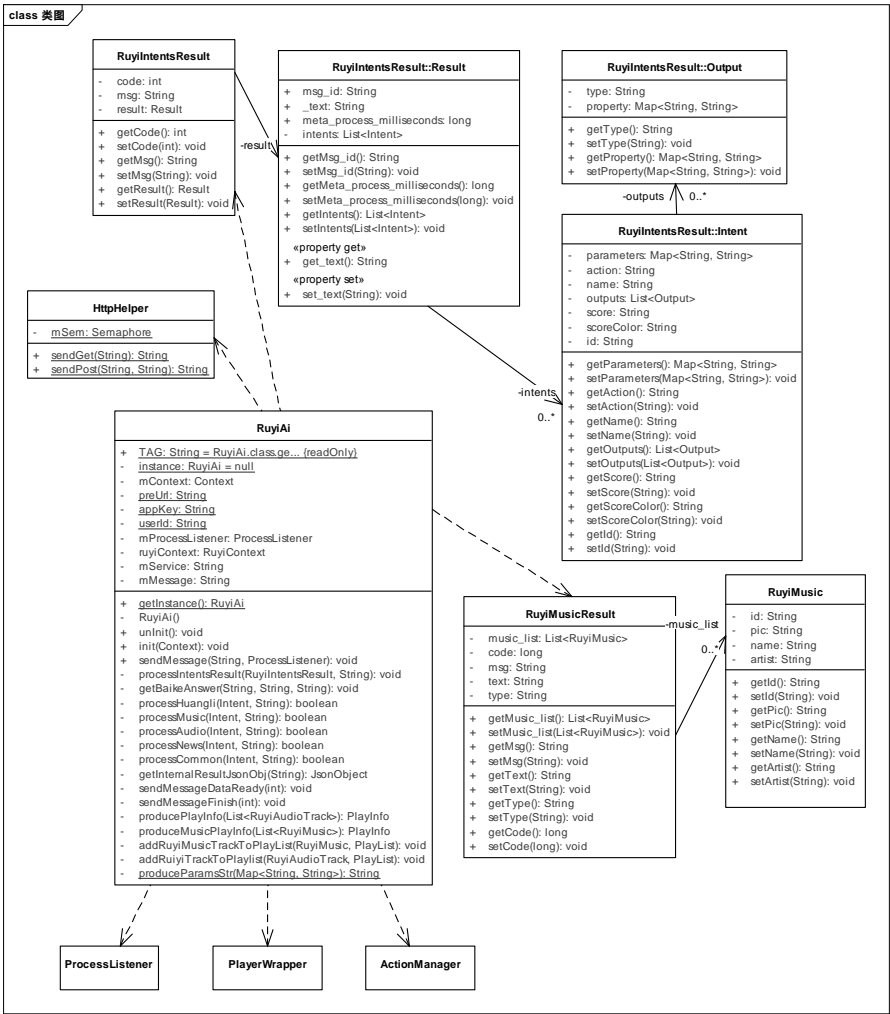


图 5-6 音乐类类图

5.3.2 有声类

5.3.2.1 设计描述

有声类为音箱主要功能之一，当用户语音中包含有声类意图时，需要详细分析属于哪一类点播方式，然后实现功能。同时需要考虑各种非正常情况的备用方案，如：资源不可用、识别错误、搜索不到资源等。

5.3.2.2 流程图

NLP 处理方根据句义识别用户是否有点播意图。意图是否可直接执行：如果用户录入的内容，视为命中了定义的有声类点播格式，且程序在检索后发现有效果，则视为用户的这条意图可以直接执行；如果用户录入的内容，不能够准确地命中有声类点播格式，或命中了点播格式，但搜索结果无可用内容，则需要做下一层判断

若意图可直接执行：则由 NLP 服务方返回相应结果到魅族，魅族音箱播放器按结果中的有声节目 ID 拉取播放地址信息进行播放。

若意图不能执行，则对用户意图做分词，拆分出能够命中相关有声类词库的关键词，有相应关键词，则以对话方式引导用户接受推荐，用户返回肯定意图，则返回推荐结果到音箱；用户不接受推荐，则结束对话，由音箱询问用户是否有新要求。

