



(12) 发明专利申请

(10) 申请公布号 CN 114639384 A

(43) 申请公布日 2022.06.17

(21) 申请号 202210526270.0

(22) 申请日 2022.05.16

(71) 申请人 腾讯科技(深圳)有限公司  
地址 518044 广东省深圳市南山区高新区  
科技中一路腾讯大厦35层

(72) 发明人 李曼曼

(74) 专利代理机构 北京同达信恒知识产权代理  
有限公司 11291  
专利代理师 朱佳

(51) Int. Cl.

G10L 15/22 (2006.01)

G10L 15/26 (2006.01)

G06F 40/30 (2020.01)

B60R 16/037 (2006.01)

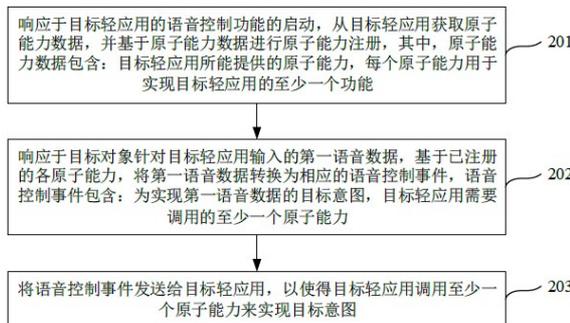
权利要求书3页 说明书26页 附图14页

(54) 发明名称

语音控制方法、装置、设备及计算机存储介质

(57) 摘要

本申请公开了一种语音控制方法、装置、设备及计算机存储介质,涉及语音控制技术领域,该方法通过启动语音控制功能时,则对目标轻应用的原子能力进行注册,即能够获知目标轻应用能够实现哪些功能,进而当接收到语音控制数据时,能够基于已注册的原子能力,来对语音控制数据的意图进行转换,获得目标轻应用可执行的语音控制事件,从而实现通过语音控制的方式实现对轻应用的控制效果,那么在进行驾驶时,则无需进行手动操作,进而提升了车辆驾驶的安全性。



1. 一种语音控制方法,其特征在于,应用于语音控制系统中,所述方法包括:

响应于目标轻应用的语音控制功能的启动,从所述目标轻应用获取原子能力数据,并基于所述原子能力数据进行原子能力注册,其中,所述原子能力数据包含:所述目标轻应用所能提供的原子能力,每个原子能力用于实现所述目标轻应用的至少一个功能;

响应于目标对象针对所述目标轻应用输入的第一语音数据,基于已注册的各原子能力,将所述第一语音数据转换为相应的语音控制事件,所述语音控制事件包含:为实现所述第一语音数据的目标意图,所述目标轻应用需要调用的至少一个原子能力;

将所述语音控制事件发送给所述目标轻应用,以使得所述目标轻应用调用所述至少一个原子能力来实现所述目标意图。

2. 如权利要求1所述的方法,其特征在于,响应于目标对象针对所述目标轻应用输入的第一语音数据,基于已注册的各原子能力,将所述第一语音数据转换为相应的语音控制事件,包括:

对所述第一语音数据进行语音识别,获取所述第一语音数据中包含的文本信息;

对所述文本信息进行语义识别,确定所述文本信息对应的目标意图;

基于预设的事件配置信息以及所述已注册的各原子能力,将所述目标意图转换为所述语音控制事件,其中,所述事件配置信息包含:基于所述目标意图对所述至少一个原子能力进行参数配置的指导信息。

3. 如权利要求2所述的方法,其特征在于,所述对所述文本信息进行语义识别,确定所述文本信息对应的目标意图,包括:

从所述目标轻应用获取应用描述信息,所述应用描述信息包含:所述目标轻应用的基本描述信息和语音控制上下文信息中的至少一种;

基于所述应用描述信息,对所述文本信息进行语义识别,确定所述目标意图。

4. 如权利要求1所述的方法,其特征在于,所述方法还包括:

将所述原子能力数据发送给云端服务器,以在云端服务器进行各原子能力的注册;

从所述目标轻应用获取应用描述信息,所述应用描述信息包含:所述目标轻应用的基本描述信息和语音控制上下文信息中的至少一种;

则在响应于目标对象针对所述目标轻应用输入的第一语音数据,基于已注册的各原子能力,将所述第一语音数据转换为相应的语音控制事件,包括:

将所述应用描述信息以及所述第一语音数据发送给所述云端服务器,并接收所述云端服务器返回的所述语音控制事件,其中,所述语音控制事件为所述云端服务器基于所述应用描述信息以及所述第一语音数据确定所述目标意图后,基于已注册的原子能力对所述目标意图进行转换得到的。

5. 如权利要求1所述的方法,其特征在于,所述语音控制系统包括接口组件和语音组件,所述接口组件封装了所述目标轻应用提供的第三方服务接口;

则所述响应于目标轻应用的语音控制功能的启动,从所述目标轻应用获取原子能力数据,并基于所述原子能力数据进行原子能力注册,包括:

响应于目标轻应用的语音控制功能的启动,触发所述接口组件调用所述第三方服务接口获取所述原子能力数据;

通过所述接口组件向所述语音组件发送所述原子能力数据,以在所述语音组件进行原

子能力注册。

6. 如权利要求5所述的方法,其特征在于,所述目标轻应用所在的终端设备包括音频采集装置,所述接口组件封装有所述终端设备的操作系统提供的语音数据获取接口;

则在响应于目标对象针对所述目标轻应用输入的第一语音数据之前,所述方法还包括:

调用所述语音组件通过所述语音数据获取接口,向所述操作系统申请所述音频采集装置的使用权限;

调用所述语音组件通过所述语音数据获取接口,接收所述音频采集装置采集的所述第一语音数据。

7. 如权利要求1~5任一所述的方法,其特征在于,在将所述语音控制事件发送给所述目标轻应用之后,所述方法还包括:

接收所述目标轻应用执行所述语音控制事件后返回的执行结果;

若所述执行结果指示需要进行反馈控制,则启动录音功能,获取所述目标对象输入的第二语音数据;

基于所述已注册的各原子能力,将所述第二语音数据转换为相应的语音控制事件,并将获得的语音控制事件发送给所述目标轻应用执行。

8. 如权利要求6所述的方法,其特征在于,在接收所述目标轻应用执行所述语音控制事件后返回的执行结果之后,所述方法还包括:

若所述执行结果指示不需要进行反馈控制,则清空所述已注册的原子能力;

通知所述操作系统释放自身占用的音频采集装置的使用权限。

9. 如权利要求1-5任一所述的方法,其特征在于,所述方法还包括:

响应于目标对象针对目标应用输入的第三语音数据,对所述第三语音数据进行语音端点检测;

若检测到所述第三语音数据中的语音开始位置,则从所述语音开始位置开始,对所述第三语音数据进行语音识别,直至检测到所述第三语音数据中的语音结束位置;

基于获得的语音识别结果确定对所述目标应用进行唤醒时,对所述目标应用进行唤醒操作。

10. 如权利要求8所述的方法,其特征在于,在响应于目标对象针对目标应用输入的第三语音数据,对所述第三语音数据进行语音端点检测之前,所述方法还包括:

响应于所述目标应用的语音唤醒功能的启动,获取所述目标应用中激活页面对应的唤醒词集合,并基于所述唤醒词集合进行唤醒词注册;

则基于获得的语音识别结果确定对所述目标应用进行唤醒时,对所述目标应用进行唤醒操作,包括:

若所述语音识别结果包含已注册的唤醒词,则确定对所述目标应用进行唤醒;

对所述目标应用进行所述唤醒操作,并将所述语音识别结果包含的文本信息发送给所述目标应用。

11. 一种语音控制装置,其特征在于,应用于语音控制系统中,所述装置包括:

能力注册单元,用于响应于目标轻应用的语音控制功能的启动,从所述目标轻应用获取原子能力数据,并基于所述原子能力数据进行原子能力注册,其中,所述原子能力数据包

含:所述目标轻应用所能提供的原子能力,每个原子能力用于实现所述目标轻应用的至少一个功能;

事件转换单元,用于响应于目标对象针对所述目标轻应用输入的第一语音数据,基于已注册的各原子能力,将所述第一语音数据转换为相应的语音控制事件,所述语音控制事件包含:为实现所述第一语音数据的目标意图,所述目标轻应用需要调用的至少一个原子能力;

传输单元,用于将所述语音控制事件发送给所述目标轻应用,以使得所述目标轻应用调用所述至少一个原子能力来实现所述目标意图。

12. 一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,

所述处理器执行所述计算机程序时实现权利要求1至10任一项所述方法的步骤。

13. 一种计算机存储介质,其上存储有计算机程序指令,其特征在于,

该计算机程序指令被处理器执行时实现权利要求1至10任一项所述方法的步骤。

14. 一种计算机程序产品,包括计算机程序指令,其特征在于,

该计算机程序指令被处理器执行时实现权利要求1至10任一项所述方法的步骤。

## 语音控制方法、装置、设备及计算机存储介质

### 技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及语音控制技术领域,提供一种语音控制方法、装置、设备及计算机存储介质。

### 背景技术

[0002] 目前,汽车市场的车机功能越来越丰富,驾驶车辆出行时,常常存在需要对车辆部件或者终端设备进行操作的情况,但这些操作时容易使得驾驶员驾驶车辆出现失误,从而影响车辆驾驶的安全性。因此,为了提升驾驶安全性,同时提升操作的便捷性,通过语音控制方式实现上述操作是一种很好的替代方式,故而目前较多的终端设备都搭载了语音控制系统,例如可以通过语音控制方式关窗或者打开某个应用程序(application,APP)等。

[0003] 目前,越来越多的应用程序中引入了小程序等轻应用形式,以小程序为例,其作为一种无需安装即点即用的应用形式,且车厂无需对小程序进行开发集成,只需要应用厂商自行进行维护,与车载场景轻度使用、设备容量小、项目周期长的特点十分契合。

[0004] 但是,目前语音控制的功能通常只停留在应用程序层面,只能实现应用程序的开启、关闭或者切换等较为简单的功能,从而无法实现诸如小程序内的应用功能的语音控制,使得目前轻应用的控制仍然依赖于手动操作,操作方式比较单一,并且还可能增加车辆驾驶安全风险。

### 发明内容

[0005] 本申请实施例提供一种语音控制方法、装置、设备及计算机存储介质,用于通过语音控制的方式实现轻应用的控制,提升车辆驾驶的安全性。

[0006] 一方面,提供一种语音控制方法,应用于语音控制系统中,所述方法包括:

响应于目标轻应用的语音控制功能的启动,从所述目标轻应用获取原子能力数据,并基于所述原子能力数据进行原子能力注册,其中,所述原子能力数据包含:所述目标轻应用所能提供的原子能力,每个原子能力用于实现所述目标轻应用的至少一个功能;

响应于目标对象针对所述目标轻应用输入的第一语音数据,基于已注册的各原子能力,将所述第一语音数据转换为相应的语音控制事件,所述语音控制事件包含:为实现所述第一语音数据的目标意图,所述目标轻应用需要调用的至少一个原子能力;

将所述语音控制事件发送给所述目标轻应用,以使得所述目标轻应用调用所述至少一个原子能力来实现所述目标意图。

[0007] 一方面,提供一种语音控制装置,应用于语音控制系统中,所述装置包括:

能力注册单元,用于响应于目标轻应用的语音控制功能的启动,从所述目标轻应用获取原子能力数据,并基于所述原子能力数据进行原子能力注册,其中,所述原子能力数据包含:所述目标轻应用所能提供的原子能力,每个原子能力用于实现所述目标轻应用的至少一个功能;

事件转换单元,用于响应于目标对象针对所述目标轻应用输入的第一语音数据,

基于已注册的各原子能力,将所述第一语音数据转换为相应的语音控制事件,所述语音控制事件包含:为实现所述第一语音数据的目标意图,所述目标轻应用需要调用的至少一个原子能力;

传输单元,用于将所述语音控制事件发送给所述目标轻应用,以使得所述目标轻应用调用所述至少一个原子能力来实现所述目标意图。

[0008] 可选的,所述事件转换单元,具体用于:

对所述第一语音数据进行语音识别,获取所述第一语音数据中包含的文本信息;  
对所述文本信息进行语义识别,确定所述文本信息对应的目标意图;

基于预设的事件配置信息以及所述已注册的各原子能力,将所述目标意图转换为所述语音控制事件,其中,所述事件配置信息包含:基于所述目标意图对所述至少一个原子能力进行参数配置的指导信息。

[0009] 可选的,所述事件转换单元,具体用于:

从所述目标轻应用获取应用描述信息,所述应用描述信息包含:所述目标轻应用的基本描述信息和语音控制上下文信息中的至少一种;

基于所述应用描述信息,对所述文本信息进行语义识别,确定所述目标意图。

[0010] 可选的,所述事件转换单元,具体用于:

将所述原子能力数据发送给云端服务器,以在云端服务器进行各原子能力的注册;

从所述目标轻应用获取应用描述信息,所述应用描述信息包含:所述目标轻应用的基本描述信息和语音控制上下文信息中的至少一种;

将所述应用描述信息以及所述第一语音数据发送给所述云端服务器,并接收所述云端服务器返回的所述语音控制事件,其中,所述语音控制事件为所述云端服务器基于所述应用描述信息以及所述第一语音数据确定所述目标意图后,基于已注册的原子能力对所述目标意图进行转换得到的。

[0011] 可选的,所述语音控制系统包括接口组件和语音组件,所述接口组件封装了所述目标轻应用提供的第三方服务接口;

则所述能力注册单元,具体用于:

响应于目标轻应用的语音控制功能的启动,触发所述接口组件调用所述第三方服务接口获取所述原子能力数据;

通过所述接口组件向所述语音组件发送所述原子能力数据,以在所述语音组件进行原子能力注册。

[0012] 可选的,所述目标轻应用所在的终端设备包括音频采集装置,所述接口组件封装有所述终端设备的操作系统提供的语音数据获取接口;则所述装置还包括申请单元,用于:

调用所述语音组件通过所述语音数据获取接口,向所述操作系统申请所述音频采集装置的使用权限;

调用所述语音组件通过所述语音数据获取接口,接收所述音频采集装置采集的所述第一语音数据。

[0013] 可选的,所述事件转换单元,还用于:

接收所述目标轻应用执行所述语音控制事件后返回的执行结果;

若所述执行结果指示需要进行反馈控制,则启动录音功能,获取所述目标对象输入的第二语音数据;

基于所述已注册的各原子能力,将所述第二语音数据转换为相应的语音控制事件,并将获得的语音控制事件发送给所述目标轻应用执行。

[0014] 可选的,所述事件转换单元,还用于:

若所述执行结果指示不需要进行反馈控制,则清空所述已注册的原子能力;  
通知所述操作系统释放自身占用的音频采集装置的使用权限。

[0015] 可选的,所述装置还包括唤醒单元,用于:

响应于目标对象针对目标应用输入的第三语音数据,对所述第三语音数据进行语音端点检测;

若检测到所述第三语音数据中的语音开始位置,则从所述语音开始位置开始,对所述第三语音数据进行语音识别,直至检测到所述第三语音数据中的语音结束位置;

基于获得的语音识别结果确定对所述目标应用进行唤醒时,对所述目标应用进行唤醒操作。

[0016] 可选的,所述唤醒单元,具体用于:

响应于所述目标应用的语音唤醒功能的启动,获取所述目标应用中激活页面对应的唤醒词集合,并基于所述唤醒词集合进行唤醒词注册;

若所述语音识别结果包含已注册的唤醒词,则确定对所述目标应用进行唤醒;

对所述目标应用进行所述唤醒操作,并将所述语音识别结果包含的文本信息发送给所述目标应用。

[0017] 一方面,提供一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述任一种方法的步骤。

[0018] 一方面,提供一种计算机存储介质,其上存储有计算机程序指令,该计算机程序指令被处理器执行时实现上述任一种方法的步骤。

[0019] 一方面,提供一种计算机程序产品或计算机程序,该计算机程序产品或计算机程序包括计算机指令,该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令,处理器执行该计算机指令,使得该计算机设备执行上述任一种方法的步骤。

[0020] 本申请实施例中,当目标轻应用启动语音控制功能时,则从目标轻应用获取原子能力数据,进行原子能力注册,当接收到针对目标轻应用输入的第一语音数据时,则基于已注册各原子能力,将第一语音数据转换为相应的语音控制事件,即为实现第一语音数据的目标意图,目标轻应用需要调用哪些原子能力,从而将语音控制事件发送给目标轻应用,以使得目标轻应用调用相应的原子能力来实现语音控制意图。也就是说,本申请实施例中通过启动语音控制功能时,则对目标轻应用的原子能力进行注册,即能够获知目标轻应用能够实现哪些功能,进而当接收到语音控制数据时,能够基于已注册的原子能力,来对语音控制数据的意图进行转换,获得目标轻应用可执行的语音控制事件,从而实现通过语音控制的方式实现对轻应用的控制效果,那么在进行驾驶时,则无需进行手动操作,进而提升了车辆驾驶的安全性。

## 附图说明

[0021] 为了更清楚地说明本申请实施例或相关技术中的技术方案,下面将对实施例或相关技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据提供的附图获得其他的附图。