关键词：关键词指的是对于有声类需求典型关键词的词库，如：声优名词库、有声节目名词库、风格名称词库。具体流程图如图 5-7 所示：

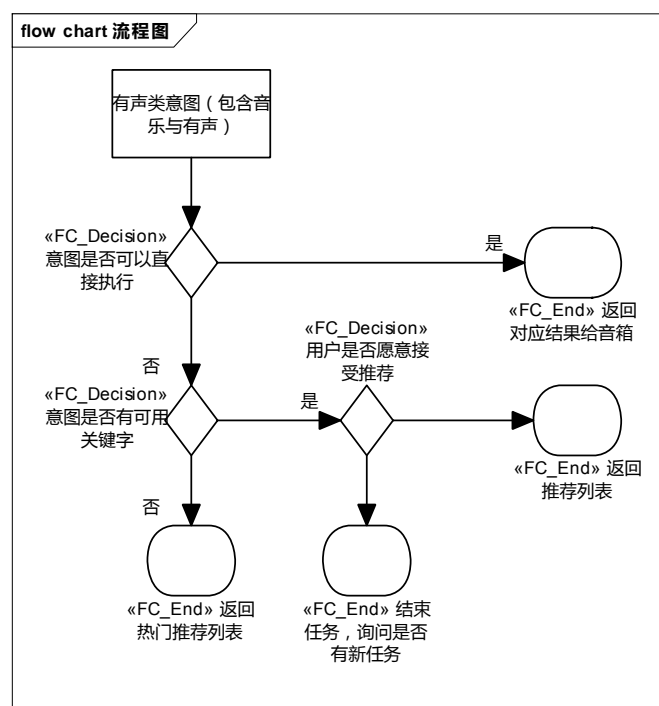


图 5-7 有声类流程图

5.3.2.3 类图

下面是有声类的核心类图。

RuyiAudioResult 类：用于封装从 RuyiIntentsResult 类的对象中提取出关于有声的信息。

PlayInfo 类：用于封装对 RuyiAudioResult 类对象内容处理后的数据的封装。

RuyiAi 类：该类中 processAudio() 方法用于处理有声类业务，如：命令下发、UI 交互，资源不可用 404 处理。具体类图如图 5-8 所示：

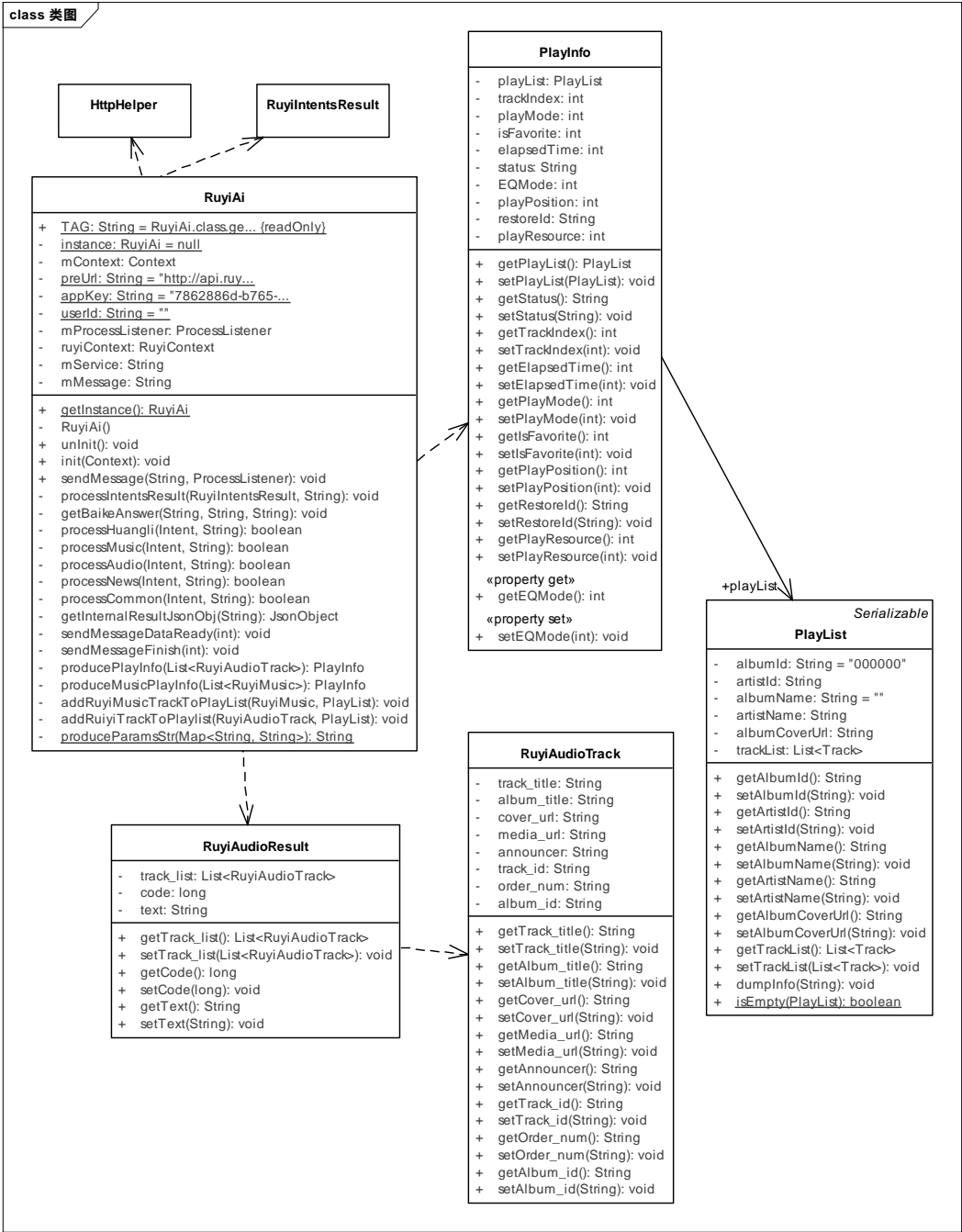


图 5-8 有声类类图

5.3.3 天气类

5.3.3.1 设计描述

默认的表达方式可以理解为日期+城市+天气，如果用户的表达里没有城市、日期，则默认为今天、当前城市。

天气的表达里，“天气”视为全面的天气预报，“晴风雨雪雾霾”等视为天气特征的表达。对于问句里问详细气象的意图，只返回天气结果里那一段的结果。例如：

明天下雪吗？→明天不下雪

明天雾霾→明天北京雾霾指数为 219，建议您务必戴上口罩

明天天气→明天北京天气晴，北风 2-3 级，气温-3—8℃，雾霾指数为 23，请注意保暖

特别要求：

当用户按天气查询天气时，若查询的天气可能引起气温变化，则在得到否定答复时，只回答是与否；得到肯定答复时，回答是与否，并同时返回查询的天气、气温的报告，以及人性化提示。例如：

Q：北京明天下雪吗？

A：不会下雪的

A：会下雪哦！北京明天小雪转大雪，-3℃~ -10℃，建议您多穿衣服，尽量少出门

返回结果的要求：

对于天气的返回结果，要求尽量人性化，做一些温暖的提示，例如：

Q：北京明天下雪吗？

A：会下雪哦！北京明天小雨转大雪，-3℃~ -10℃，建议您多穿衣服，尽量少出门

Q：北京明天多少度？

A：北京明天 14 摄氏度，微风，有轻微雾霾，建议您穿上外套，戴上口罩

对于返回结果的文案，日期描述的结果，按用户说法返回。举例说明：

问：北京明天下雨吗？

答：北京明天不下雨

问：12 月 21 号（日）上海下雨吗？

答：12 月 21 号（日）上海会下（大）雨，请记得带伞

5.3.3.2 流程图

当用户进行天气查询时，识别正确后，从返回结果中提取内容进行处理，处理后进行动态进行语音播报和 UI 交互，如果识别错误，则由其他处理方式进行处理。具体流程图如图 5-9 所示：

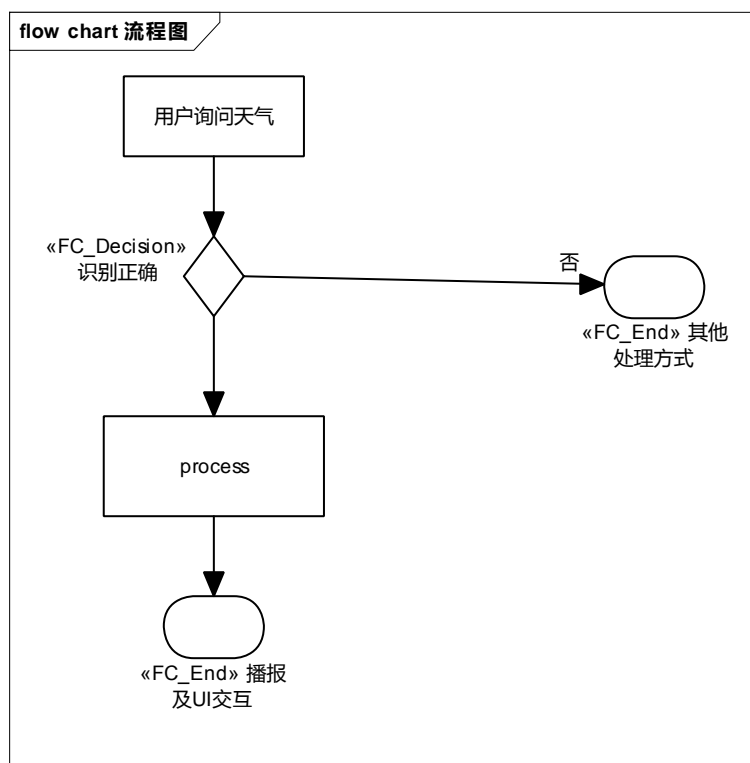


图 5-9 天气类流程图

5.3.3.3 类图

下面是天气类的核心类图。

RuyiAi 类：把第三方 NLP 返回的结果提取、处理后封装成 CompoundData 对象，然后下发给 ActionManager 类。

CompoundData 类：包含用于动态语音合成的数据 compoundStr 和回调类 ProcessListener。

ActionManager 类：用于调用语音合成引擎及交互 UI。

ActionHandler 类：用于辅助 ActionManager 类异步调用语音合成引擎。

TTSEngine 接口：语音合成接口，定义了开始合成 startCompound() 方法，合成 compound 方法，是否在合成 isCompounding() 方法，停止合成 stopCompoud() 方法，合成完成 completeCompound() 方法以及回调接口 TTSCompoundListener、TTSEngineInitListener，具体是逻辑由实现类自行实现。

BaiduTTSEngine 类：具体的语音动态合成引擎，实现 TTSEngine 接口并提供回调方式。具体类图如图 5-10 所示：

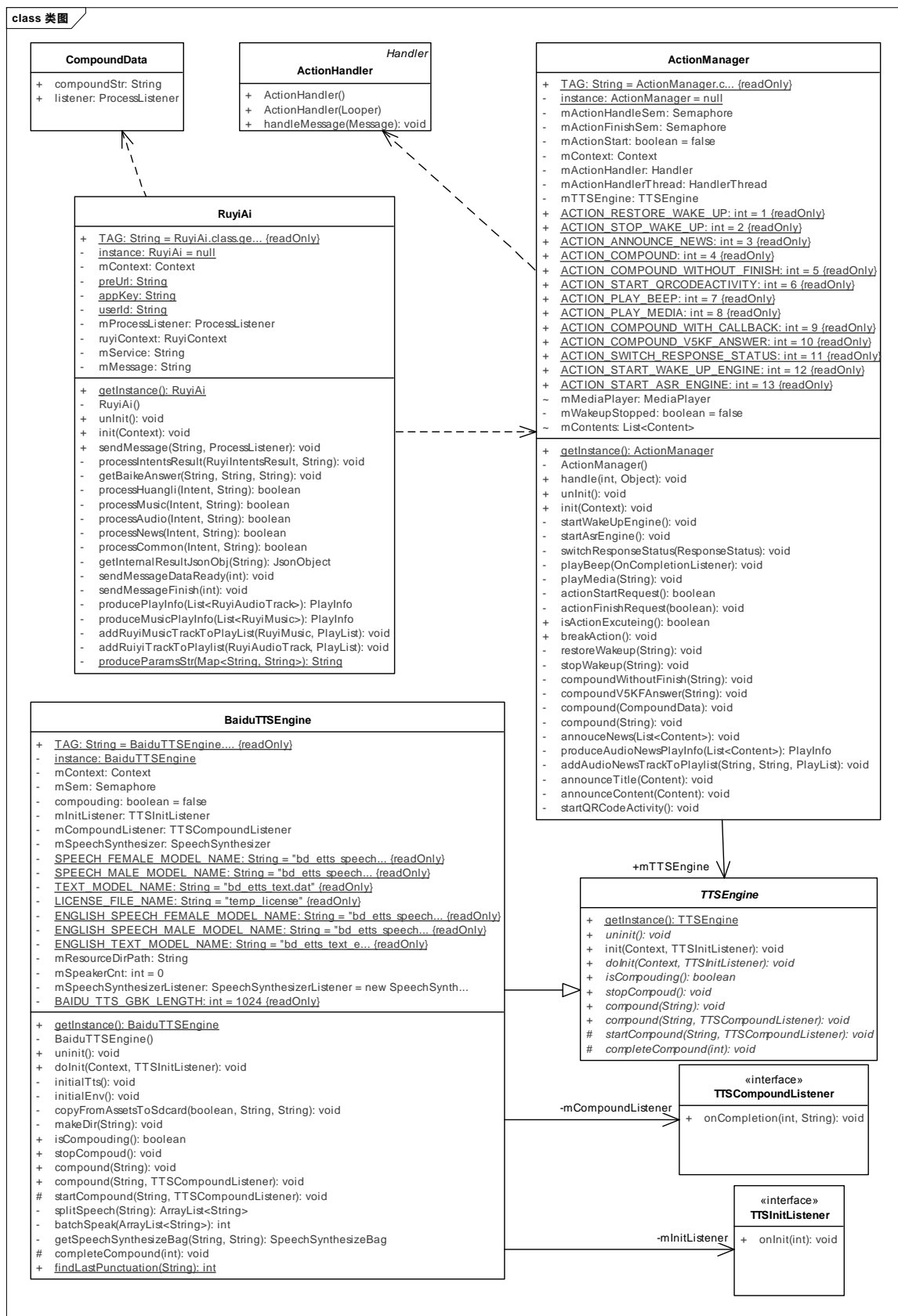


图 5-10 天气类类图

5.3.4 新闻类

5.3.4.1 设计描述

固定类型的新闻，如：科技新闻、国内新闻、娱乐新闻，需要返回【人声阅读】的音频文件，发送给音箱直接可以播放，以保证视听感受。

按事物查询的新闻，需要走网络查询的接口，查找到对应的新闻，并需要保持新闻长度在 200 字以内。对于源数据少于 200 字的，直接播放；对于源数据多于 200 字的，需要截取前 200 字，若抓取的新闻源有概要类信息，则直接播放概要。

按时间查询的新闻，需要返回【人声阅读】的新闻里，2 日内的热门新闻。

5.3.4.2 流程图

当用户询问新闻且由第三方 NLP 进行处理时。更加返回结果，首先判断是否是有声类新闻，如果是，直接播报。否则进一步提取出返回的文本内容，判断是否包含新闻内容，如果是，处理后播报。否则交由其他功能处理。在播报的同时需要显示相应的 UI。具体流程图如图 5-11 所示：