- [0022] 图1为本申请实施例提供的应用场景示意图;  
图2为本申请实施例提供的语音控制方法的流程示意图;  
图3A~图3C为本申请实施例提供的语音控制的界面示意图;  
图4为本申请实施例提供的事件转换的一种流程示意图;  
图5为本申请实施例提供的多轮语音会话控制过程的流程示意图;  
图6A~图6C为本申请实施例提供的目标轻应用的执行结果的示意图;  
图7为本申请实施例提供的针对应用进行语音控制的流程示意图;  
图8A和图8B为本申请实施例提供的应用唤醒的页面示意图;  
图9为本申请实施例提供的语音控制系统的架构示意图;  
图10为本申请实施例提供的功能类之间的关系示意图;  
图11为本申请实施例提供的系统语音整体框架示意图;  
图12为本申请实施例提供的小程序使用语音控制的流程示意图;  
图13为本申请实施例提供的语义识别和执行的一种示例图;  
图14为本申请实施例提供的意图识别的流程示意图;  
图15为本申请实施例提供的应用进行语音唤醒的流程示意图;  
图16为本申请实施例提供的语音控制装置的一种结构示意图;  
图17为本申请实施例提供的一种计算机设备的结构示意图;  
图18为本申请实施例提供的另一种计算机设备的结构示意图。

## 具体实施方式

[0023] 为使本申请的目的、技术方案和优点更加清楚明白,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。在不冲突的情况下,本申请中的实施例及实施例中的特征可以相互任意组合。并且,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。

[0024] 为便于理解本申请实施例提供的技术方案,这里先对本申请实施例使用的一些关键名词进行解释:

**轻应用:**轻应用是一种以客户端为宿主,在宿主运行环境下运行的程序,用户可以通过在宿主运行环境下直接打开轻应用,获得轻应用提供方提供的服务,轻应用例如可以为小程序、公众号等形式。其无需下载、即搜即用的特性使得无需占用终端设备的存储空间安装客户端,更加节省存储空间,且操作快捷。本申请实施例中,轻应用的宿主运行环境可以为应用程序客户端,也可以为车载终端设备的操作系统。

[0025] 小程序(mp-parser,MPP):是一种轻应用形式,其同样具备无需下载、即搜即用的特性。

[0026] 原子能力(atomicability,aa):对于一段程序而言,程序的原子性是指整个程序中的所有操作,要么全部完成,要么全部不完成,不可能停滞在中间的环节,相应的,原子性操作是不可中断的,要么全部执行成功,要么全部执行失败,从而,原子能力是指可独立运行的最小能力单元。对于轻应用而言,一个原子能力能够实现该轻应用中的至少一个功能,例如对于XX讲故事,其为一个阅读型的轻应用,那么其原子能力能够用于实现播放故事、搜索故事等相关的功能,或者,对于XX视频,其为一个视频播放应用,那么其原子能力能够用于实现视频搜索、视频播放、视频暂停等相关的功能。

[0027] 在一个应用程序中,各个功能通常都是封装成各个功能函数,来被调用实现相应的功能,因而从另一个角度去理解原子能力,一个原子能力也可以认为是一个功能函数,其可以包含函数名、参数等相关信息,一个原子能力被调用时则是执行一个功能函数,例如视频播放功能的实现是基于调用视频播放接口函数来实现的。

[0028] 意图:是指输入的语音数据期望达到的目的。例如输入了一段“用XX讲故事播放XX故事”,其意图在于通过“XX讲故事”这个应用程序(可以为客户端,也可以为轻应用)播放“XX故事”这个内容,一般理解,为了实现这个目的,那么需要在“XX讲故事”这个应用程序搜索到“XX故事”这个内容的资源,在调用“XX讲故事”的播放器启动“XX故事”的播放,当然,在实际情况下还可能涉及到其他环节。

[0029] 语音端点检测:又称语音活动检测(Voice Activity Detection,VAD),语音边界检测等,用于针对一段音频数据检测其语音状态,语音状态包含开始状态、暂停状态、暂停恢复状态以及结束状态等。

[0030] 自动语音识别(Automatic Speech Recognition,ASR):其目标是将语音数据中的词汇内容转换为计算机可读的输入,例如按键、二进制编码或者字符序列等。

[0031] 激活页面:指应用中打开的页面,通常是指处于该应用的显示界面的当前页面,例如,当应用处于前台运行状态时,激活页面则是指该应用当前打开的页面,当应用处于后台运行状态时,则激活页面可以是指应用切换前打开的页面。

[0032] 本申请实施例涉及人工智能技术,人工智能(Artificial Intelligence,AI)是利用数字计算机或者数字计算机控制的机器模拟、延伸和扩展人的智能,感知环境、获取知识并使用知识获得最佳结果的理论、方法、技术及应用系统。换句话说,人工智能是计算机科学的一个综合技术,它企图了解智能的实质,并生产出一种新的能以人类智能相似的方式做出反应的智能机器。人工智能也就是研究各种智能机器的设计原理与实现方法,使机器具有感知、推理与决策的功能。

[0033] 人工智能技术是一门综合学科,涉及领域广泛,既有硬件层面的技术也有软件层面的技术。人工智能基础技术一般包括如传感器、专用人工智能芯片、云计算、分布式存储、大数据处理技术、操作/交互系统、机电一体化等技术。人工智能软件技术主要包括计算机视觉技术、语音处理技术、自然语言处理技术以及机器学习/深度学习、自动驾驶、智慧交通等几大方向。

[0034] 语音技术(Speech Technology)的关键技术有自动语音识别技术和语音合成技术以及声纹识别技术。让计算机能听、能看、能说、能感觉,是未来人机交互的发展方向,其中

语音成为未来最被看好的人机交互方式之一。

[0035] 自然语言处理(Nature Language processing,NLP)是计算机科学领域与人工智能领域中的一个重要方向。它研究能实现人与计算机之间用自然语言进行有效通信的各种理论和方法。自然语言处理是一门融语言学、计算机科学、数学于一体的科学。因此,这一领域的研究将涉及自然语言,即人们日常使用的语言,所以它与语言学的研究有着密切的联系。自然语言处理技术通常包括文本处理、语义理解、机器翻译、机器人问答、知识图谱等技术。

[0036] 本申请实施例提供的方案涉及人工智能的语音技术以及自然语言处理技术等技术。具体的,本申请实施例当目标轻应用启动语音控制功能时,则从目标轻应用获取原子能力数据,进行原子能力注册,当接收到语音数据时,采用语音技术对采集的语音数据进行语音识别,并基于自然语言处理技术对语音识别结果进行语义理解,以确定语音数据对应的目标意图,从而基于已注册的原子能力,对目标意图进行转换,生成目标轻应用可执行的语音控制事件。此外,还可以基于语音技术进行语音合成,来生成与驾驶人员进行人机交互的语音,并进行播放。

[0037] 下面对本申请实施例的设计思想进行简要介绍。

[0038] 目前,越来越多的应用程序中引入了小程序等轻应用形式,但是,目前语音控制的功能通常只停留在应用程序层面,只能实现应用程序的开启、关闭或者切换等较为简单的功能,从而无法实现诸如小程序内的应用功能的语音控制,使得目前轻应用的控制仍然依赖于手动操作,操作方式比较单一,并且还可能增加车辆驾驶安全风险。

[0039] 鉴于此,本申请实施例提供一种语音控制方法,在该方法中,当目标轻应用启动语音控制功能时,则从目标轻应用获取原子能力数据,进行原子能力注册,当接收到针对目标轻应用输入的第一语音数据时,则基于已注册的各原子能力,将第一语音数据转换为相应的语音控制事件,即为实现第一语音数据的目标意图,目标轻应用需要调用哪些原子能力,从而将语音控制事件发送给目标轻应用,以使得目标轻应用调用相应的原子能力来实现语音控制意图。也就是说,本申请实施例中通过启动语音控制功能时,则对目标轻应用的原子能力进行注册,即能够获知目标轻应用能够实现哪些功能,进而当接收到语音控制数据时,能够基于已注册的原子能力,来对语音控制数据的意图进行转换,获得目标轻应用可执行的语音控制事件,从而实现通过语音控制的方式实现对轻应用的控制效果,那么在进行驾驶时,则无需进行手动操作,进而提升了车辆驾驶的安全性。

[0040] 此外,本申请实施例中,还可以通过语音唤醒的方式唤醒终端设备中的目标应用,具体而言,通过对目标应用的激活页面的唤醒词集合进行注册,当接收到语音数据时,则进行语音识别,从而基于识别结果与唤醒词集合进行匹配,当能够匹配得上时,则对目标应用进行唤醒操作,执行相应的功能,实现语音唤醒的功能。

[0041] 下面对本申请实施例的技术方案能够适用的应用场景做一些简单介绍,需要说明的是,以下介绍的应用场景仅用于说明本申请实施例而非限定。在具体实施过程中,可以根据实际需要灵活地应用本申请实施例提供的技术方案。

[0042] 本申请实施例提供的方案可以适用于大多数语音控制场景中,尤其适用于车载终端设备的语音控制场景。如图1所示,为本申请实施例提供的一种应用场景示意图,在该场景中,可以包括车载终端设备101和云端服务器102。

[0043] 车载终端设备101例如可以为车辆中控设备或者与车辆进行通信连接的终端设备等,包括但不限于手机、电脑、智能语音交互设备、智能家电、车载终端、飞行器等。车载终端设备101可以运行有语音控制进程以及被控应用和轻应用,云端服务器102可以为语音控制进程提供后台服务的后台服务器,例如可以是独立的物理服务器,也可以是多个物理服务器构成的服务器集群或者分布式系统,还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、即内容分发网络(Content Delivery Network,CDN)、以及大数据和人工智能平台等基础云计算服务的云服务器,但并不局限于此。

[0044] 需要说明的是,本申请实施例中的语音控制方法可以由车载终端设备101或云端服务器102单独执行,也可以由云端服务器102和车载终端设备101共同执行。例如,由车载终端设备101响应于目标轻应用的语音控制功能的启动,从目标轻应用获取原子能力数据,并基于原子能力数据进行原子能力注册,并基于已注册的各原子能力,将输入的第一语音数据转换为相应的语音控制事件,以及将语音控制事件发送给目标轻应用,以使得目标轻应用调用至少一个原子能力来实现目标意图。或者,由云端服务器102执行上述步骤。又或者,由车载终端设备101响应于目标轻应用的语音控制功能的启动,从目标轻应用获取原子能力数据,并基于原子能力数据进行原子能力注册,同时接收输入的第一语音数据,并由云端服务器102利用其丰富的计算资源对第一语音数据进行语义识别和转换,生成相应的语音控制事件,再由车载终端设备101将语音控制事件发送给目标轻应用,以使得目标轻应用调用至少一个原子能力来实现目标意图,本申请在此不做具体限定,下文主要是以车载终端设备101和云端服务器102集合为例进行举例说明的。

[0045] 以车载终端设备101执行上述步骤为例,车载终端设备101可以包括一个或多个处理器、存储器以及与终端交互的I/O接口等。其中,车载终端设备101的存储器中还可以存储本申请实施例提供的语音控制方法的程序指令,这些程序指令被处理器执行时能够用以实现本申请实施例提供的语音控制方法的步骤,以实现相应的语音控制过程。

[0046] 本申请实施例中,车载终端设备101和云端服务器102之间可以通过一个或者多个网络103进行直接或间接的通信连接。该网络103可以是有线网络,也可以是无网络,例如无线网络可以是移动蜂窝网络,或者可以是无线保真(Wireless-Fidelity,WIFI)网络,当然还可以是其他可能的网络,本发明实施例对此不做限制。

[0047] 需要说明的是,图1所示只是举例说明,实际上车载终端设备和云端服务器的数量不受限制,在本申请实施例中不做具体限定。当然,本申请实施例提供的方法并不限于图1所示的应用场景中,还可以用于其他可能的应用场景,本申请实施例并不进行限制。对于图1所示的应用场景的各个设备所能实现的功能将在后续的方法实施例中一并进行描述,在此先不过多赘述。

[0048] 在一种可能的应用场景中,为了便于降低检索的通信时延,可以在各个地区部署云端服务器102,或为了负载均衡,可以由不同的云端服务器102分别去服务不同地区的车载终端设备101,例如,车载终端设备101位于地点a,与服务地点a的云端服务器102建立通信连接,车载终端设备101位于地点b,与服务地点b的云端服务器102建立通信连接,多个云端服务器102组成一数据共享系统,通过区块链实现数据的共享。

[0049] 对于数据共享系统中的每个云端服务器102,均具有与该云端服务器102对应的节

点标识,数据共享系统中的每个云端服务器102均可以存储有数据共享系统中其他云端服务器102的节点标识,以便后续根据其他云端服务器102的节点标识,将生成的区块广播至数据共享系统中的其他云端服务器102。每个云端服务器102中可维护一个节点标识列表,将云端服务器102名称和节点标识对应存储至该节点标识列表中。其中,节点标识可为网络之间互联的协议(Internet Protocol,IP)地址以及其他任一种能够用于标识该节点的信息。

[0050] 下面结合上述描述的应用场景,参考附图来描述本申请示例性实施方式提供的方法,需要注意的是,上述应用场景仅是为了便于理解本申请的精神和原理而示出,本申请的实施方式在此方面不受任何限制。

[0051] 参见图2所示,为本申请实施例提供的语音控制方法的流程示意图,该方法的具体实施流程如下:

步骤201:响应于目标轻应用的语音控制功能的启动,从目标轻应用获取原子能力数据,并基于原子能力数据进行原子能力注册,其中,原子能力数据包含:目标轻应用所能提供的原子能力,每个原子能力用于实现目标轻应用的至少一个功能。

[0052] 本申请实施例中,目标轻应用的语音控制功能的启动,可以基于多种情况的触发,包括但不限于如下几种情况:

(1)目标轻应用的启动,即当目标轻应用启动时,则目标轻应用的语音控制功能相应启动。

[0053] 具体的,目标轻应用可以采用按钮方式进行启动,如点击目标轻应用的图标或者硬件按钮快速启动的方式启动目标轻应用,或者,目标轻应用也可以采用语音控制的方式启动,参见图3A所示,为语音控制方式启动目标轻应用的示意图,其中,车载终端设备当前的页面为“爱电影”页面,若想要进入“XX读书”这个轻应用页面,则可以按照图3A的方式,通过语音输入“打开XX读书”,则车载终端设备则会接收到这条语音数据,并通过语音识别可以确定用户输入的文本内容为“打开XX读书”,从而在当前页面的语音控制显示区域内显示该文本内容,同时,还会基于该文本执行相应的动作,即启动“XX读书”轻应用,则相应的,针对“XX读书”的语音控制功能相应随之启动。

[0054] (2)在目标轻应用的页面中针对语音控制功能按钮进行操作。参见图3B和图3C所示,图3B为轻应用内页面的示意图,该页面为轻应用内的搜索页面,包含了搜索文本输入框,同时,在该搜索页面内还包括了语音控制功能按钮,通过该语音控制功能按钮可以通过语音输入的方式进行搜索内容的输入,从而对该语音控制功能按钮进行操作,则可以触发该轻应用的语音控制功能的启动。

[0055] 需要说明的是,在图3B中可见,语音控制功能按钮可以设置与多个位置,除了在实现特定功能,如图3B中文本输入框位置设置之外,还可以在如图3B的菜单栏中进行设置,从而快速启动语音控制功能,并且即使当前页面中未集成语音控制功能按钮,也能够触发对该轻应用的语音控制功能的启动。

[0056] 本申请实施例中,当目标轻应用的语音控制功能启动时,则可以从目标轻应用获取原子能力数据,并基于原子能力数据进行原子能力注册,也就是说获知目标轻应用所能实现的功能以及如何实现的方式,以便后续对语音数据进行转换成目标轻应用可执行的语音控制事件。

[0057] 在一种可能的实施方式中,可以通过在车载终端设备中集成语音控制进程,而通常支持语音控制的车载终端设备的操作系统本身也会集成语音控制进程,为进行区别,这里将车载终端设备的操作系统提供的语音控制进程称之为系统语音控制进程,简称为系统语音,将本申请实施例提供的语音控制进程称之为扩展语音控制进程,简称为扩展语音,系统语音和扩展语音作为独立的进程在车载终端设备中运行,即在车载终端设备自身语音控制的基础上,对语音控制功能进行了进一步的扩展。

[0058] 那么,当目标轻应用的语音控制功能启动时,则触发扩展语音从目标轻应用获取原子能力数据,并基于原子能力数据进行原子能力注册,注册的过程的目的在于使得扩展语音分析和获知目标轻应用所能实现的功能以及如何实现的方式。

[0059] 具体的,扩展语音可以采用独立进程的方式运行,即扩展语音与其他应用程序互为独立的程序;或者,也可以将扩展语音作为软件开发工具包(Software Development Kit, SDK)的形式集成到能够作为轻应用宿主的应用程序中。

[0060] 步骤202:响应于目标对象针对目标轻应用输入的第一语音数据,基于已注册的各原子能力,将第一语音数据转换为相应的语音控制事件,语音控制事件包含:为实现第一语音数据的目标意图,目标轻应用需要调用的至少一个原子能力。

[0061] 本申请实施例中,在目标轻应用的语音控制功能启动后,则目标对象可以针对目标轻应用进行语音输入,以进行相应的功能控制。参见图3C所示,触发语音控制功能后,则呈现相应的语音控制输入区域,便于显示目标对象输入的语音数据对应的文本信息。

[0062] 在一种可能的实施方式中,车载终端设备中的扩展语音可以集成语音识别和语义识别的功能,那么针对目标对象输入的第一语音数据,可以由车载终端设备的扩展语音进行事件的转化。