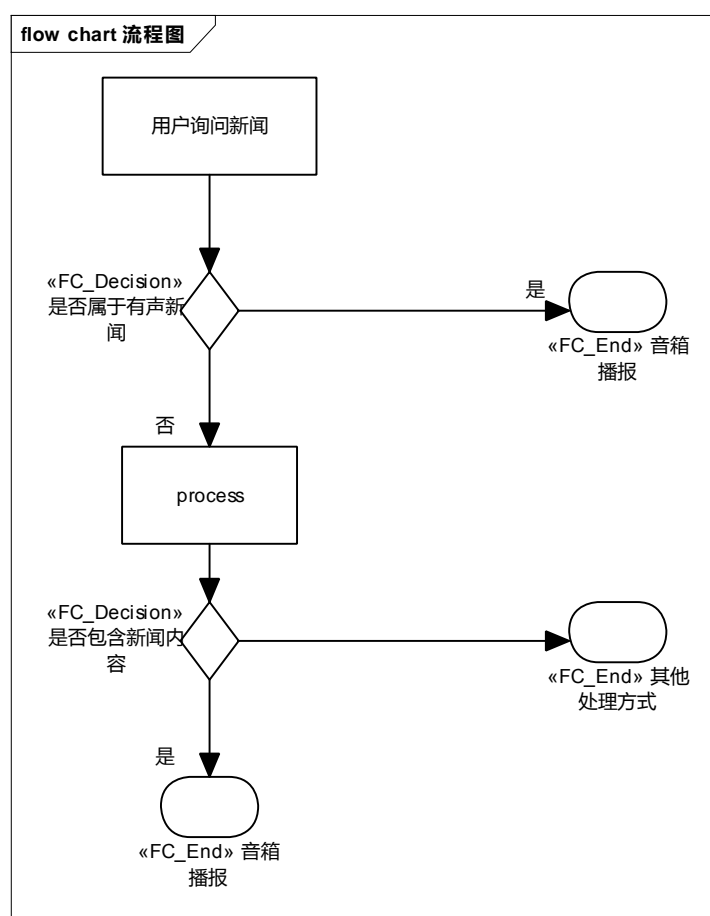


图 5-11 第三方 NLP——新闻类流程图

5.3.4.3 类图

下面是新闻类的核心类图。

RuyiAi 类：该类的 processIntentsResult() 方法对第三方 NLP 返回结果进行前期处理，判断是有声类新闻还是文本新闻，如果是有声类新闻则按有声类处理，否则按文本新闻处理。processNews() 方法用来处理文本新闻，从第三方 NLP 返回的 JSON 数据中提出标题、描述，然后传递给 NewsAnnouncer 类处理，播放新闻和显示 UI 交互。

Content 类：用来封装从第三方 NLP 结果中提取处理的数据。

ProcessListener 接口：用来在新闻处理完回调对语音唤醒，语音识别的控制。具体类图如图 5-12 所示：

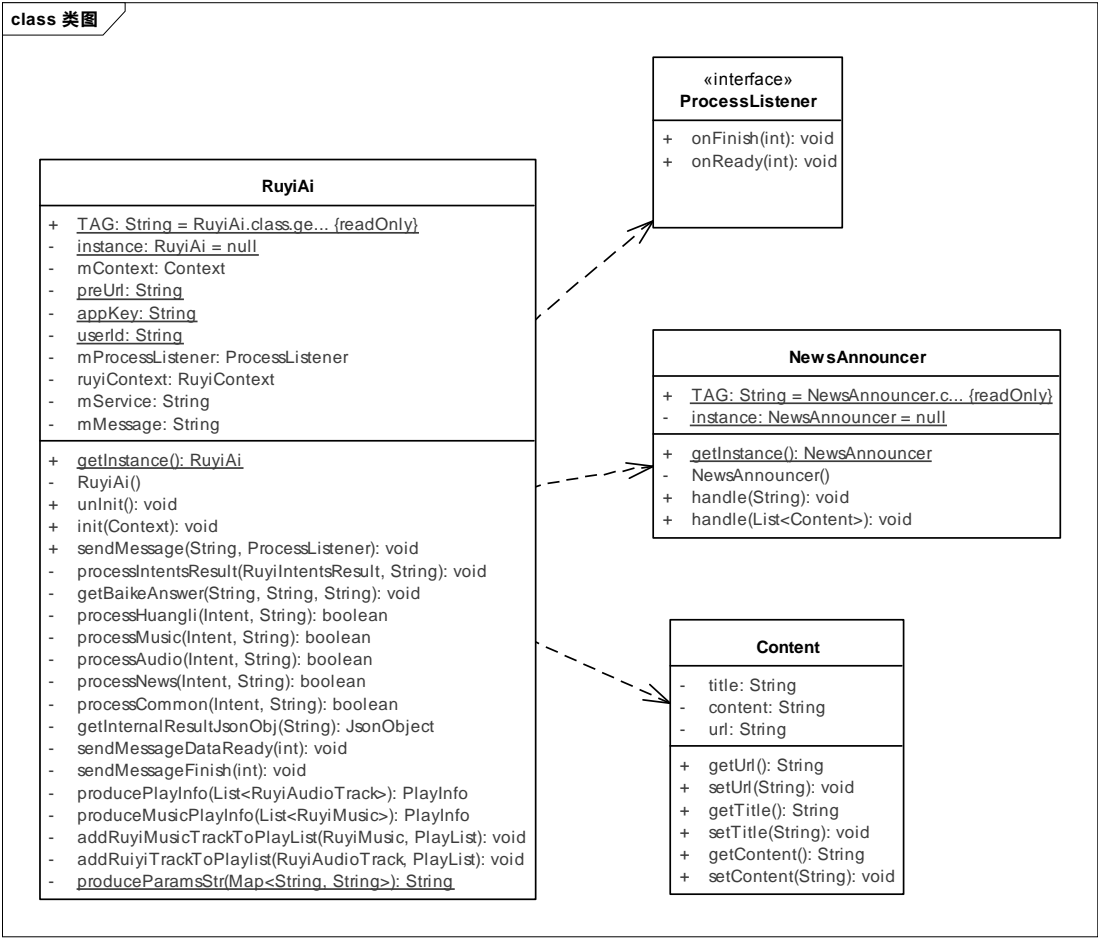


图 5-12 第三方 NLP——新闻类类图

5.3.5 闹钟与提醒类

5.3.5.1 设计描述

按照用户不同的闹钟需求，返回不同的参数，由音箱自行在闹钟上做不同行为。主要有新建、删除、更改三类功能。NLP 服务方需要覆盖用户不同意图的说法，比如新建闹钟的说法有“定、设定、设置、添加、设”。

未确定循环方式的闹钟，默认循环方式为只响 1 次允许用户更改已设定闹钟参数，包含循环方式、时间点、日期。

在定义产品时，魅族倾向于认为闹钟功能是【有规律地面向未来做日程规划】，即认为闹钟功能一般是未长期的未来服务，并且有一定循环规律；魅族倾向于认为提醒功能是【即时性地面向短期未来做规划】，即认为提醒功能面向的是短暂的未来。虽然不排除用户会有制定长期可循环提醒、短期闹钟的 case，但多数情况下只为核心定义考虑。