[0063] 具体的,参见图4所示,为将第一语音数据转换为相应的语音控制事件的一种流程示意图。

[0064] 当扩展语音获取到第一语音数据时,则扩展语音可以首先对第一语音数据进行语音识别,获取第一语音数据中包含的文本信息。参见图4所示,当前正在进行音乐播放,目标对象输入了一段语音数据,通过对其进行语音识别,得到相应的文本信息为“下一首”,同时,使得到的文本信息还可以向目标对象进行显示,如图4中在显示界面上显示当前用户输入的文本为“下一首”。进而,基于识别到的文本信息进行语义识别,以确定文本信息对应的目标意图,从而知晓目标对象所想要目标轻应用实现的功能,如图4所示,基于语义识别可以知道意图为切换下一首,即图4所示的“next”操作。此外,还可以预先存储时间配置信息,事件配置信息可以包含基于目标意图对至少一个原子能力进行参数配置的指导信息,即指导获知意图后如何选择原子能力和配置原子能力参数,其可以作为一种剧本配置,即在目标意图作为一种目标,事件配置信息给定了实现这种目标需要配置的剧本过程,这样,才能够基于预设的事件配置信息以及已注册的各原子能力,将目标意图转换为语音控制事件(action),参见图4所示,通过事件转换可以得到相应的action为“WeCarFlow: canPlayNext”,其中,“canPlayNext”为调用的切换至下一首的函数名称,“WeCarFlow”为当前播放音乐的APP名称,即指示“WeCarFlow”执行“canPlayNext”函数,以切换至下一首歌曲。

[0065] 本申请实施例中,通过基于人工智能的语音识别技术和自然语言理解技术对目标

对象输入的语音数据进行意图识别,从而准确得到目标对象的意图,从而提升了转换得到的action的准确性,并且通过预先配置事件配置信息和进行原子能力注册,使得扩展语音能够拥有action转换的能力,以将目标对象的语音数据转换为目标轻应用可执行的action,从而实现对轻应用的语音控制过程,提升车辆驾驶的安全性。

[0066] 考虑到在实际应用过程中,意图的识别与轻应用本身信息也是存在一定的关联关系的,同一文本在不同的轻应用中可能被识别为不同的意图,因而在进行意图识别时,还可以结合目标轻应用的应用描述信息来进行,以提升意图识别的准确性。

[0067] 具体的,还可以从目标轻应用获取应用描述(context)信息,context信息可以包含如下信息中的至少一种:

(1) 目标轻应用的基本描述信息,例如轻应用标识(APP ID)、版本号等轻应用信息(APP info)、当前登录的账户(account)信息、当前定位的位置(location)信息以及当前系统信息(system info),system info指示当前设备的操作系统,如可以为Linux系统、窗口(windows)系统等。

[0068] (2) 语音控制上下文信息中的至少一种,在进行语音控制时可能存在多轮会话,而这多轮会话之间可能存在关联,例如目标对象在上一轮对话时,目标对象输入“播放XX故事”,经过搜索发现搜索结果存在多个,那么可能会让目标对象进行选择,例如提示目标对象“播放第几个呢”,进而目标对象输入“播放第一个”,从而结合上下文进行理解,才能够准确确定目标对象需要播放的是当前搜索结果的第一个内容。

[0069] 进而,在进行意图识别时,则可以结合应用描述信息,对文本信息进行语义识别,从而得到准确的目标意图。

[0070] 在一种可能的实施方式中,考虑到车载终端设备通常计算能力有限,而语义识别通常需要利用预训练的语义识别模型,因而通常需要消耗较多的计算资源,可能使得意图识别过程较长,出现向目标对象反馈延迟较长的情况,因此,为了提升意图识别的速度,那么可以利用云端丰富的计算资源来为语义识别提供强大的计算能力。

[0071] 因此,当扩展语音接收到目标对象输入的第一语音数据时,则将第一语音数据传输给云端服务器,进而云端服务器进行上述语音识别和语义识别的过程,并将识别到的文本信息和action返回给扩展语音。需要说明的是,在实际过程中,目标对象输入语音数据是存在一定的过程的,而扩展语音或者云端服务器可以在目标对象输入的过程中同时进行语音识别过程,即目标对象一边输入,后台一边进行识别,无需等待用户输入完毕再继续后续的识别过程,提升语音控制的实时性。

[0072] 具体的,若采用云端服务器进行上述语音识别和语义识别的过程,则为了该过程的正常执行,扩展语音还可以将原子能力数据发送给云端服务器,从而在云端服务器进行各原子能力的注册。

[0073] 考虑到云端服务器具备一定的存储能力,因而可以在云端服务器进行各轻应用的原子能力的存储,也就是说,当云端服务器中未注册目标轻应用的原子能力时,扩展语音可以将获取的目标轻应用的原子能力数据发送给云端服务器,以进行目标轻应用的原子能力的注册,而一旦注册,则云端服务器可以在整个流程结束时,进行反注册清空相关数据,或者,还可以持续保留已注册的原子能力,那么当下一次针对该车载终端设备对目标轻应用进行语音控制或者其他车载终端设备对其进行语音控制时,只需要发送当前需要控制的目

标轻应用的标识信息,例如APPID和版本号,而无需再次发送原子能力数据进行注册,从而节省传输资源,并且降低了传输的数据量,以加快整体流程的响应速度。

[0074] 云端服务器已进行原子能力注册后,则扩展语音从目标轻应用获取应用描述信息后,还可以将应用描述信息以及第一语音数据发送给云端服务器,这样,云端服务器可以根据应用描述信息以及第一语音数据确定目标意图后,基于已注册的原子能力对目标意图进行转换获得action,并将action返回给扩展语音。

[0075] 步骤203:将语音控制事件发送给目标轻应用,以使得目标轻应用调用至少一个原子能力来实现目标意图。

[0076] 本申请实施例中,通过启动语音控制功能时,则对目标轻应用的原子能力进行注册,进而当接收到语音控制数据时,能够基于已注册的原子能力,来对语音控制数据的意图进行转换,获得目标轻应用可执行的action,从而实现通过语音控制的方式实现对轻应用的控制效果,那么在进行驾驶时,则无需进行手动操作,进而提升了车辆驾驶的安全性。

[0077] 考虑到在进行语音控制时可能存在多轮会话,因而在步骤203之后,还可以实施如图5所示的过程,其为本申请实施例提供的多轮语音会话控制过程的流程示意图。

[0078] 步骤204:接收目标轻应用执行语音控制事件后返回的执行结果。

[0079] 步骤205:基于执行结果判断是否需要进行反馈控制。

[0080] 目标轻应用执行action后,则扩展语音可以从目标轻应用获取执行结果。

[0081] 例如,参见图6A~图6C所示,为目标轻应用的执行结果的一种示意图,如图6A所示,当输入的第一语音数据的文本信息为“用XX音乐播放儿歌”时,则XX音乐执行的结果是搜索得到XX音乐内的儿歌,但可能无法确定目标对象需要播放哪首儿歌,参见图6B所示,那么XX音乐可以输出反馈语音“为您找到儿歌歌曲XX首,需要播放第几个呢”,并在显示界面上显示搜索结果,根据此执行结果可以知道,播放歌曲的目的并未达到,则可以确定目标对象需要进行反馈(feedback)。

[0082] 或者,当输入的第一语音数据的文本信息为“切换下一首”,则XX音乐执行切换操作,切换到下一首歌曲即达到了目的,从而根据执行结果可以确定无需进行feedback。

[0083] 步骤206:步骤205的判断结果为是,即若执行结果指示需要进行反馈控制,则启动录音功能,获取目标对象输入的第二语音数据。

[0084] 步骤207:基于已注册的各原子能力,将第二语音数据转换为相应的语音控制事件,并将获得的语音控制事件发送给目标轻应用执行。

[0085] 参见图6B所示,当判断需要feedback时,则在启动录音,采集目标对象输入的第二语音数据输入“播放第2个”,则通过语音识别和语义识别过程后,可以确定目标对象的意图在于播放当前页面上显示的歌曲2,则生成播放该歌曲的action,并发送给目标轻应用执行,从而使得目标轻应用执行播放歌曲2的操作,呈现如图6C所示的音乐播放界面。

[0086] 步骤206和207的过程则与第一语音数据转换过程类似,因此在此不再进行赘述。

[0087] 步骤208:步骤205的判断结果为否,若执行结果指示不需要进行反馈控制,则对目标轻应用进行反注册,例如清空目标轻应用已注册的原子能力,并通知操作系统释放自身占用的音频采集装置的使用权限。

[0088] 本申请实施例中,可以通过多轮语音会话的方式,逐步控制实现目标对象的意图,且在每一轮的操作中,目标对象均无需进行手动操作,从而提升了车辆驾驶的安全性。

[0089] 本申请实施例中,除了轻应用的语音控制之外,仍然可以对车载终端设备中的应用进行语音控制,参见图7所示,为针对应用进行语音控制的流程示意图。

[0090] 步骤701:响应于目标应用的语音唤醒功能的启动,获取目标应用的唤醒词集合,并基于唤醒词集合进行唤醒词注册。

[0091] 本申请实施例中,应用的唤醒是指启动应用执行某个功能,可以包含如下几种情况下的唤醒:

(1)在车载终端设备处于息屏状态下,唤醒应用来执行功能。

[0092] (2)当应用在后台运行时,唤醒应用来执行功能。

[0093] (3)当应用未启动时,唤醒应用来执行功能。

[0094] 当然,也可以包括其他可能的唤醒情况,本申请实施例对此不做限制。

[0095] 具体的,为了实现应用的唤醒,首先需要设置该应用的唤醒词,进而当输入的语音数据能够与唤醒词匹配上时,才认定目标对象想要唤醒应用,因此,在后续流程实施之前,需要被唤醒的应用需要向扩展语音进行唤醒词的注册。例如针对即时通讯应用,可以注册“打开XX应用”“登录”“发送消息”“取消发送”等唤醒词。

[0096] 参见图8A所示,为本申请实施例提供的页面示意图。目标对象可以通过启动系统语音,拉起扩展语音,例如目标对象语音输入“你好,X车”,启动当前车载终端设备的系统语音并拉起扩展语音,呈现如图8A的画面,其中显示了可选择的快捷选项,当然,目标对象也可以手动操作选择,或者也可以语音录入选择。

[0097] 步骤702:响应于目标对象针对目标应用输入的第三语音数据,对第三语音数据进行语音端点检测。

[0098] 本申请实施例中,当使用语音控制方式进行应用的控制时,则扩展语音启动录音,获取目标对象输入的第三语音数据,同时,由于目标对象输入的语音数据中并非全部的时长均包含文本信息,因而可以通过对其语音端点检测,从第三语音数据中检测出有效语音数据,也就是目标对象真正在说话的部分语音片段。

[0099] 以上述图8A为例,用户可以采用语音输入的方式录入第三语音数据“发聊天消息”。

[0100] 具体的,基于语音片段的位置,语音端点检测出的状态可以包括:

(1)开始状态,即一个有效语音片段的起始位置。

[0101] (2)暂停状态,即距离上一次检测到有效语音的时长不大于设定时长的位置,或者目标对象请求暂停录音录入。

[0102] (3)结束状态,即一个有效语音片段的终止位置。

[0103] (4)超时状态,即超过一定的时长未检测任何有效语音。

[0104] (5)网络连接超时状态,即与云端服务器的连接超过一定的时长。

[0105] (6)暂停恢复状态,即从暂停状态到检测到有效语音时的状态,或者目标对象请求恢复录音录入。

[0106] 步骤703:若检测到第三语音数据中的语音开始位置,则从语音开始位置开始,对第三语音数据进行语音识别,直至检测到第三语音数据中的语音结束位置。

[0107] 在具体实施过程中,目标对象在进行语音输入的过程中,语音端点检测的过程也可以随之进行,当检测到第三语音数据中的语音开始位置,即出现有效语音的位置时,则从

该位置开始,对第三语音数据进行语音识别,直至检测到出现语音结束位置为止,也就是检测到出现结束状态为止,从而能够减少语音识别的数据量。进一步提升语音识别的速度。

[0108] 步骤704:基于获得的语音识别结果确定是否对目标应用进行唤醒。

[0109] 步骤705:若步骤704的确定结果为是,即确定对目标应用进行唤醒时,对目标应用进行唤醒操作。

[0110] 具体的,针对获得的语音识别结果,可以对上述注册的唤醒词进行遍历,从而确定语音识别结果中是否包含已注册的唤醒词,若包含,则确定需要对目标应用进行唤醒,从而执行唤醒操作,例如打开目标应用,并向目标应用发送语音识别结果,以便目标应用进行相应的处理,或者,将语音识别结果转换成相应的控制指令发送给目标应用,实现对目标应用的唤醒控制,这样,目标对象无需进行手动唤醒控制,以提升驾驶的安全性。

[0111] 参见图8B所示,当“发聊天消息”匹配上聊天应用时,则对聊天应用进行唤醒操作,从而呈现如图8B所示的聊天消息发送界面,目标对象可以针对聊天应用录入需要发送的聊天语音进行发送,聊天对象可以基于目标对象的操作或者语音选择而确定,也可以是聊天应用的激活页面所对应的聊天对象。

[0112] 本申请实施例中,考虑到误唤醒的问题,采取了分页面场景注册唤醒词的方式,具体而言,每个页面场景都对应着不同的唤醒词集合,当目标应用切换激活页面时,则从目标应用获取激活页面对应的唤醒词集合,并基于唤醒词集合进行唤醒词注册,每个页面场景都需要走完一次图7所示的完整的流程。

[0113] 在一种可能的实施方式中,可以按照页面场景类型进行唤醒词集合的划分,如划分为登录场景、发送消息场景、语音聊天场景或者视频聊天场景,同一种页面场景可以对应相同的唤醒词集合,则切换至新的页面时,则获取该页面场景对应的唤醒词集合进行注册。例如,目标应用中从页面1切换至页面2时,则获取页面2其页面场景类型对应的唤醒词集合进行注册。

[0114] 在另一种可能的实施方式中,为了更为精准的对每个页面进行控制,也可以针对每个页面进行唤醒词集合的设定,则切换至新的页面时,则获取该页面对应的唤醒词集合进行注册。例如,目标应用中从页面1切换至页面2时,则获取页面2的唤醒词集合进行注册。

[0115] 请参见图9所示,为本申请实施例提供的一种语音控制系统的架构示意图。本申请实施例中,上述的语音控制方法过程,可以通过图9所示的架构图实现。在该架构中,主要包括轻应用MPP、接口组件Moss、语音组件Speech和云端服务器,其中接口组件Moss、语音组件Speech组成了上述提及的扩展语音,接口组件Moss、语音组件Speech在实际应用时,可以为同一进程的两部分,也可以分别为两个独立的进程,例如语音组件Speech作为单独的进程模块,接口组件Moss作为系统语音与speech的适配层。且需要说明的是,图9所示的云端服务器的功能同样可以在车载终端设备中实现,这里具体是以云端服务器实现为例进行介绍的。上述各个模块的介绍如下:

#### 一、接口组件Moss

Moss层主要作为操作系统服务接口和第三方服务接口的封装层,封装了车载终端设备中的轻应用提供的第三方服务接口和操作系统提供的系统服务接口,例如用于实现语音数据采集的语音数据获取接口,如图9所示,Moss层包括如下两个模块:

(1)应用接口管理模块,即图9所示的client模块,其封装了Moss platform应用程

序接口(Application Programming Interface,API),可以包含应用、轻应用以及扩展语音自身提供的API。

[0116] (2) 系统接口管理模块,即图9所示的server模块,其封装了操作系统提供的系统服务接口,可以实现向操作系统申请音频焦点以实现录音(recorder)功能,即通过调用操作系统提供的语音数据获取接口,发起语音采集装置的使用权限,申请成功后,则可以获取到语音采集装置采集的语音数据。此外,server模块还封装了与操作系统通信的传输协议,例如可以采用图9所示的protobuf(protocol buffer,为一种混合语言数据标准)消息协议,当然,也可以采用其他可能的协议,本申请实施例对此不做限制。

[0117] 参见图9所示,Moss与MPP之间,以及Moss与Speech之间可以通过进程间通信(Inter-Process Communication,IPC)进行通信,图9所示的箭头为语音识别过程中的数据流示意。

[0118] 二、语音组件Speech

Speech是一种文本融合语音模块,是本申请实施例实施语音控制过程的核心模块,能够实现语音控制中的语音识别和语义识别等过程,其可以包含如下几个模块:

(1) 语音识别和执行(VrLogic)模块,VrLogic模块主要是与云端服务器的语义相关服务模块进行交互的,用于使用云端服务器提供的在线语义识别功能。VrLogic模块包含IntraDmResponse子模块和DmActomic子模块,IntraDmResponse子模块为Speech内部唤醒和调用的接口模块,实现内部之间的接口调用和响应,DmActomic子模块用于实现云端服务器的交互,来实现在线语义识别功能。

[0119] (2) AISDK模块,AISDK模块内置了多种基于人工智能的语音处理能力,例如图9所示,AISDK模块包含VAD子模块、ASR子模块、NLU模块和唤醒引擎子模块,VAD子模块用于实现VAD功能,ASR子模块用于实现ASR功能,NLU模块用于实现语义识别相关功能,唤醒引擎子模块用于实现唤醒控制相关功能。

[0120] (3) 应用管理(Clientmanger)模块,用于维护VAD子模块的6种状态,即上述实施例中提及的开始状态、暂停状态等,Clientmanger模块还用于注册应用的技能(如原子能力和唤醒词)、唤醒、识别和回调功能。

[0121] (4) 对话管理(Sessionmanager)模块,当存在多个会话时,则需要Sessionmanager模块进行会话管理,能够实现会话打断以及会话的多轮逻辑管理功能。

[0122] (5) 分发(Dispatcher)模块,Dispatcher模块主要用于实现落域和仲裁结果的分发,把云端服务器返回的语音数据返回给不同的应用或者轻应用。

[0123] (6) SpeechMsglistener接口,为Speech对外提供的调用接口,用于接收和管理外部的其他进程发送的信息。

[0124] (7) SpeechClientService接口,为Speech对内提供的调用接口,用于协调内部之间的调用,当SpeechMsglistener接口接收到外部发送的信息时,需要调用SpeechClientService接口发送给内部的相应模块,如图9中将信息发送给Client管理模块。