闹钟与提醒，不需要用户复核任务是否完成。在音箱产品上体现时，当用户打断了闹钟或提醒事件，就视为服务已经到位——用户已经感知到音箱的提醒了。因此，对于闹钟和提醒功能的管理只需要魅族记录、更新状态，不需要向云端汇报。

按照用户输入的提醒内容，拆分出时间点、用户意图，返回时间参数、具体时间、答复用户的文字。

5.3.5.2 流程图

当用户要求设置闹钟或者提醒时，如果识别正确，则提取出返回结果中的时间进行处理，如：修改、增加、删除，然后根据类型分别设置。设置完毕后音箱需要进行语音播报结果，并在 UI 上显示交互内容。具体流程图如图 5-13 所示：

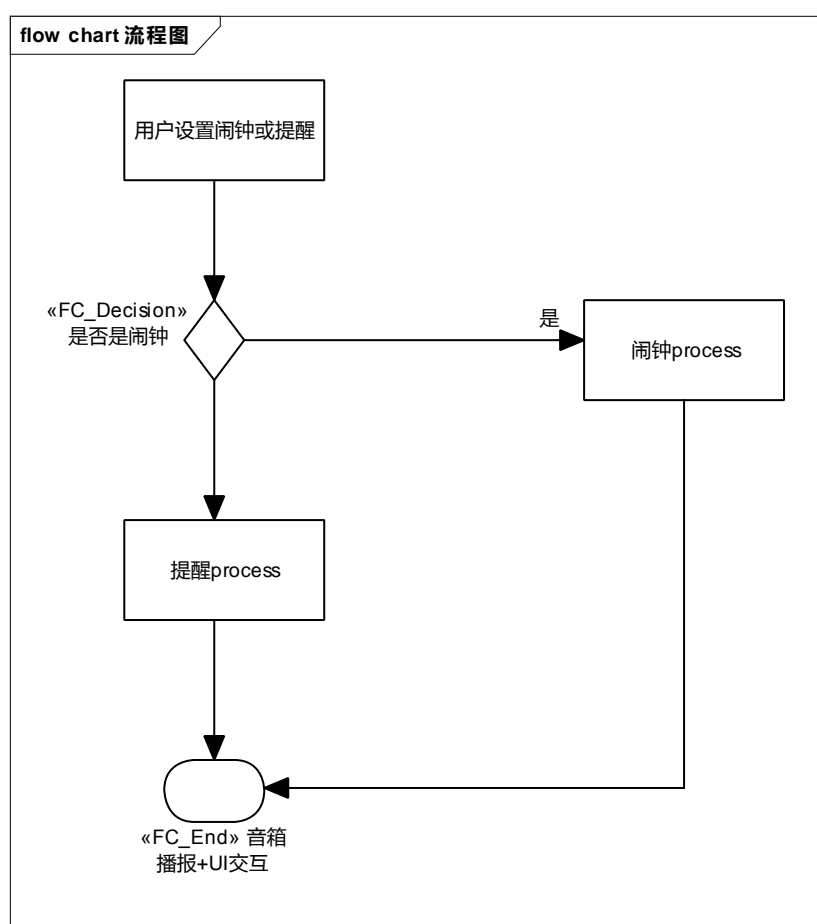


图 5-13 闹钟与提醒类流程图

5.3.5.3 类图

下面闹钟与提醒类核心类图。

RuyiAi 类：其中 processTime（）方法对第三方 NLP 结果进行前期处理，然后传递给 TimeResultProcessor 类进行处理。

TimeResultProcessor 类：对传递进来的数据进行进一步处理然，如果属于闹钟则传递给 ClockProcessor 类进行处理，如果属于提醒类则传递给 RemindProcessor 类进行处理。

ClockProcessor 类：对闹钟进行处理，提取出闹钟的具体时间，同时把所有已设置的闹钟从后台取出来进行操作，完成后设置闹钟并显示 UI 交互和提示音。

RemindProcessor 类：对提醒进行处理，提取从提醒的具体时间然后下发给音箱，同时显示交互的 UI 和提示音。具体类图如图 5-13 所示：

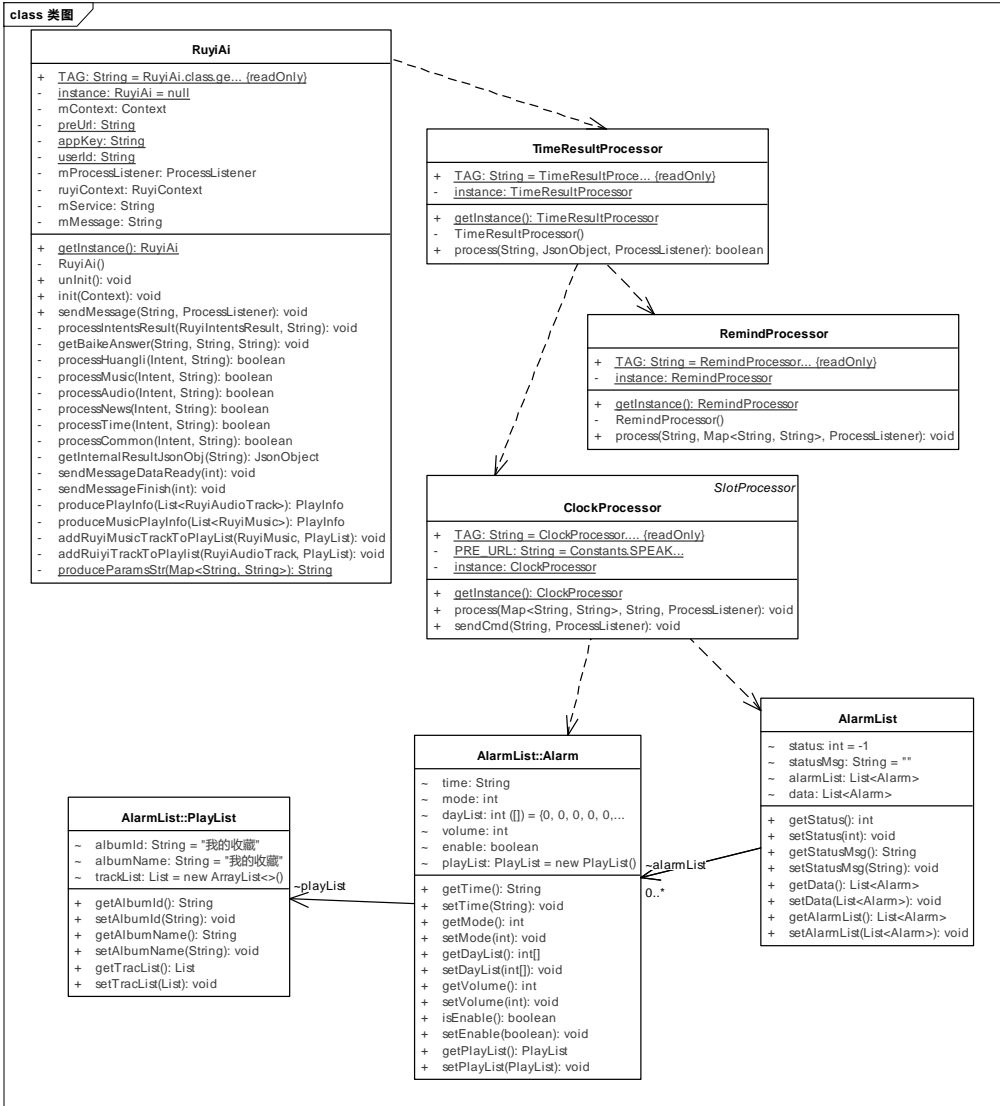


图 5-14 闹钟与提醒类类图

5.3.6 百科类

5.3.6.1 设计描述

百科类作为用户资料类查询，根据返回结果内容最多不超过 200 个字，如果超过则对结果进行截取。同时百科类也作为其他功能无法处理时的一种兜底策略，对用户的对话内容进行百度百科查询。

5.3.6.2 流程图

用户查询百科，如：故宫，王菲。语音识别正确后，第三方 NLP 进行处理，判断是否收到百科结果，收到则进行动态语音合成播报并显示 UI 交互，否则进行百度百科查询，如果查询到内容，动态进行语音合并播报显示 UI 交互。如果查询不到内容，给出提示音，关闭语音唤醒，语音识别继续。具体流程图如图 5-15 所示：

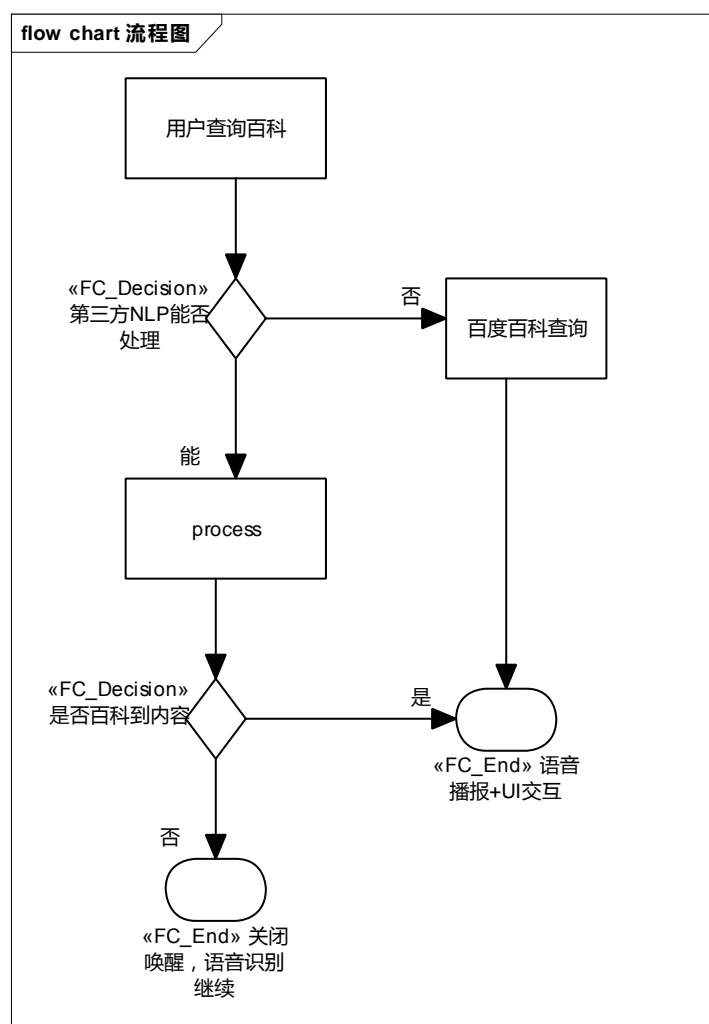


图 5-15 百科类流程图

5.3.6.3 类图

下面是百科类的核心类图。

RuyiAi 类：其中 processCommon() 方法为根据第三方 NLP 结果进行处理，如果处理不了，则通过 getBaikeAnswer() 方法进行百科查询。

BaikeDataCrawler 类：用于通过 HTTP 查询百科内容，并在查询完后告诉监听器处理结果。

BaikeDataCrawlerListener 接口：用于在 BaikeDataCrawler 类处理完后回调结果处理逻辑。如动态语音合成播报结果、显示交互 UI。具体类图如图 5-16 所示：