[0125] 参见图9所示,在进行语音识别时,Speech的SpeechMsglistener接口接收外部发送的信息,并通过SpeechClientService接口通知给内部的Client管理模块,Client管理模块进行client原子能力的注册,通过Sessionmanager模块拉起engine模块,将语音数据传

入AISDK进行识别,若需进行在线的语音识别,则可以调用VrLogic模块与云端服务器进行交互,获得的语音结果通过Dispatcher模块进行分发,即调用SpeechClientService接口发送给不同的客户端。

### [0126] 三、云端服务器

云端服务器主要实现语义识别相关功能,参见图9所示,云端服务器可以包括如下几个模块:

(1)对话管理代理服务(Dmproxy Server)模块,Dmproxy Server模块为云端服务器的接入层,用于实现与扩展语音之间的交互,例如接收扩展语音发送来的Context信息和ASR结果,并将最终得到的对话管理(Dialog Management,DM)结果返回给扩展语音。

[0127] (2)对话管理服务(Dm Server)模块,用于进行语义仲裁,选取优选的语义识别结果,参见图9所示,Dm Server模块将Context信息和ASR结果发送给NLU模块进行语义识别,基于其返回的语义识别结果进行语义仲裁,选取最终优选的NLU结果。

[0128] (3)NLU服务(Server)模块,用于接收Dm Server模块传入的应用描述信息和ASR结果,调用云端服务器的语义识别能力进行语义识别,并将语义识别结果返回给Dm Server模块。如图9所示,云端服务器可以预先集成DingDang(一种语义识别引擎)、AISpeech以及自建语义识别模型等语音识别能力。

[0129] (4)技能服务(SkillServer)模块和剧本配置模块,剧本配置模块存储了事件配置信息,SkillServer模块能够基于剧本配置模块的事件配置信息进行action的转化,进而通过DM Server模块和Dmproxy Server模块返回给车载终端设备。参见图9所示,Dm Server模块将NLU结果发送给SkillServer模块,从而SkillServer模块能够基于剧本配置模块的事件配置信息进行action的转化。

### [0130] 四、轻应用MPP

轻应用主要需要提供语音控制相关的信息。

[0131] (1)Context信息,包含账户信息、location、APP info以及system info等信息,在Speech需要时,可以调用Speech提供的speechclient接口发送给Speech。

[0132] (2)此外,还可以从轻应用获取车联统一网关接入模块(TAACC)信息。

[0133] (3)当语音识别完成时,轻应用的显示界面需要进行相应的显示或者播控操作等,此时可以调用SpeechmessageHandler接口将相关信息发送至MPP的相关模块,如需要用户界面(UI)显示时,则调用SpeechmessageHandler接口发送给MPP,MPP进行相应的显示,或者需要控制MPP进行音乐播放时,则调用SpeechmessageHandler接口将播放请求发送给MPP,MPP进行相应音乐的播放。

[0134] 对应于图9的功能模块,本申请实施例针对每个功能模块设计了相应的功能类,请参见图10所示,为本申请实施例提供的Speech中涉及到的功能类之间关系示意图,Speech可以包含如下几种功能类:

(1)WeAppSpeechMsgListener类对应于图9中SpeechMsglistener接口,该类用于APP向Speech发送消息时的消息收听,其可以包括如下几种功能:

- Registered\_clients//用于client注册
- App\_session\_listener//用于APP的会话收听
- HandleStartWechatRecord//用于处理APP的开始录音请求

- HandleStopWechatRecord//用于处理Wechat停止录音请求

参见图10所示,WeAppSpeechMsgListener类可以对SpeechClientService类进行调用(use)。

[0135] (2)SpeechClientService类对应于图9中的SpeechClientService接口,其可以包括如下功能(以-标注的为私有(private)类,以+标注的为共有(public)类):

- Client\_manager\_//用于实现会话管理功能
- StartWechatRecord//用于实现启动录音
- HandleVadEvent//用于处理Vad事件
- HandleRegisterClient//用于处理client注册
- HandleRegisterWakeup//用于处理唤醒注册
- HandleUnRegisterWakeup//用于处理唤醒反注册
- + ProcessRecordData//用于处理录音数据

参见图10所示,SpeechClientService类可以调用APP在录音时注册的唤醒词被唤醒后的返回(WechatRecordWakeupcallback)类,其功能包含处理语音请求的服务(p\_client\_service)和唤醒后的回调(OnWakeupWithDoa),DOA表示音区;此外,SpeechClientService类还可以调用APP在录音时的会话收听(WechatRecordSessionListener)类,其功能包含处理语音请求的服务(p\_client\_service)、引擎状态回调(OnEngineState)和语音识别音量时的返回(OnSRVolume);此外,还可以调用APP在录音时的回调类WechatRecordcallback类,其功能。

[0136] (3)ClientManager类对应图9所示的Clientmanger模块,其可以包括如下功能:

- scene\_id\_map\_//用于场景ID管理
- client\_set\_//用于客户端管理
- session\_manager\_//用于会话管理
- + RegisterClient//用于client注册
- + RegisterWakeupcallback//用于唤醒回调注册
- + RegisterListener//用于接收对象注册

参见图10所示,ClientManager类可以调用Client类实现APP相关内容的管理,Client类的子类功能包括唤醒场景(scene\_wakeup)管理、客户端类型(IClientType)管理、APP唤醒事件注册(RegisterAppWakeupEvent)、APP注册的唤醒词唤醒回调(OnAppWakeup)以及APP语音识别事件回调(OnAppSREvent),Client类进一步可调用WakeupScene类进行唤醒场景设置,其功能包含时间设置(event\_set)和场景标识(scene\_id)设置以及添加唤醒事件(AddWakeupEvent),WakeupScene类进一步可调用WakeupEvent类进行唤醒事件设置,其功能包含事件标识(event\_id)管理、唤醒词设置(wakeup\_word\_set)和提取场景词(Getscenewords)等。

[0137] 此外,ClientManager类可以调用客户端管理(IClientManager)接口,Client类还可以调用IClientRegisterAppDomain接口,其为进行客户端注册APP时的落域接口。

[0138] (4)SessionManager类对应图9所示的Sessionmanager模块,实现会话管理相关功能,其可以包括如下功能:

- engine\_//用于引擎管理

- + struct EngineConfig//用于引擎结构配置管理
- engine\_listener\_//用于接收引擎的消息
- session\_listeners\_//用于会话收听管理
- session\_container\_//用于会话容器管理
- + RegisterWakeupWords//用于注册唤醒词
- + StartWakeup//用于触发唤醒

参见图10所示,SessionManager类能够在ClientManager类的调用下,实现会话管理相关功能,其可以调用会话容器(SessionContainer)类实现会话容器相关功能,SessionContainer类其功能包括会话清单(session\_list)管理、分发(dispatcher)、触发唤醒会话(StartWakeupSession)和处理语义(HandleSemantic)等功能;SessionContainer类调用Session类实现会话功能,Session类包括分发(dispatcher)、启动会话(StartSession)和处理SR结果(HandleSRResult)等功能;Session类的分发功能可以调用FrameworkDispatcher类来实现,FrameworkDispatcher类包含客户端管理(client\_mgr)、对话管理(Session\_mgr)、分发唤醒事件(DispathWakeupEvent)和分发SR事件(DispathSREvent)等功能。

[0139] 此外,Speech还可以包含ISessionListener类,ISessionListener类继承(extends)SessionManager类,能够实现录音数据返回(OnrecordData)、录音状态切换返回(OnrecordStatusChange)等功能。

[0140] (5)Engine类对应于图9所示的Engine模块,实现相应的引擎相关的功能,其可以包括如下功能类:

- mlEngineListener//用于收听引擎消息
- + aisdklnit//用于AISDK的引擎初始化
- + dingdang\_upload\_wakeup\_keywords\_json//用于为dingdang引擎加载唤醒词文件
- + dingdang\_register\_event\_callback//用于为dingdang引擎注册事件回调
- onEventCallback//用于实现事件回调功能
- onWakeupCallback//用于实现唤醒回调功能

参见图10所示,Engine类可以调用Recorder类获取相关的语音数据,Recorder类能够实现IPC通信的客户端(mlPCCClient)的管理、录音数据返回(onDataProcessed)和启动录音(startRecorder)等功能,Recorder类可以调用platformmlPC类实现IPC通信相关功能,platformmlPC类具体可以包括打开进程队列(open\_queue)、关闭(close\_queue)以及申请音频焦点(requestAudioFocus)等功能。

[0141] 为实现Engine类的功能,Engine类还可以调用AISDK实现相关功能,以及调用IEngineListener类获取引擎相关信息,IEngineListener类包含了列举引擎模型(enum EngineMode)、语音识别结果返回(onSpeechResult)、引擎状态返回(onEngineState)以及正常模式的唤醒(onNormalWordWakeup)等功能。

[0142] 此外,Speech还可以包含IEngine类,IEngine类继承Engine类,能够实现模型切换(switchMode)、启动录音(startRecord)、注册唤醒词(registerWakeupWords)和反注册唤醒词(unregisterWakeupWords)等功能。

[0143] 可以看到,上述各类的主要业务逻辑为SeesionManager类实现,Clientmanager用于为不同客户端使用Speech后的管理维护,Engine使用AISDK的具体引擎模块,SpeechClientServer负责IPC通信的接口。

[0144] 请参见图11所示,为本申请实施例提供的系统语音整体框架示意图。该整体框架主要划分为客户端和云端两部分,其中,在客户端部分,与上述语音控制系统类似,Speech作为单独的进程模块,Moss层作为系统语音控制进程与Speech的适配层,因而针对Speech和Moss层的各功能则不再进行赘述。

[0145] 参见图11所示,车载终端设备的操作系统包括系统语音和Tai controller,Tai controller是指车载操作系统的系统层,系统语音同样可以采用云端服务器的方式提供协助,如图11所示,云端可以包括两部分,一部分为系统语音对应的云端服务器,另一部分为扩展语音对应的云端服务器,系统语音可以与其云端服务器进行交互获取服务,例如在系统语音在获取语音数据后,可以将语音数据发送给其对应的云端服务器,云端服务器的接入层接收语音数据后,会基于自身的ASR能力提取该语音数据中的文本信息,并基于自身的NLU能力对文本信息进行语义识别,获得相应的语义识别结果,此外,还需要基于自身的DM能力结合语义识别结果,来确定向目标对象答复的内容。

[0146] Moss层封装了第三方服务接口和操作系统提供的系统服务接口,作为Speech与第三方应用以及操作系统之间的适配层,Moss层与APP或者MPP之间可以采用IPC方式实现交互,同样的,Moss层与Speech之间也可以通过IPC方式进行交互。

[0147] 参见图11所示,系统语音的云端在进行ASR处理之后,ASR结果可以传输至扩展语音的云端进行处理,与系统语音的云端类似,扩展语音的云端也可以利用自身的NLU能力进行语义识别,以及利用DM能力进行对话管理过程,最终可以综合两者得到的语义识别结果,采用语义仲裁的方式选择出最优选的语义识别结果,从而得到目标对象的目标意图。

[0148] 下面,结合上述的架构,以轻应用具体为小程序为例,对轻应用进行语音控制的过程进行介绍,参见图12所示,为小程序使用语音控制的流程示意图。

[0149] 步骤1201:系统语音接收到“打开XX小程序”语音指令,调用Moss层封装的服务接口,发起启动XX小程序的请求。

[0150] 步骤1202:Moss层向XX小程序发送启动请求。

[0151] 步骤1203:Moss层向车载终端设备的文本语音转换(Text To Speech,TTS)模块发起播放请求,使得TTS模块播放转换后的欢迎语句,例如播放“欢迎使用XX语音系统”。

[0152] 步骤1204:Moss层向车载终端设备的UI模块发起显示请求,使得UI模块进行try语料显示,例如显示“欢迎使用XX语音系统”。

[0153] 步骤1205:Moss层向Speech注册上述小程序的client原子能力,同时调用speech开启语音识别的接口,触发Speech内部进行语音识别准备。

[0154] 具体的,在进行原子能力注册时,Moss层从自身封装的第三方服务接口获取小程序的原子能力数据,并将原子能力数据发送给Speech,从而Speech能够基于原子能力数据进行该小程序的原子能力注册,可见,Moss层起到了中间适配层的作用,实现小程序与Speech之间的信息交互,同时,通过Moss层,还可以分离Speech和应用或者小程序之间的依赖关系,功能模块相互分离,使得流程可定位。

[0155] 步骤1206:Speech申请音频焦点,并切换内部引擎模式为语音识别(Speech

Recognition,SR) 模式。

[0156] 具体的,Speech需要申请音频焦点,以启动录音,实现音频数据的获取、传输以及在线识别等功能。例如,车载终端设备可以包括音频采集装置,如麦克风,Moss层封装有车载终端设备的操作系统提供的语音数据获取接口,则Speech可以调用语音数据获取接口向操作系统申请音频采集装置的使用权限,也就是目前Speech需要占用音频采集装置,以获取音频采集装置采集的语音组件。

[0157] 此外,为了实现语音识别的速度,除了可以申请车载终端设备的音频采集装置的使用权限之外,还可以向云端申请音频焦点,设置SR模式,可以包括传输通道的占用以及语音识别能力的占用,这样,一旦有语音数据输入,可以将其立即传输至云端进行语音识别,从而提升语音识别的速度。

[0158] 步骤1207:Speech获取语音数据后请求智平后台,智平后台通过语音识别后的文本信息发送到文本融合后台。智平后台和文本融合后台均为扩展语音的云端服务器的一部分,用于实现不同的功能,例如智平后台对应图9中Dmproxy Server模块、Dm Server模块、SkillServer模块和剧本配置模块等,文本融合后台对应NLU模块等,当然,在实际应用中,可依据实际需求进行设置。

[0159] 步骤1208:智平后台将文本信息到Speech。

[0160] 步骤1209:Moss层将Context信息传输给Speech。

[0161] 步骤1210:Speech上传Context信息至智平后台。

[0162] 本申请实施例中,Moss层可从小程序获取Context信息,包括上下文信息以及小程序的基础信息,并发送给Speech。此外,当Context信息发生变化时,也会立即上传更新。

[0163] 步骤1211:文本融合后台根据文本信息进行NLU的处理,最后输出NLU结果,NLU结果主要包含原子能力的函数名和参数,TTS文本等,并将NLU结果返回给智平后台。

[0164] 步骤1212:智平后台可以进行语义仲裁,并将仲裁后的NLU结果返回给Speech。

[0165] 需要说明的是,可以参见图12所示,智平后台将NLU结果发送给系统语音的云端后台进行语义仲裁,也可以是从系统语音的云端后台获取系统DM(包含系统语音云端识别的NLU结果)后,由智平后台进行语义仲裁。

[0166] 步骤1213:Speech基于文本信息和NLU结果进行UI显示,例如Speech可以采用流式上屏的方式将文本信息显示在设备屏幕上,即识别结果可以实时显示在设备屏幕上。需要说明的是,图12中虽未指出,但这里Speech可以通过Moss层将需要显示的内容发送给MPP进行显示。

[0167] 步骤1214:Speech基于文本信息和NLU结果请求TTS模块进行TTS播报。参见图12所示,Speech向TTS模块发起请求,以播放文本信息或者语音控制结果等。

[0168] 步骤1215:生成action后,则智平平台将action下发至Speech。

[0169] 步骤1216:Speech将action分发给Moss层。

[0170] 步骤1217:Moss层将action通知给MPP进行执行,例如,Moss层通过IPC通信方式通知MPP进行最后的原子能力函数的具体执行。

[0171] 步骤1218:Moss层接收MPP执行完返回的结果。

[0172] 步骤1219:如果涉及多轮交互,即需要feedback时,Moss层通知Speech需要feedback。

[0173] 步骤1220:Speech基于再次录入的语音数据和MPP执行结果再次请求到文本融合后台。例如,可以通过taacc传递语音数据和MPP执行结果。

[0174] 步骤1221:文本融合后台再次生成新的action发送给Speech。

[0175] 步骤1222:Speech发送新的action给Moss层。

[0176] 步骤1223:Moss层通知MPP执行新的action。

[0177] 步骤1224:Moss层接收到MPP退出的指示。

[0178] 步骤1225:Moss层通知Speech反注册client。

[0179] 步骤1226:Speech反注册client原子能力,释放音频焦点,语音识别过程结束。

[0180] 步骤1227:停止SR模式,设置为模型视图(Model View Whatever,MVW)模式。

[0181] 本申请实施例中,如果涉及多轮交互,则Speech还需将MPP执行结果再次请求到文本融合后台,MPP也会第二轮执行action直到不需要feedback。小程序执行完action后会反注册client,同时调用Speech的停止语音识别接口,Speech会根据stop函数做idle的模式切换,整个识别过程结束。

[0182] 本申请实施例中,Speech在轻应用中的应用,主要是Speech的在线语音识别功能,通过系统语音拉起扩展语音,识别后的语义结果返回到MPP中执行,从而实现打开MPP、用MPP播放、搜索以及实现播控的功能,在部分场景中会有与DM的二次交互,需要云端把执行结果传到语义后台。

[0183] 参见图13所示,为语义识别和执行的一种示例图。

[0184] 具体的,可以由Speech进行ASR识别,识别到输入的语音文本“你好,X车”(X车是指对于当前车载终端设备设置的昵称),将“你好,X车”的ASR结果上传智平后台,同时还将“你好,X车”的ASR结果进行UI显示。智平后台获得“你好,X车”对应的action后,将“你好,X车”对应的action下发给Speech,Speech调用TTS模块进行相应的语音播报。此外,Speech将“你好,X车”对应的action发送给Moss层,Moss层拉起系统语音执行action,同时,若无需feedback,则Speech释放音频焦点,并设置MVW模式。

[0185] 参见图14所示,为本申请实施例提供的意图识别的流程示意图。其中,MPP上需要实现Speech用到的context信息,例如账号、system信息、小程序前台状态以及页面文本等信息,在进行语义识别时,MPP将小程序的基本信息传入到Speech丰富落域参考信息,同时,系统语音或者扩展语音对接收的语音数据进行ASR识别,获得ASR结果。一方面,ASR结果和context信息传入系统语音进行语音识别,获得系统语音识别得到的系统语义,另一方面,ASR结果和context信息传入扩展语音的NLU server模块,NLU server模块获得context信息和ASR结果后,会去语义平台请求得到具体落域和意图,最后通过对系统语音得到的系统语义和NLU server模块输出的语义进行语义仲裁,选取最优语义结果,最后通过下发到客户端。其中,可以预先配置语义优先级,在进行语义仲裁时,则可以根据语义优先级配置选取最优语义结果,以提高语义准确性,精确保证语音正确识别,完成用户真实意图。

[0186] 本申请实施例中,Specch在目标应用中的作用为离线唤醒功能,下面,结合上述的架构,对应用进行语音唤醒的过程进行介绍,参见图15所示,为应用进行语音唤醒的流程示意图。

[0187] 步骤1501:目标应用通过Speech的接口注册“登录”、“发送消息”“取消”等唤醒词。

[0188] 步骤1502:Speech开始进行录音,并把录音数据加载到AISDK模块里的VAD子模块

和唤醒引擎中。

[0189] 步骤1503:VAD子模块进行声音的检测,判断是否触发vad start事件。

[0190] 步骤1504:当触发vad start事件时,则返回给AISDK模块vad start事件通知,包括语音开始的时间和状态。

[0191] 步骤1505:唤醒引擎对语音数据进行语音识别,并将语音识别结果流式反馈给AISDK模块。

[0192] 步骤1506:AISDK模块通过遍历已注册的唤醒词,判断语音识别结果是否包含目标应用注册过的唤醒词。

[0193] 步骤1507:若AISDK模块确定语音识别结果包含已注册的唤醒词,向Speech通知需要触发唤醒的结果。

[0194] 步骤1508:Speech将唤醒的文本结果通知给目标应用做相应的处理。

[0195] 步骤1509:语音数据录入完毕后,AISDK模块通知VAD子模块重置其状态。

[0196] 综上所述,本申请实施例中,提出了一种车载语音架构,支持APP唤醒、语音识别、语义识别等功能,能够让APP和小程序等其他业务高效接入后,提供语音控制功能,该方案可以精确保证语音正确识别,精准完成真实意图,方便业务方友好接入,分离出与业务无关的VFramework层,UI和逻辑层分离,分离语音和应用的依赖关系,功能模块分离,流程可定位,流式显示语音识别结果,减少用户等待时间,提高体验感。此外,提供了完整的客户端和云端一体架构,灵活可控,结合自身业务定制需求,对端上信息利用的更充分,对用户的真实意图把控更准确。

[0197] 请参见图16,基于同一发明构思,本申请实施例还提供了一种语音控制装置160,应用于语音控制系统中,该装置包括:

能力注册单元1601,用于响应于目标轻应用的语音控制功能的启动,从目标轻应用获取原子能力数据,并基于原子能力数据进行原子能力注册,其中,原子能力数据包含:目标轻应用所能提供的原子能力,每个原子能力用于实现目标轻应用的至少一个功能;

事件转换单元1602,用于响应于目标对象针对目标轻应用输入的第一语音数据,基于已注册的各原子能力,将第一语音数据转换为相应的语音控制事件,语音控制事件包含:为实现第一语音数据的目标意图,目标轻应用需要调用的至少一个原子能力;

传输单元1603,用于将语音控制事件发送给目标轻应用,以使得目标轻应用调用至少一个原子能力来实现目标意图。

[0198] 可选的,事件转换单元1602,具体用于:

对第一语音数据进行语音识别,获取第一语音数据中包含的文本信息;

对文本信息进行语义识别,确定文本信息对应的目标意图;

基于预设的事件配置信息以及已注册的各原子能力,将目标意图转换为语音控制事件,其中,事件配置信息包含:基于目标意图对至少一个原子能力进行参数配置的指导信息。

[0199] 可选的,事件转换单元1602,具体用于:

从目标轻应用获取应用描述信息,应用描述信息包含:目标轻应用的基本描述信息和语音控制上下文信息中的至少一种;

基于应用描述信息,对文本信息进行语义识别,确定目标意图。

- [0200] 可选的,事件转换单元1602,具体用于:  
将原子能力数据发送给云端服务器,以在云端服务器进行各原子能力的注册;  
从目标轻应用获取应用描述信息,应用描述信息包含:目标轻应用的基本描述信息和语音控制上下文信息中的至少一种;  
将应用描述信息以及第一语音数据发送给云端服务器,并接收云端服务器返回的语音控制事件,其中,语音控制事件为云端服务器基于应用描述信息以及第一语音数据确定目标意图后,基于已注册的原子能力对目标意图进行转换得到的。
- [0201] 可选的,语音控制系统包括接口组件和语音组件,接口组件封装了目标轻应用提供的第三方服务接口;则能力注册单元1601,具体用于:  
响应于目标轻应用的语音控制功能的启动,触发接口组件调用第三方服务接口获取原子能力数据;  
通过接口组件向语音组件发送原子能力数据,以在语音组件进行原子能力注册。
- [0202] 可选的,终端设备包括音频采集装置,接口组件封装有终端设备的操作系统提供的语音数据获取接口;则装置还包括申请单元1604,用于:  
调用语音组件通过语音数据获取接口,向操作系统申请音频采集装置的使用权限;  
调用语音组件通过语音数据获取接口,接收音频采集装置采集的第一语音数据。
- [0203] 可选的,事件转换单元1602,还用于:  
接收目标轻应用执行语音控制事件后返回的执行结果;  
若执行结果指示需要进行反馈控制,则启动录音功能,获取目标对象输入的第二语音数据;  
基于已注册的各原子能力,将第二语音数据转换为相应的语音控制事件,并将获得的语音控制事件发送给目标轻应用执行。
- [0204] 可选的,事件转换单元1602,还用于:  
若执行结果指示不需要进行反馈控制,则清空已注册的原子能力;  
通知操作系统释放自身占用的音频采集装置的使用权限。
- [0205] 可选的,该装置还包括唤醒单元1605,用于:  
响应于目标对象针对目标应用输入的第三语音数据,对第三语音数据进行语音端点检测;  
若检测到第三语音数据中的语音开始位置,则从语音开始位置开始,对第三语音数据进行语音识别,直至检测到第三语音数据中的语音结束位置;  
基于获得的语音识别结果确定对目标应用进行唤醒时,对目标应用进行唤醒操作。
- [0206] 可选的,唤醒单元1605,具体用于:  
响应于目标应用的语音唤醒功能的启动,获取目标应用中激活页面对应的唤醒词集合,并基于唤醒词集合进行唤醒词注册;  
若语音识别结果包含已注册的唤醒词,则确定对目标应用进行唤醒;  
对目标应用进行唤醒操作,并将语音识别结果包含的文本信息发送给目标应用。
- [0207] 该装置可以用于执行本申请各实施例中所示的方法,因此,对于该装置的各功能

模块所能够实现的功能等可参考前述实施例的描述,不多赘述。

[0208] 请参见图17,基于同一技术构思,本申请实施例还提供了一种计算机设备。在一种实施例中,该计算机设备可以为图1所示的云端服务器,该计算机设备如图17所示,包括存储器1701,通讯模块1703以及一个或多个处理器1702。

[0209] 存储器1701,用于存储处理器1702执行的计算机程序。存储器1701可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统,以及运行即时通讯功能所需的程序等;存储数据区可存储各种即时通讯信息和操作指令集等。

[0210] 存储器1701可以是易失性存储器(volatile memory),例如随机存取存储器(random-access memory,RAM);存储器1701也可以是非易失性存储器(non-volatile memory),例如只读存储器,快闪存储器(flash memory),硬盘(hard disk drive,HDD)或固态硬盘(solid-state drive,SSD);或者存储器1701是能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质,但不限于此。存储器1701可以是上述存储器的组合。

[0211] 处理器1702,可以包括一个或多个中央处理单元(central processing unit,CPU)或者为数字处理单元等等。处理器1702,用于调用存储器1701中存储的计算机程序时实现上述语音控制方法。

[0212] 通讯模块1703用于与终端设备和其他服务器进行通信。

[0213] 本申请实施例中不限定上述存储器1701、通讯模块1703和处理器1702之间的具体连接介质。本申请实施例在图17中以存储器1701和处理器1702之间通过总线1704连接,总线1704在图17中以粗线描述,其它部件之间的连接方式,仅是进行示意性说明,并不引以为限。总线1704可以分为地址总线、数据总线、控制总线等。为便于描述,图17中仅用一条粗线描述,但并不描述仅有一根总线或一种类型的总线。

[0214] 存储器1701中存储有计算机存储介质,计算机存储介质中存储有计算机可执行指令,计算机可执行指令用于实现本申请实施例的语音控制方法。处理器1702用于执行上述各实施例的语音控制方法。

[0215] 在另一种实施例中,计算机设备也可以是其他计算机设备,如图1所示的车载终端设备。在该实施例中,计算机设备的结构可以如图18所示,包括:通信组件1810、存储器1820、显示单元1830、摄像头1840、传感器1850、音频电路1860、蓝牙模块1870、处理器1880等部件。

[0216] 通信组件1810用于与服务器进行通信。在一些实施例中,可以包括电路无线保真(Wireless Fidelity,WiFi)模块,WiFi模块属于短距离无线传输技术,计算机设备通过WiFi模块可以帮助用户收发信息。

[0217] 存储器1820可用于存储软件程序及数据。处理器1880通过运行存储在存储器1820的软件程序或数据,从而执行终端设备的各种功能以及数据处理。存储器1820可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。存储器1820存储有使得终端设备能运行的操作系统。本申请中存储器1820可以存储操作系统及各种应用程序,还可以存储执行本申请实施例语音控制方法的代码。

[0218] 显示单元1830还可用于显示由用户输入的信息或提供给用户的信息以及终端设

备的各种菜单的图形用户界面(graphical user interface,GUI)。具体地,显示单元1830可以包括设置在终端设备正面的显示屏1832。其中,显示屏1832可以采用液晶显示器、发光二极管等形式来配置。显示单元1830可以用于显示本申请实施例中的各种语音控制页面以及车载的相关显示页面,例如小程序页面或者应用页面等。

[0219] 显示单元1830还可用于接收输入的数字或字符信息,产生与终端设备的用户设置以及功能控制有关的信号输入,具体地,显示单元1830可以包括设置在终端设备正面的触摸屏1831,可收集用户在其上或附近的触摸操作,例如点击按钮,拖动滚动框等。

[0220] 其中,触摸屏1831可以覆盖在显示屏1832之上,也可以将触摸屏1831与显示屏1832集成而实现终端设备的输入和输出功能,集成后可以简称触摸显示屏。本申请中显示单元1830可以显示应用程序以及对应的操作步骤。

[0221] 摄像头1840可用于捕获静态图像,用户可以将摄像头1840拍摄的图像通过应用发布评论。摄像头1840可以是一个,也可以是多个。物体通过镜头生成光学图像投射到感光元件。感光元件可以是电荷耦合器件(charge coupled device,CCD)或互补金属氧化物半导体(complementary metal-oxide-semiconductor,CMOS)光电晶体管。感光元件把光信号转换成电信号,之后将电信号传递给处理器1880转换成数字图像信号。

[0222] 终端设备还可以包括至少一种传感器1850,比如加速度传感器1851、距离传感器1852、指纹传感器1853、温度传感器1854。终端设备还可配置有陀螺仪、气压计、湿度计、温度计、红外线传感器、光传感器、运动传感器等其他传感器。

[0223] 音频电路1860、扬声器1861、传声器1862可提供用户与终端设备之间的音频接口。音频电路1860可将接收到的音频数据转换后的电信号,传输到扬声器1861,由扬声器1861转换为声音信号输出。终端设备还可配置音量按钮,用于调节声音信号的音量。另一方面,传声器1862将收集的声音信号转换为电信号,由音频电路1860接收后转换为音频数据,再将音频数据输出至通信组件1810以发送给比如另一终端设备,或者将音频数据输出至存储器1820以便进一步处理。

[0224] 蓝牙模块1870用于通过蓝牙协议来与其他具有蓝牙模块的蓝牙设备进行信息交互。例如,终端设备可以通过蓝牙模块1870与同样具备蓝牙模块的可穿戴计算机设备(例如智能手表)建立蓝牙连接,从而进行数据交互。

[0225] 处理器1880是终端设备的控制中心,利用各种接口和线路连接整个终端的各个部分,通过运行或执行存储在存储器1820内的软件程序,以及调用存储在存储器1820内的数据,执行终端设备的各种功能和处理数据。在一些实施例中,处理器1880可包括一个或多个处理单元;处理器1880还可以集成应用处理器和基带处理器,其中,应用处理器主要处理操作系统、用户界面和应用程序等,基带处理器主要处理无线通信。可以理解的是,上述基带处理器也可以不集成到处理器1880中。本申请中处理器1880可以运行操作系统、应用程序、用户界面显示及触控响应,以及本申请实施例的语音控制方法。另外,处理器1880与显示单元1830耦接。

[0226] 在一些可能的实施方式中,本申请提供的语音控制方法的各个方面还可以实现为一种程序产品的形式,其包括程序代码,当程序产品在计算机设备上运行时,程序代码用于使计算机设备执行本说明书上述描述的根据本申请各种示例性实施方式的语音控制方法中的步骤,例如,计算机设备可以执行各实施例的步骤。

[0227] 程序产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以是但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPR0M或闪存)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。

[0228] 本申请的实施方式的程序产品可以采用便携式紧凑盘只读存储器(CD-ROM)并包括程序代码,并可以在计算装置上运行。然而,本申请的程序产品不限于此,在本申请件中,可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被命令执行系统、装置或者器件使用或者与其结合使用。

[0229] 可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了可读程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。可读信号介质还可以是可读存储介质以外的任何可读介质,该可读介质可以发送、传播或者传输用于由命令执行系统、装置或者器件使用或者与其结合使用的程序。

[0230] 可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于无线、有线、光缆、RF等等,或者上述的任意合适的组合。

[0231] 可以以一种或多种程序设计语言的任意组合来编写用于执行本申请操作的程序代码,程序设计语言包括面向对象的程序设计语言—诸如Java、C++等,还包括常规的过程式程序设计语言—诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算装置上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户计算装置上部分在远程计算装置上执行、或者完全在远程计算装置或服务器上执行。在涉及远程计算装置的情形中,远程计算装置可以通过任意种类的网络包括局域网(LAN)或广域网(WAN)连接到用户计算装置,或者,可以连接到外部计算装置(例如利用因特网服务提供商来通过因特网连接)。

[0232] 应当注意,尽管在上文详细描述中提及了装置的若干单元或子单元,但是这种划分仅仅是示例性的并非强制性的。实际上,根据本申请的实施方式,上文描述的两个或更多单元的特征和功能可以在一个单元中具体化。反之,上文描述的一个单元的特征和功能可以进一步划分为由多个单元来具体化。

[0233] 此外,尽管在附图中以特定顺序描述了本申请方法的操作,但是,这并非要求或者暗示必须按照该特定顺序来执行这些操作,或是必须执行全部所示的操作才能实现期望的结果。附加地或备选地,可以省略某些步骤,将多个步骤合并为一个步骤执行,和/或将一个步骤分解为多个步骤执行。

[0234] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0235] 尽管已描述了本申请的优选实施例,但本领域内的技术人员一旦得知了基本创造

性概念,则可对这些实施例做出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本申请范围的所有变更和修改。