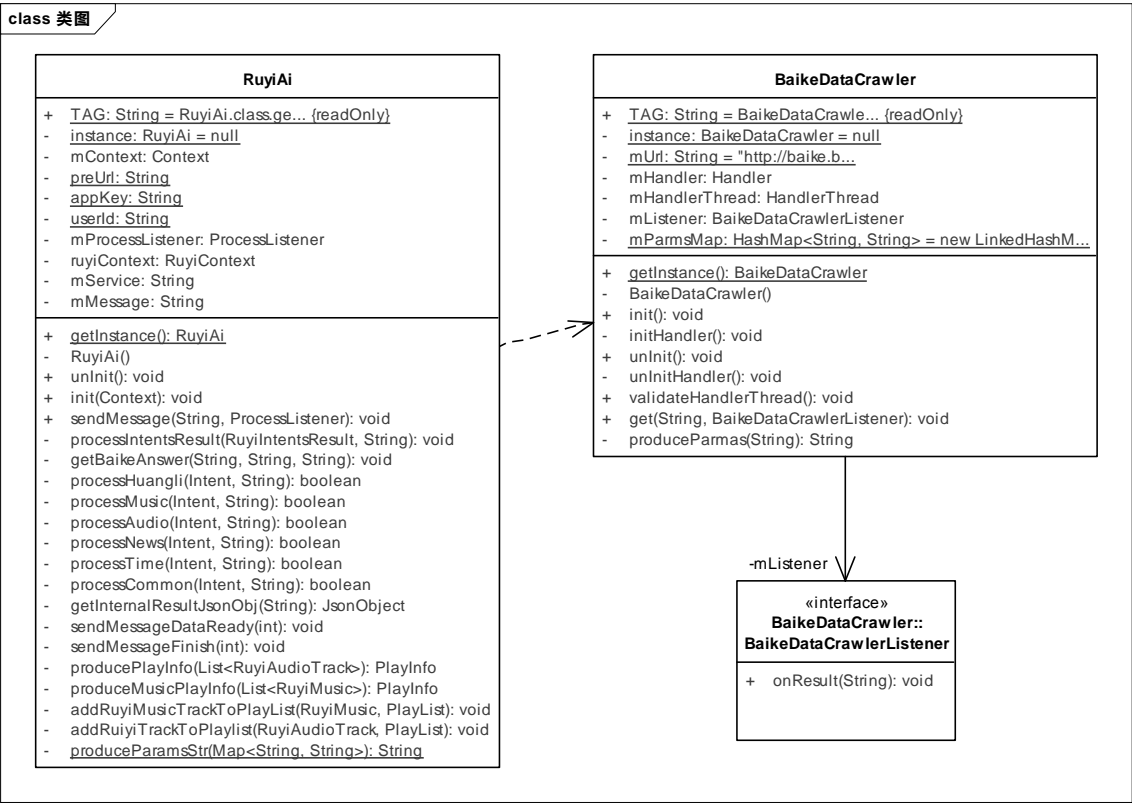


图 5-16 百科类类图

5.3.7 聊天类

5.3.7.1 设计描述

聊天类功能作为可触发的功能类型，以及其他功能无法满足情况下的兜底策略。若一个需求掉落到聊天后依然无法处理，则视为处理失败。对话需要能记忆上文的内容、domain 等，以保持连续性。