[0236] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

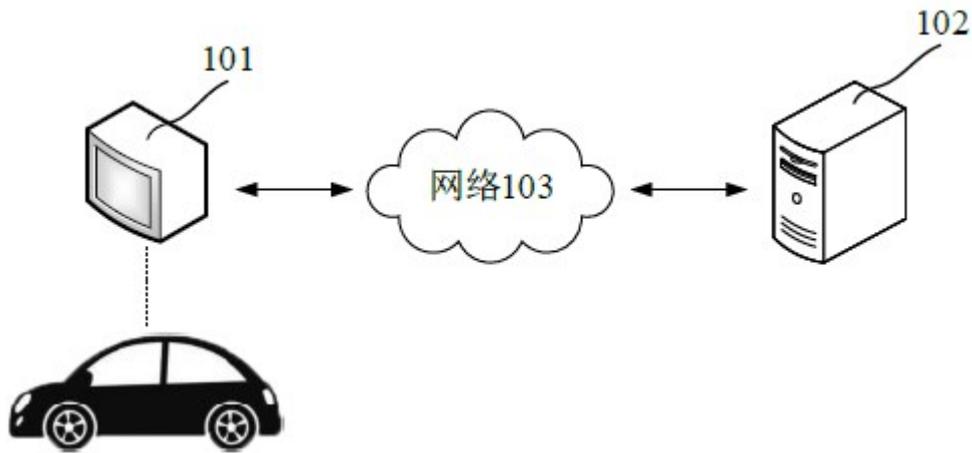


图1

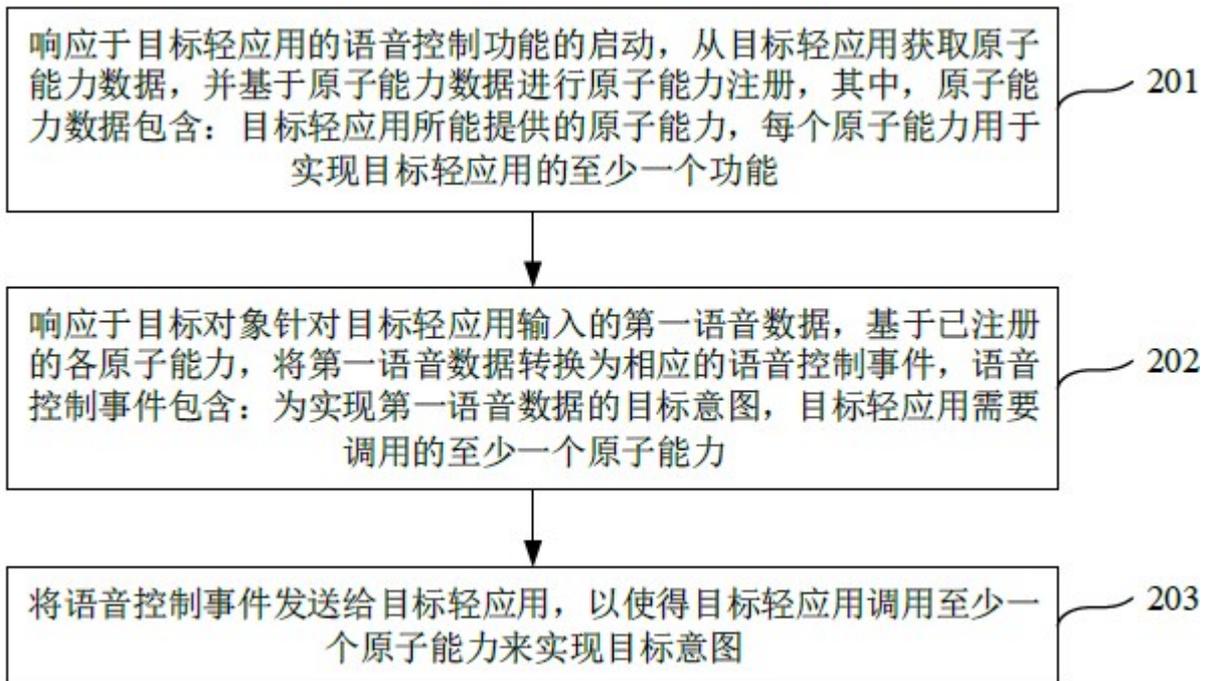


图2

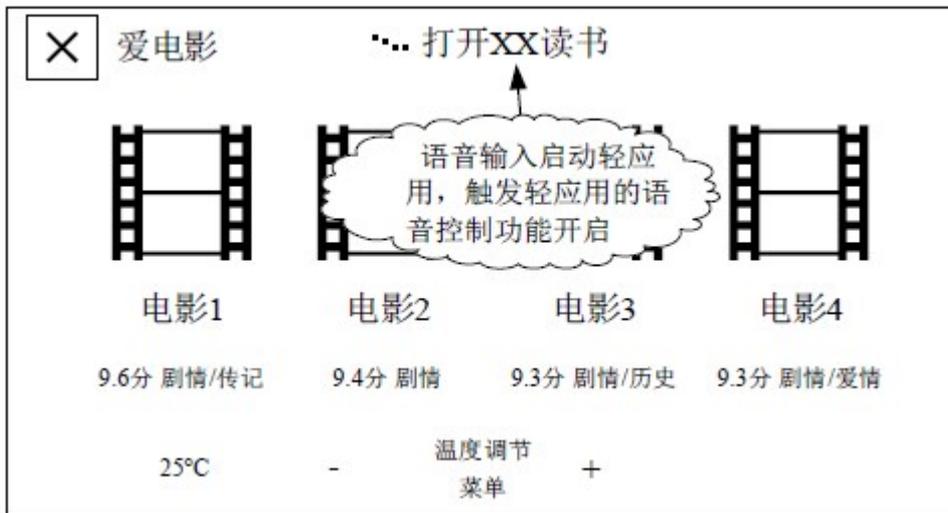


图3A

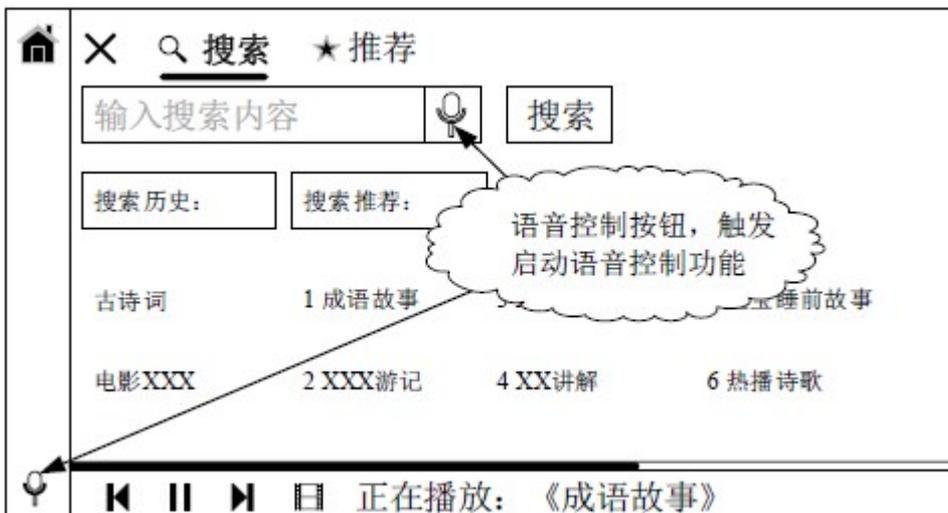


图3B

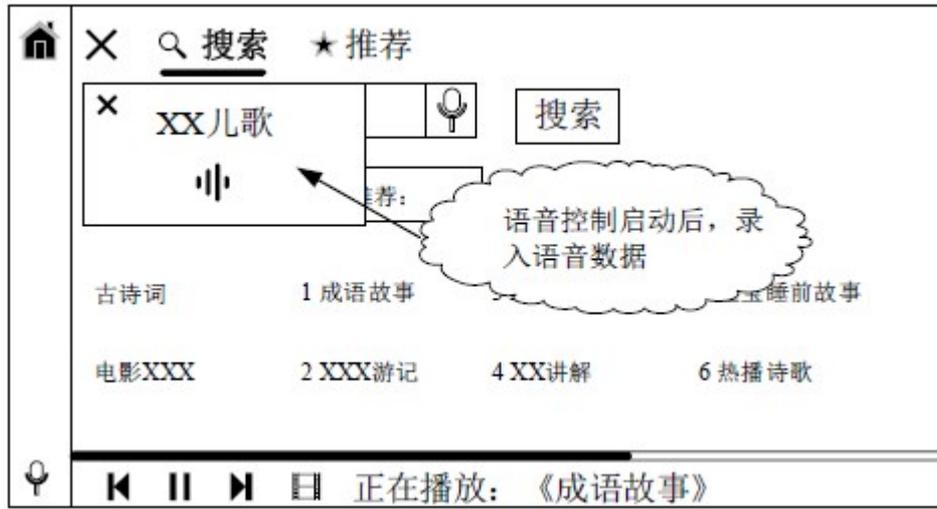


图3C

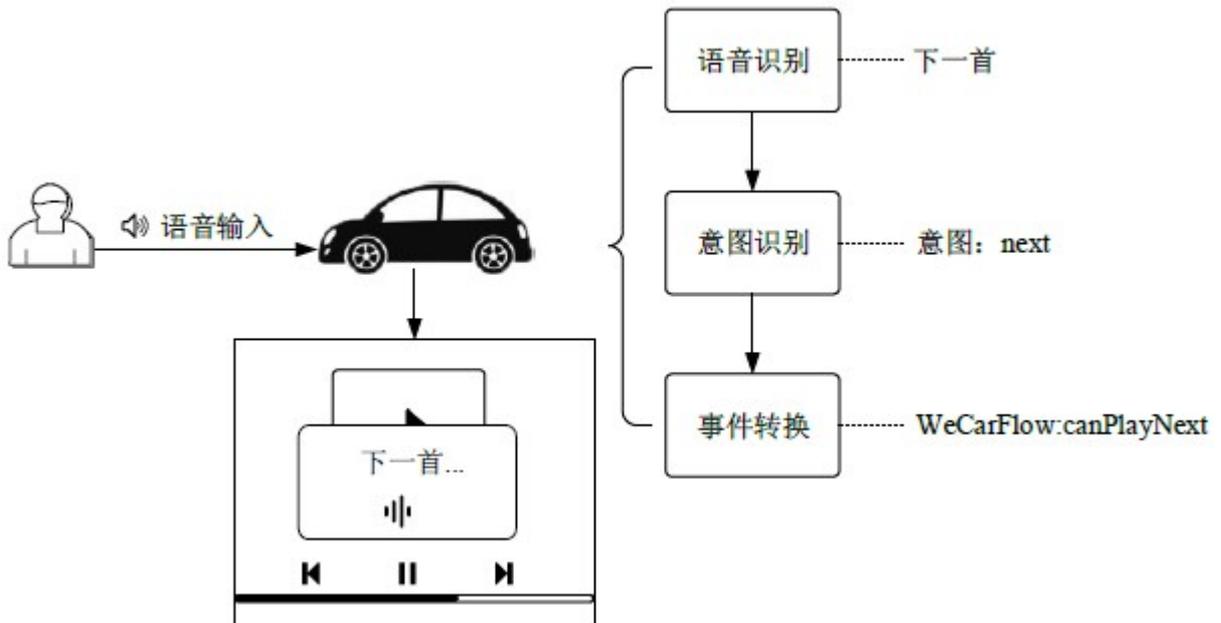


图4

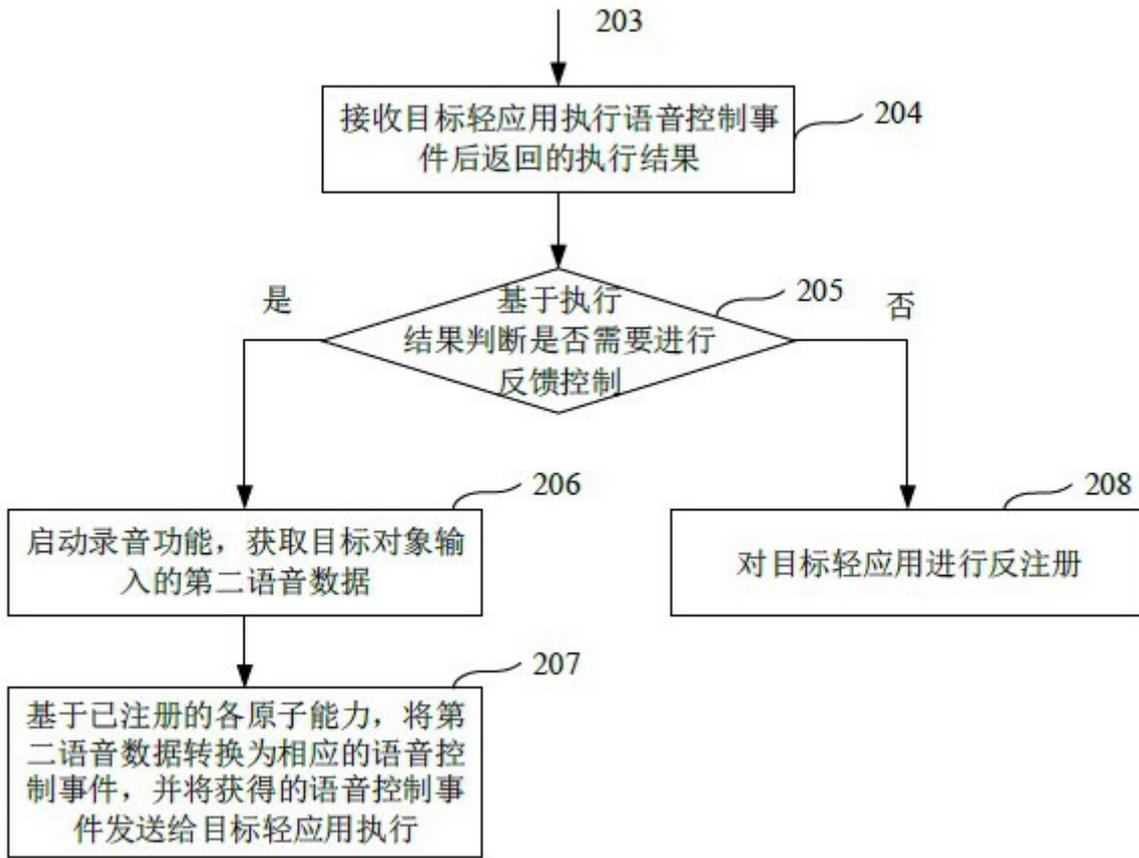