5.3.7.2 流程图

用户进行语音交互，判断是否为聊天类型。如果是，则根据第三方返回内容，提取机器聊天结果进行语音动态合成并显示 UI 交互，交互完成后回调控制语音识别的开启

与关闭。否则交由其他类型进行处理。具体流程图如图 5-17 所示。

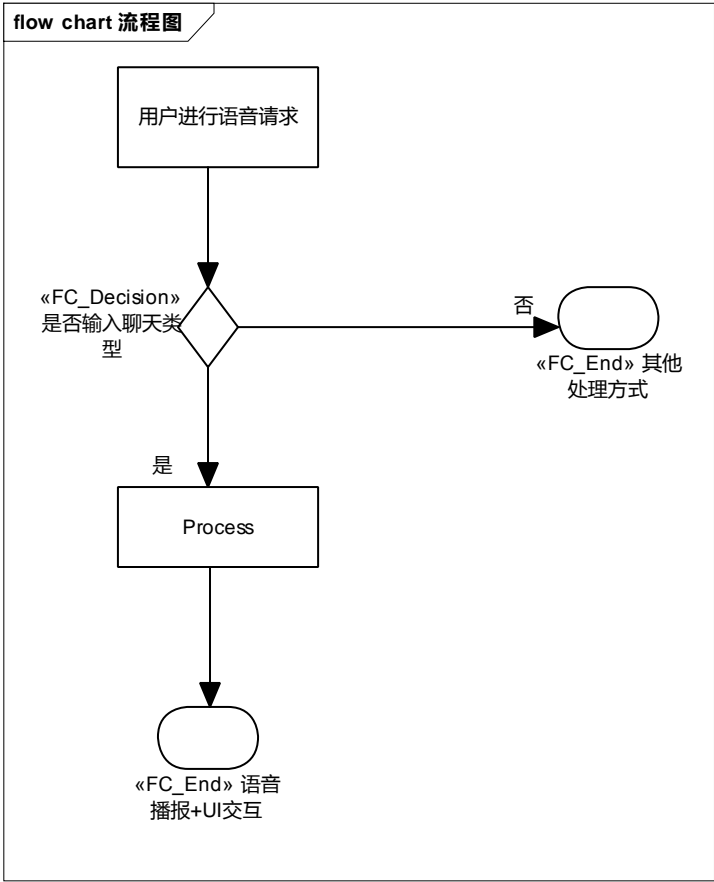


图 5-17 聊天类流程图

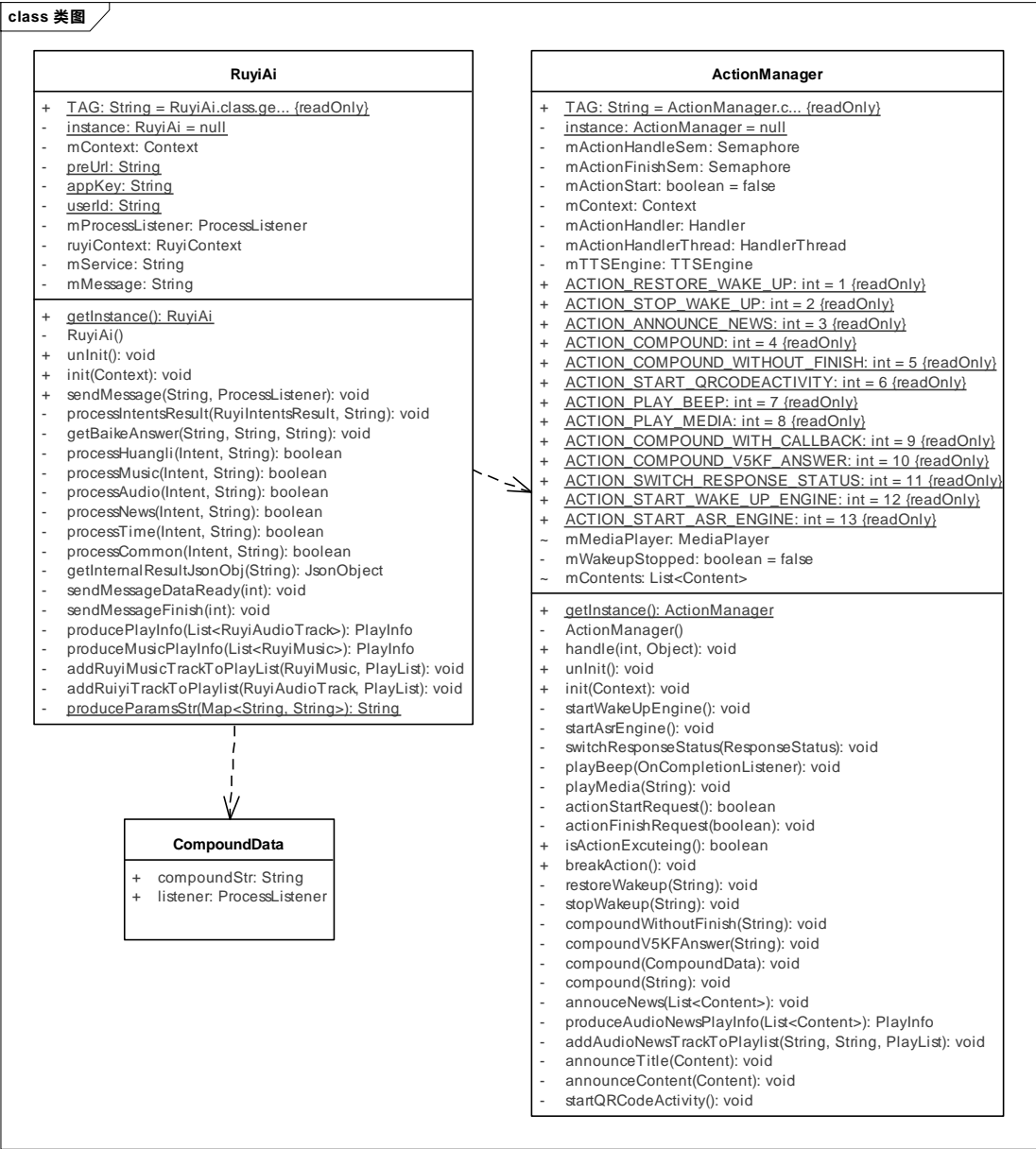
5.3.7.3 类图

下面是聊天类的核心类图。

RuyiAi 类：根据第三方 NLP 处理结果，如果用户语音识别后的文本类型属于聊天（chat_kids）类型。则从第三方 NLP 返回的 JSON 数据中提取出语音合成文本。

CompoundData 类：用于封装根据用户语音机器返回的聊天结果。

ActionManager 类：对聊天结果调用语音合成引擎动态合成，并在合成结束后调用结果处理逻辑。具体类图如图 5-18 所示：



5-18 聊天类类图

5.4 本章小结

本章通过 UML 建模分析对系统的模块设计与实现进行了详细描述。

6 结论

本项目来源于本人在魅族科技有限公司北京分公司的实际项目——“gravity 智能音箱”。我在实习过程中主要负责语音控制系统的设计与实现。通过背景分析发现智能音箱很有发展前景，也是未来智能家居的入口，在音箱上的进行语音交互是智能音箱的标志性特色，也是核心竞争力。语音控制系统的设计与实现就是这样的需求下诞生的，旨在提供良好的语音交互效果，完成通过语音交互与音箱功能的对接及日常对话。

本文从背景分析、相关技术、需求分析、系统架构、模块设计与实现等五个方面对该系统进行阐释。

关于背景分析，主要论述了项目的背景及意义、智能音箱应用现状、论文主要工作。通过背景分析读者能了解到智能音箱的发展前景以及对论文有一个整体性的了解，便于理解论文的内容。

关于相关技术，简单介绍了项目涉及到的关键技术。关键技术包括：语音识别，语音合成，自然语言处理，Android activity，JSON 数据，多线程与回调。通过对关键技术的阐述让读者了解项目的难度以及方便阅读文论。

关于需求分析，对项目做了简介，模块划分，以及主要功能的用例图分析。通过需求分析，读者能了解到语音控制系统的主要功能以及系统模块间的组织关系，从而对项目有个全面的了解。

关于系统架构，说明了系统的整体架构及主要的系统接口。通过架构分析，整体性介绍了系统工作流程，模块的作用以及模块与模块之间的连接关系。通过系统接口论述，让读者理解模块与模块之间具体是如何对接以及相互协作的，语音控制系统是如何设计的。

关于模块设计与实现，通过设计描述，流程图，类图对模块设计与实现进行了详细论述，让读者对系统的实现细节有了更好的理解，了解到语音控制系统是如何实现的。

通过项目开发与论文撰写，我学到了很多東西同时也发现自己还有很多不足需要改进提升。项目开发过程中涉及到了一些前沿技术，如：语音识别，自然语言处理等，极大的开阔了我的眼界，也让我对技术更加热爱与崇拜。开发过程中少不了与其他同事沟通，协作开发，让我明白了及时沟通的重要性，还有好的沟通技巧能很好的提高沟通效率。我也需要不停的学习进步，一点点慢慢的改正自己不足之处。

参考文献

- [1] 李刚.《疯狂 Android 讲义》电子工业出版社,2011-6
- [2] 高洪岩.《Java 多线程编程核心技术》.机械工业出版社,2015-6
- [3] 弗里曼.《Head First 设计模式（中文版）》.中国电力出版,2007-9
- [4] 张伟,面向智能家居的嵌入式语音控制系统的设计[D],郑州大,2015(05)
- [5] 付蔚,唐鹏光,李倩智能家居语音控制系统的设计[J].自动化仪表.,2014(01)
- [6] 高弋坤. 科大讯飞:移动互联网进入语音时代[J]. 通信世界,2010(42)
- [7] 夏霆.智能家居系统——语音识别[J]. 广东建材. 2010(08)
- [8] 陈哲.智能家居语音控制系统的设计与实现[D].电子科技大学,2013(09)
- [9] 宗成庆.《统计自然语言处理（第 2 版）》.清华大学出版社
- [10] 董晓刚. android 中基于回调机制的事件处理[J]. 中国电子商务, 2013 (13): 88-88
- [11] 陈伟. Android Handler 消息传递机制在人机界面软件设计中的应用[J]. 科技信息, 2014 (13):91-91
- [12] 邵维忠.统一建模语言 UML 述评.计算机研究与发展,1999,36(4);385-394
- [13] 魏博.软件非功能性需求建模与设计决策评估方法研究[D].中国科学院大学,2013
- [14] Liu Zhi-Gang,Huang Wei.The design of smart home system based on Wi-Fi. ComputationalProblem-Solving (ICCP) . 2012

致 谢

大学四年结束了，一路走来，磕磕绊绊，有伤心难过，也有兴奋快乐，在这个过程中很多人给了我帮助与指导，真心感谢每一个出现在我生活中的人，每一个帮助我的人。

在这里首先感谢我的导师，不论是论文撰写，还是工作就业都给了我极大的帮助。没有导师的指导，论文写作过程中遇到的很多问题，我都将很难解决。找过导师好多次，每一次导师都是极其耐心的帮我分析每一个问题，梳理我想不清的问题，正是有导师热心的帮助我才能顺利完成论文。在就业选择的时候也咨询了老师意，得到了很宝贵的建议，让我对就业是事也更加清晰明确了。

然后要感谢我实习单位的同事和导师。实习的时候跟着导师学到了很多，在导师严格要求和鼓励下，在技术方面我得到了很大的提升。在项目开发过程中也得到了很多同事的帮助，每次有项目难点要攻克，我解决不了的时候，他们都及时的给了我帮助。在他们身上学习到了认真负责的工作态度，和对专业素质不断提升的进取心。

在学校呆了四年，怎么能不感谢培养我的学校、学院以及同学呢。正是有他们的帮助，我的大学四年才过能非常的丰富多彩。不管是学习，还是生活中他们在我这里都担当了很重要的角色，我会记得他们的好。

最后再次感谢所有人，祝大家工作顺心，生活愉快，身体健康，感谢大家出现在我的生命里。