图5

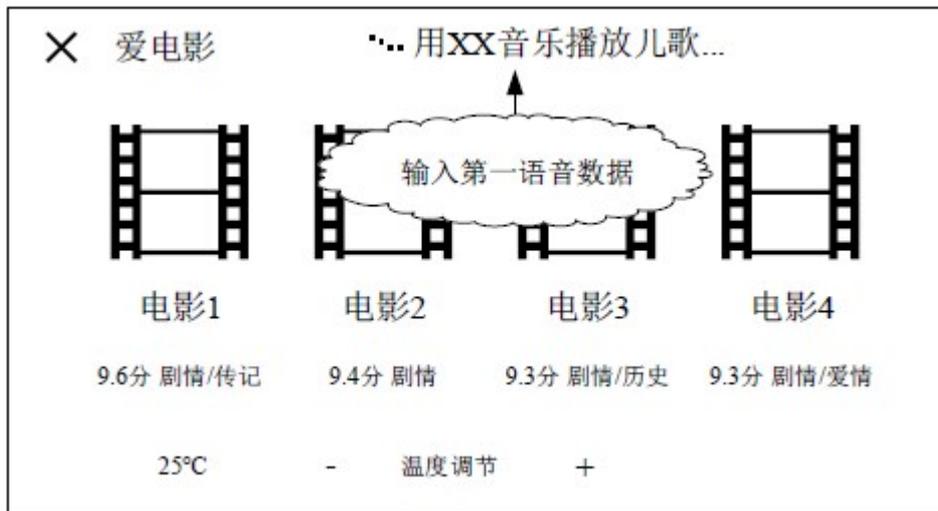


图6A

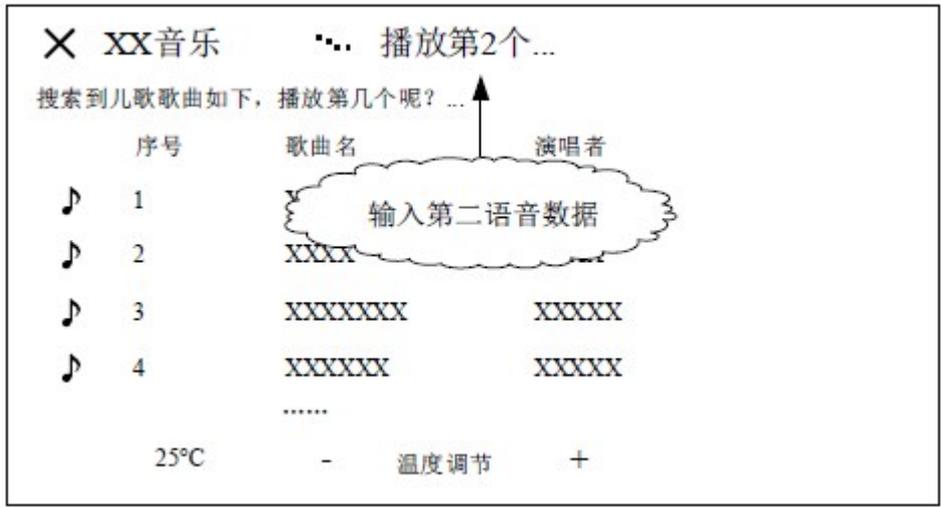


图6B

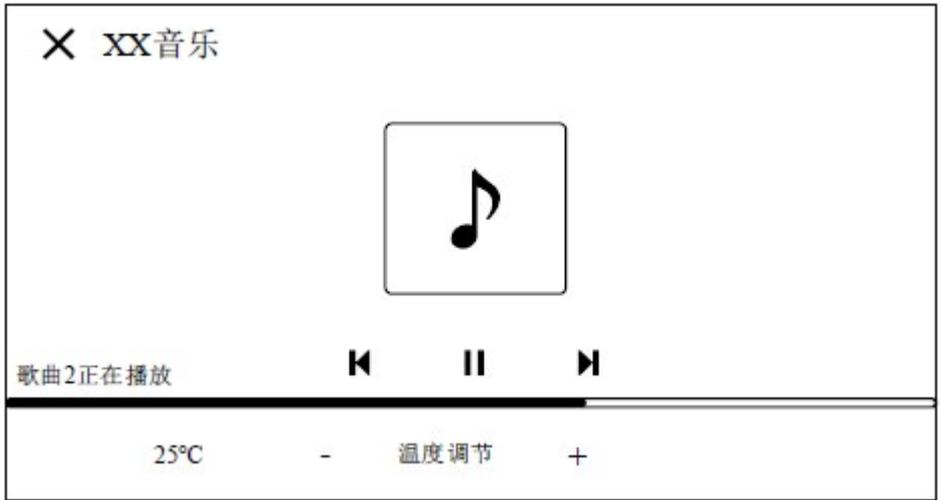


图6C

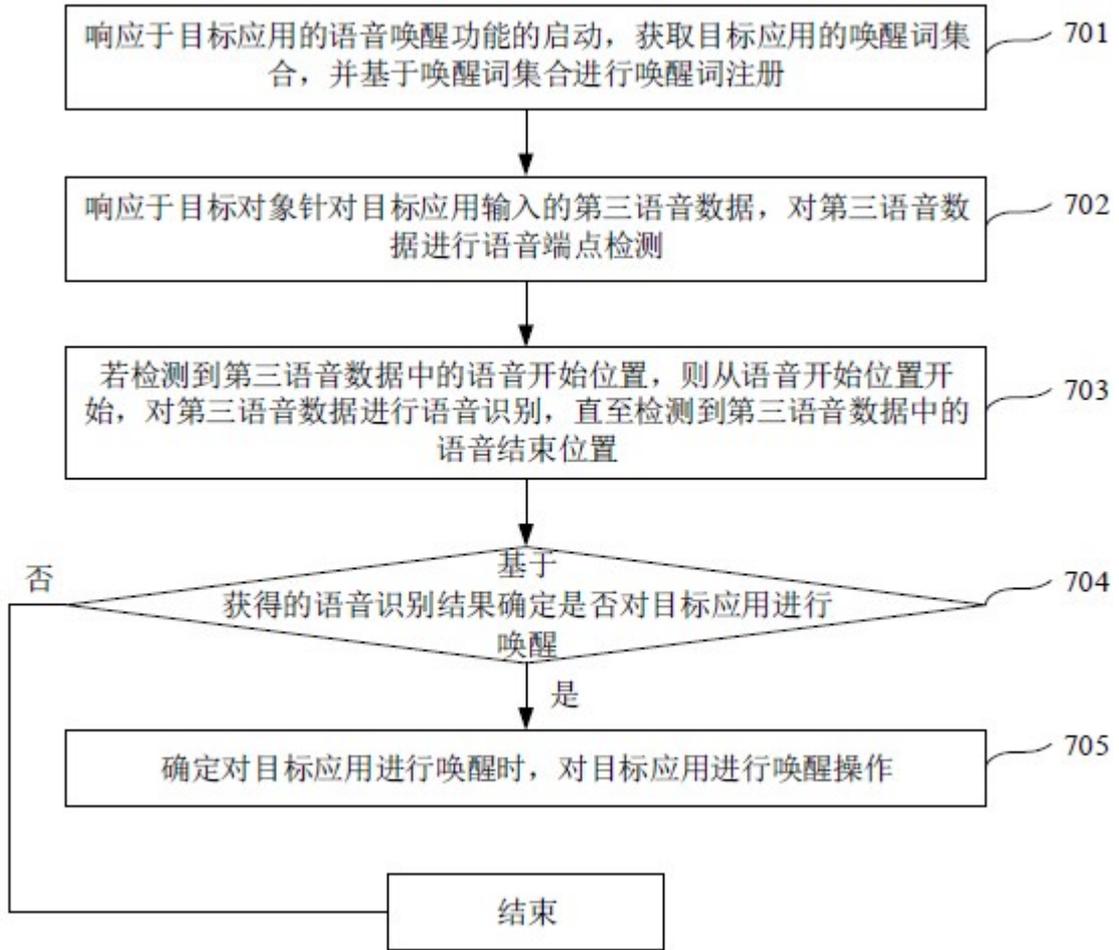


图7

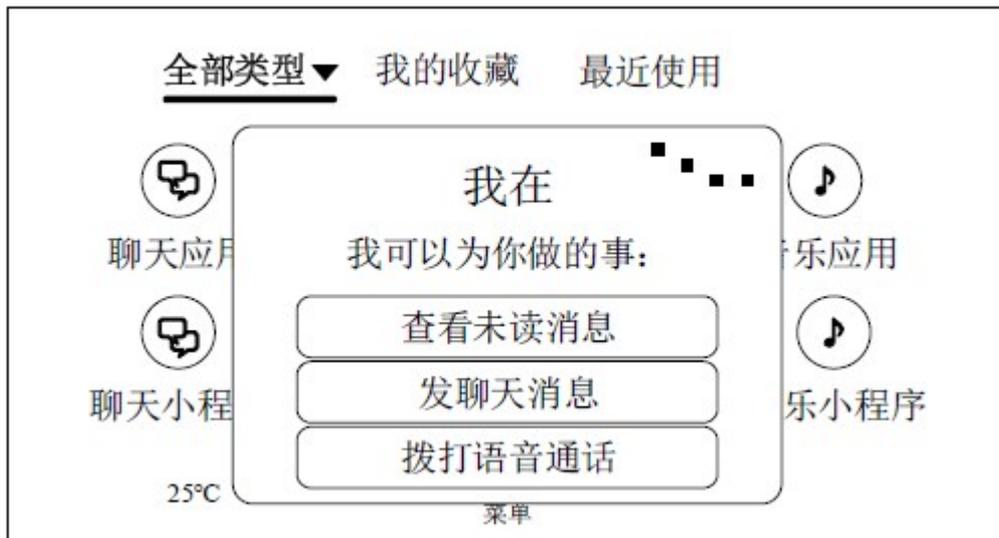


图8A

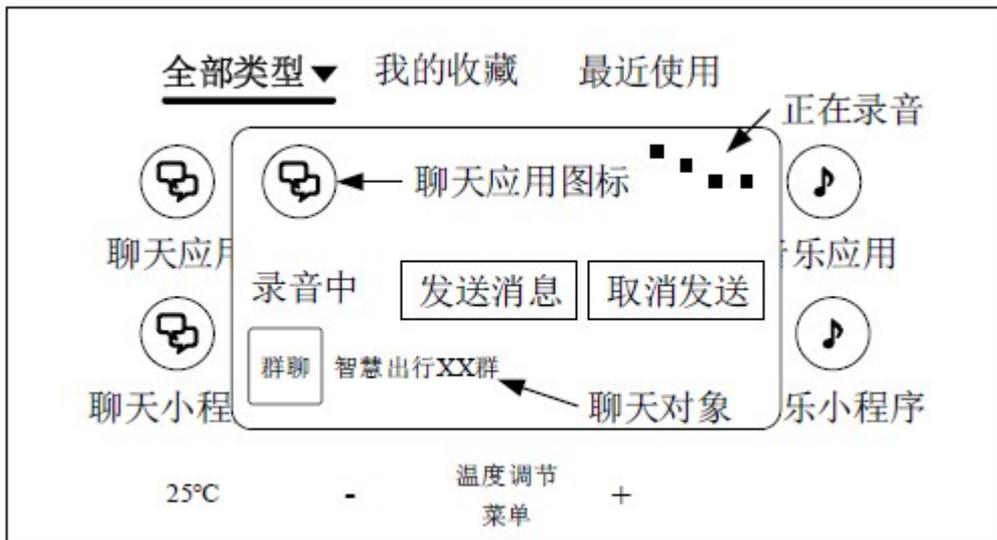


图8B

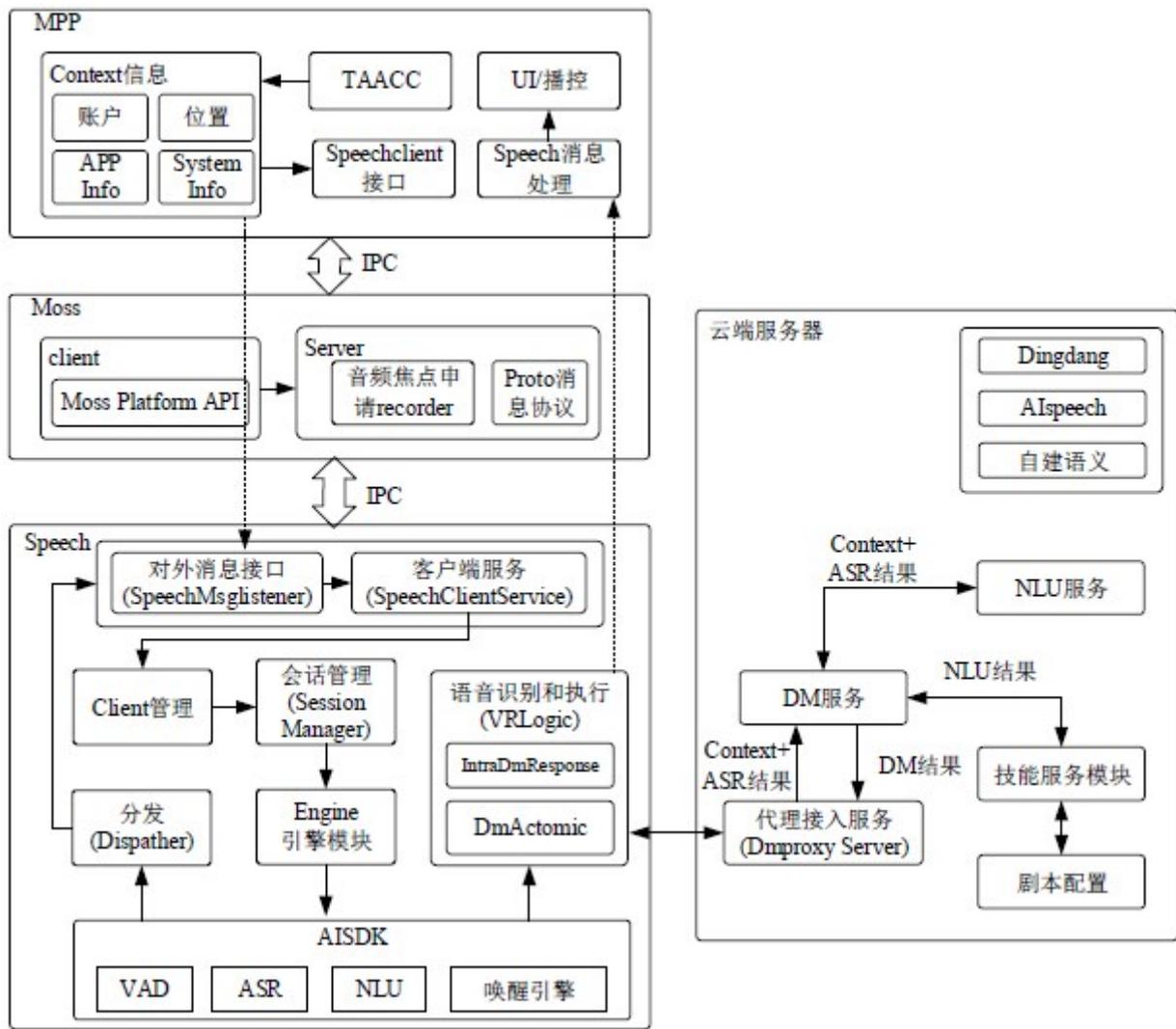


图9

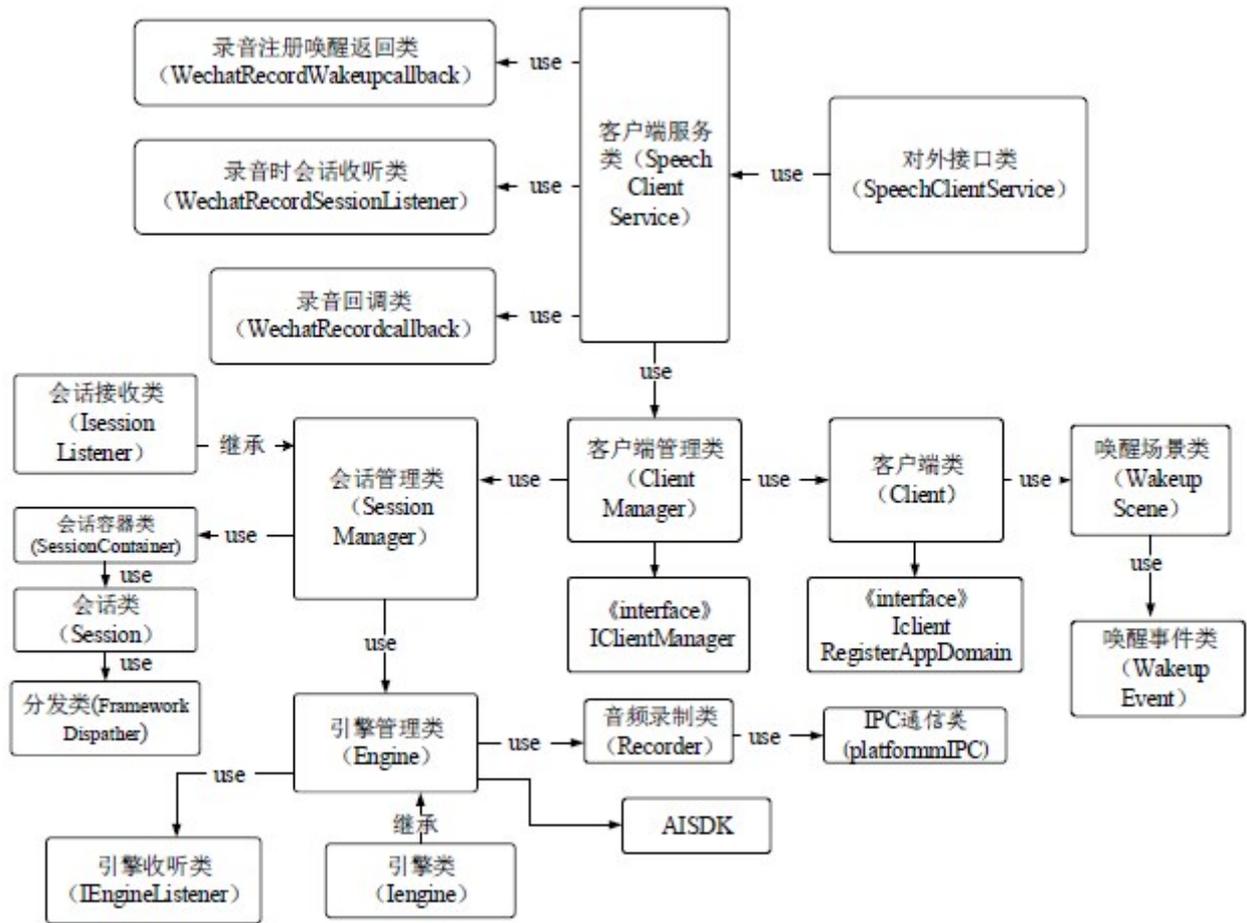


图10

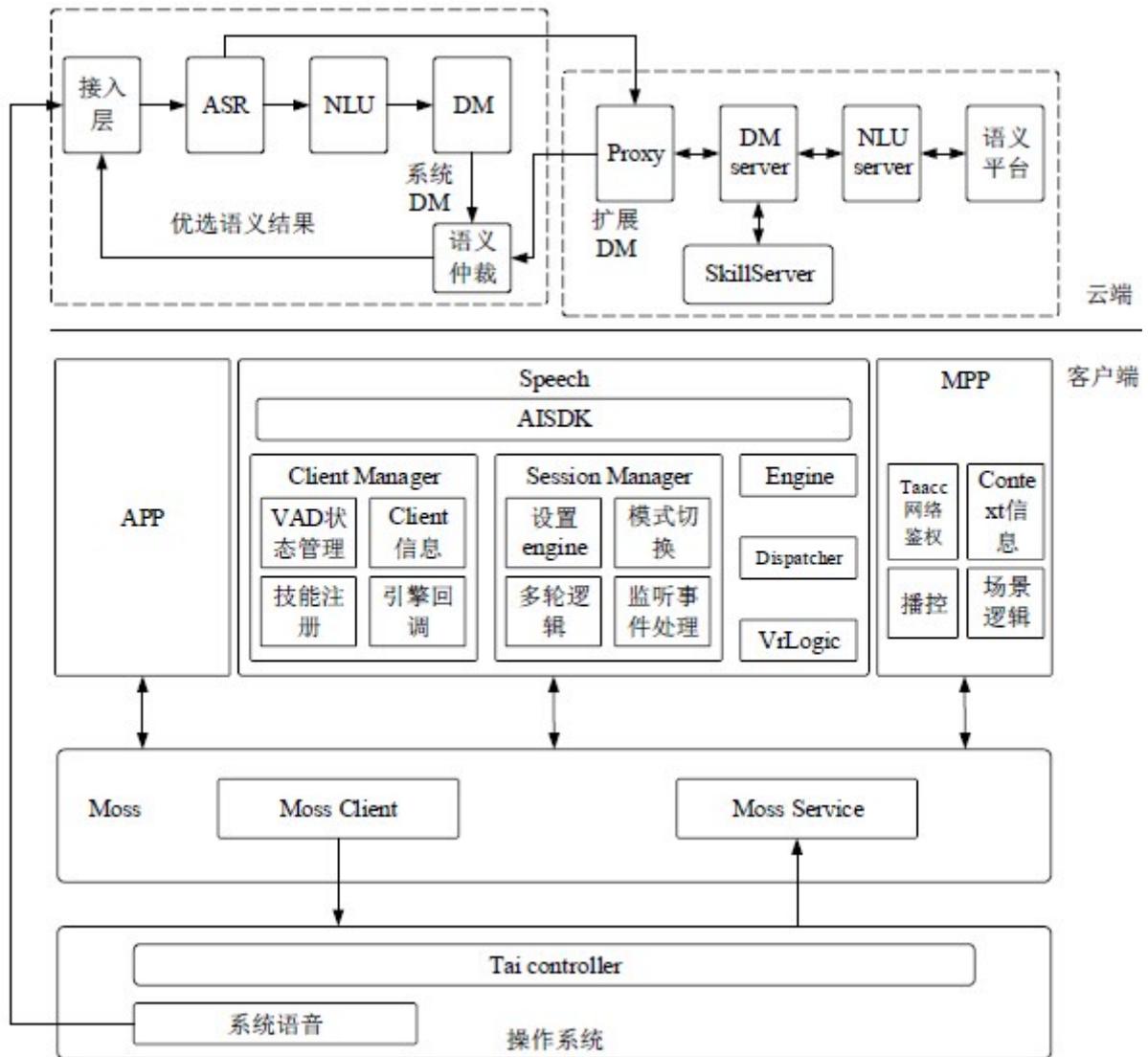


图11

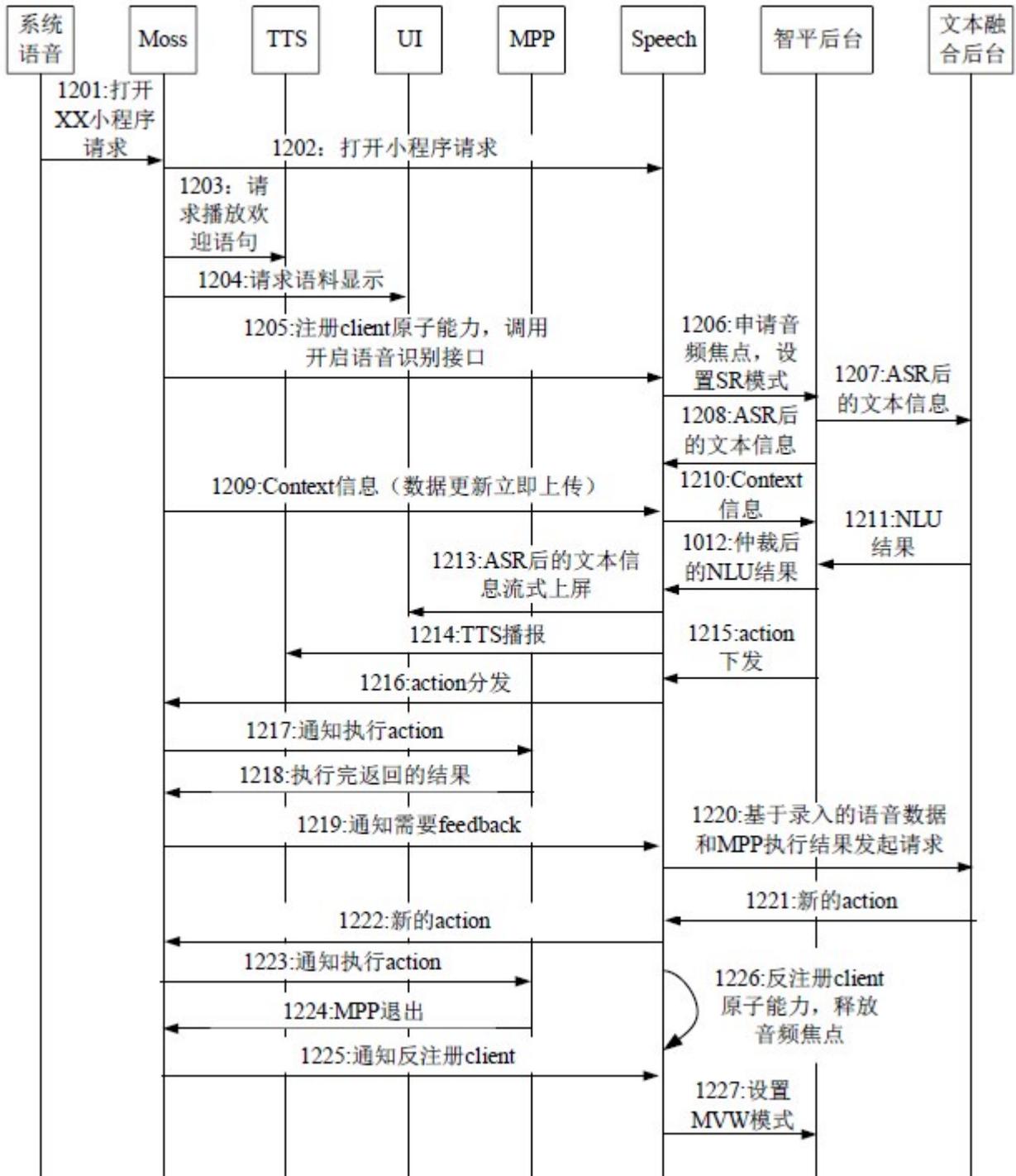


图12

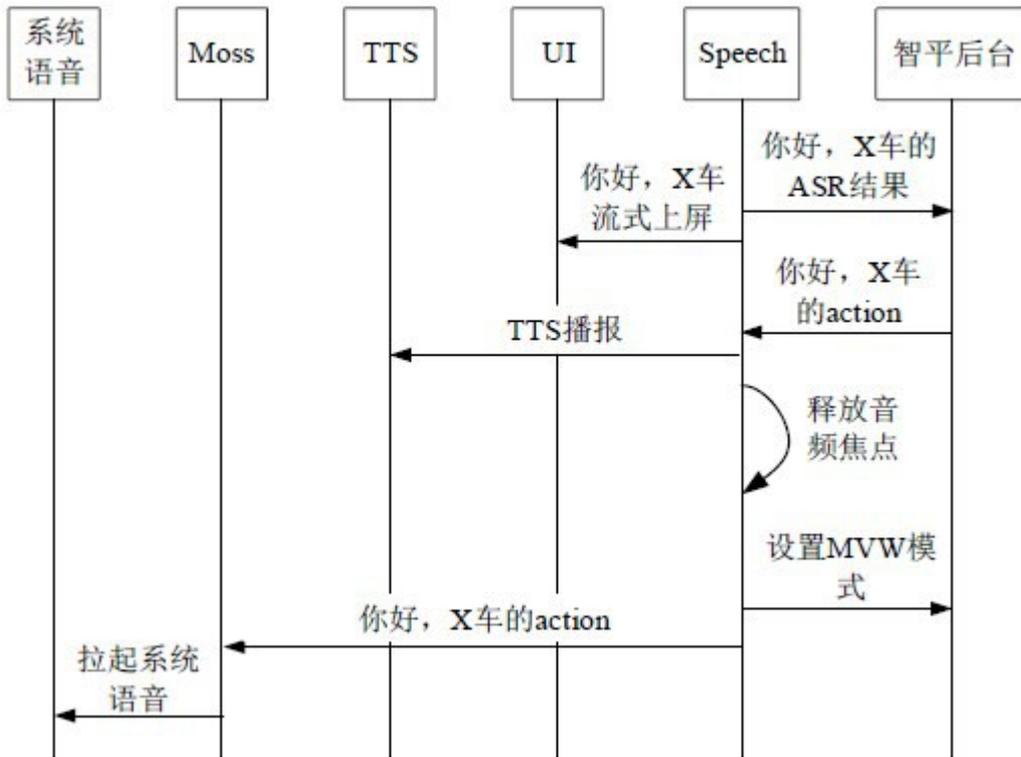


图13

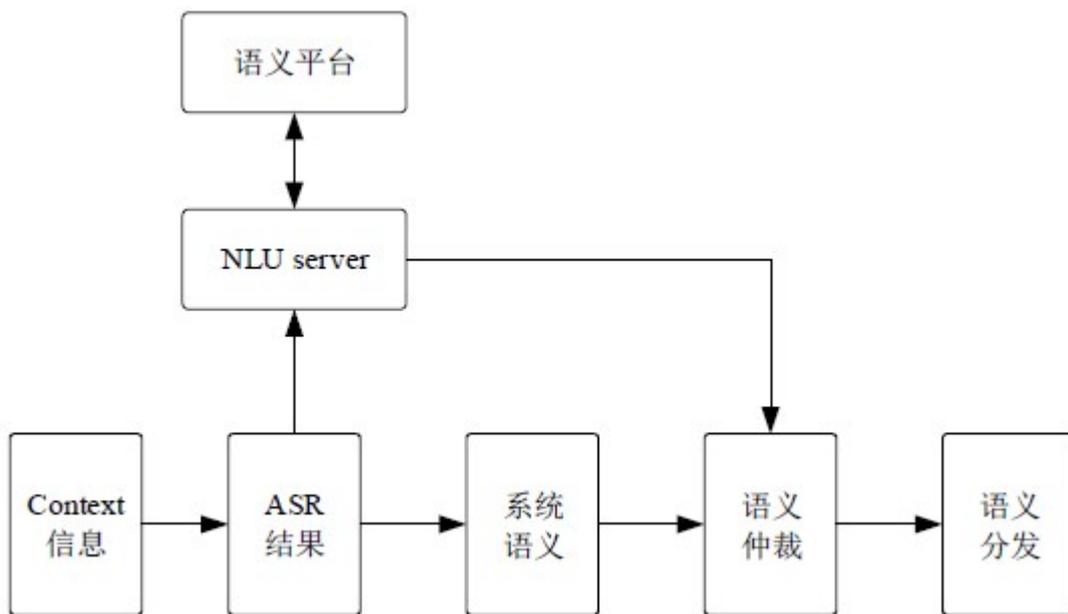


图14

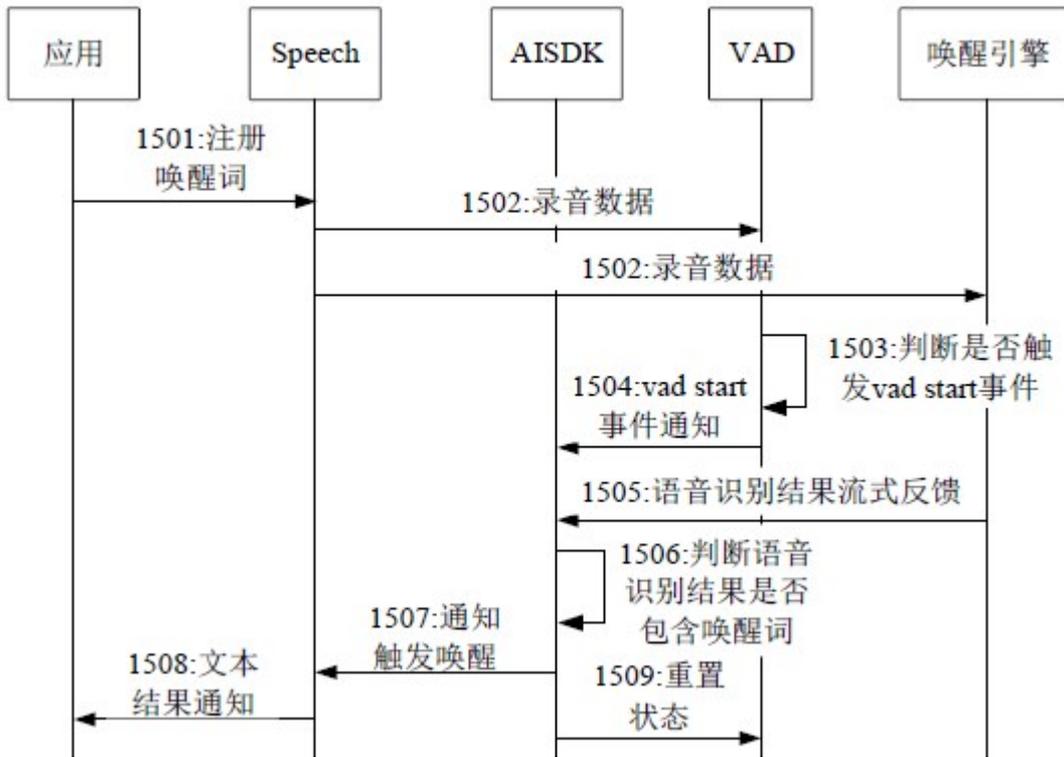


图15

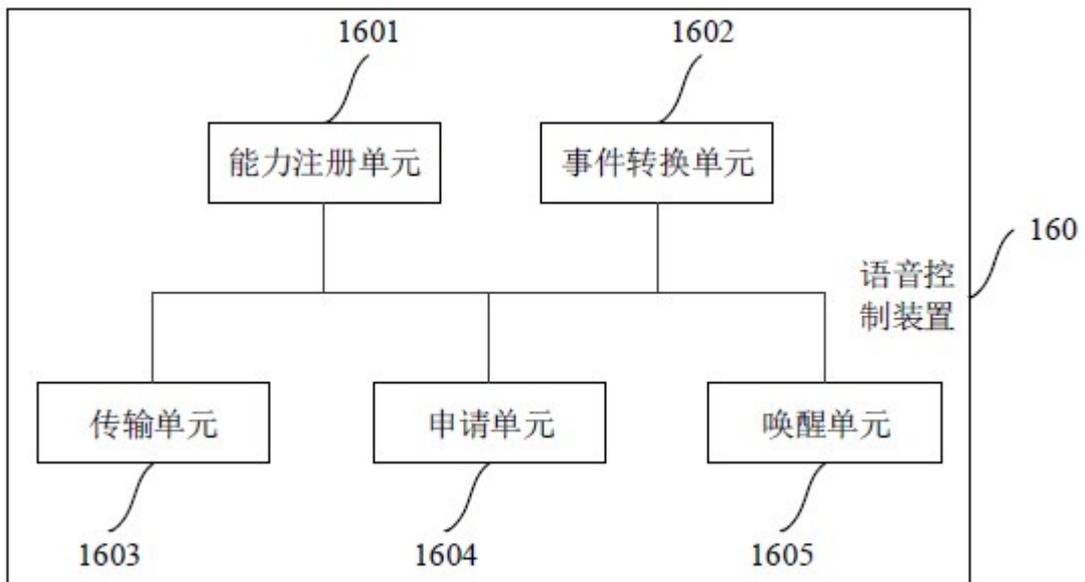


图16

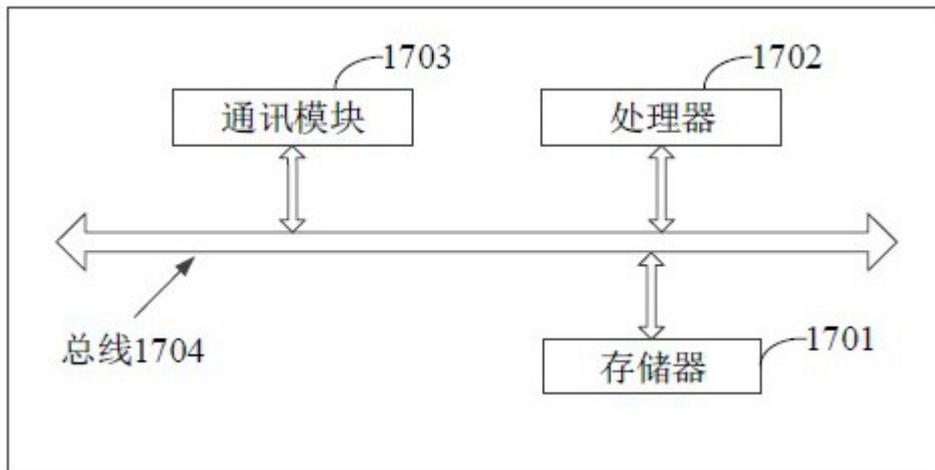


图17

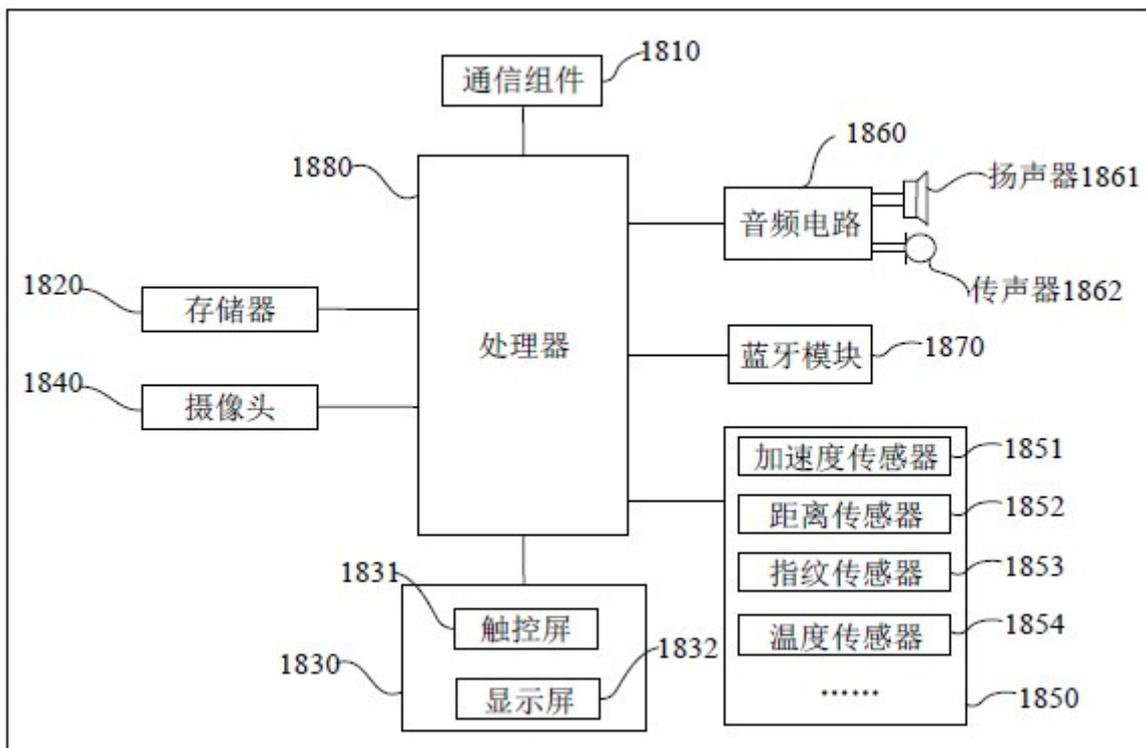


图18