



(12)发明专利申请

(10)申请公布号 CN 111045749 A

(43)申请公布日 2020.04.21

(21)申请号 201911307685.3

(22)申请日 2019.12.18

(71)申请人 青岛海信传媒网络技术有限公司
地址 266061 山东省青岛市崂山区香港东路248号131室

(72)发明人 张小涛 李帅 武兵

(74)专利代理机构 北京同立钧成知识产权代理有限公司 11205
代理人 袁义科 刘芳

(51)Int.Cl.
G06F 9/445(2018.01)
G06F 16/958(2019.01)

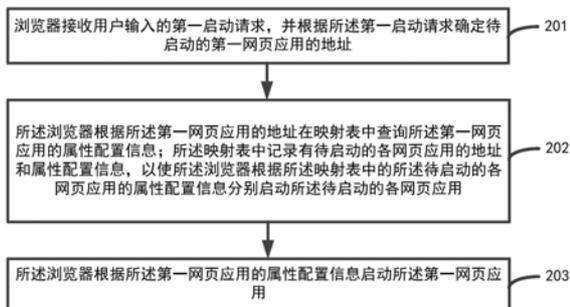
权利要求书2页 说明书12页 附图8页

(54)发明名称

应用程序启动方法及设备

(57)摘要

本发明实施例提供一种应用程序启动方法及设备,该方法包括浏览器接收用户输入的第一启动请求,并根据该第一启动请求确定待启动的第一网页应用的地址;浏览器根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息;所述映射表中记录有待启动的各网页应用的地址和属性配置信息,以使所述浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用;浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用。本发明实施例能够在单个浏览器中启动多个单页Web应用,从而降低了所占用的内存空间,节约了硬件成本,并且减少了代码维护工作,提高工作效率的同时也降低了人工成本。



1. 一种应用程序启动方法,其特征在于,包括:

浏览器接收用户输入的第一启动请求,并根据所述第一启动请求确定待启动的第一网页应用的地址;

所述浏览器根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息;所述映射表中记录有待启动的各网页应用的地址和属性配置信息,以使所述浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用;

所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用。

2. 根据权利要求1所述的方法,其特征在于,所述待启动的各网页应用中包括用户界面应用;所述浏览器接收用户输入的第一启动请求之前,还包括:

所述浏览器启动后在映射表中查询用户界面应用的属性配置信息,并根据所述用户界面应用的属性配置信息启动所述用户界面应用。

3. 根据权利要求2所述的方法,其特征在于,所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用之后,还包括:

所述浏览器根据所述第一网页应用的地址,对当前应用的记录信息进行更新。

4. 根据权利要求2所述的方法,其特征在于,所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用之后,还包括:

所述浏览器的按键管理模块接收应用管理器发送的第一按键信息;

所述浏览器的按键管理模块将所述第一按键信息发送给所述用户界面应用以使所述用户界面应用对所述第一按键信息进行处理,得到第一处理结果并返回;

所述浏览器的按键管理模块接收所述第一处理结果,并对所述第一处理结果进行判断;

若所述第一处理结果为所述第一按键信息已被执行,则将所述第一处理结果返回给所述应用管理器。

5. 根据权利要求4所述的方法,其特征在于,所述浏览器的按键管理模块接收所述第一处理结果,并对所述第一处理结果进行判断之后,还包括:

若所述第一按键信息未被执行,则所述浏览器的按键管理模块将所述第一按键信息发送给当前应用,以使所述当前应用对所述第一按键信息进行处理,得到第二处理结果并返回;

所述浏览器的按键管理模块接收所述第二处理结果,并对所述第二处理结果进行判断;

若所述第二处理结果为所述第一按键信息已被执行,则将所述第二处理结果返回给所述应用管理器。

6. 根据权利要求5所述的方法,其特征在于,所述浏览器的按键管理模块接收所述第二处理结果,并对所述第二处理结果进行判断之后,还包括:

若所述第二处理结果为所述第一按键信息未被执行,则所述浏览器的按键管理模块根据预设应用列表,将所述第一按键信息依次发送给所述预设应用列表中各应用进行处理以使当前对所述第一按键信息进行处理的应用返回处理结果,直至接收到所述预设应用列表中最后一个应用返回的第三处理结果,或者当前对所述第一按键信息进行处理的应用返回

所述第一按键信息已被执行的第四处理结果;其中,所述预设应用列表包括已启动的各网页应用的地址;将处理结果发送给应用管理器;

所述浏览器的按键管理模块将所述第三处理结果或所述第四处理结果返回给所述应用管理器。

7. 根据权利要求1-6任一项所述的方法,其特征在于,所述浏览器接收用户输入的第一启动请求之前,还包括:

根据待启动的各网页应用的地址和属性配置信息,创建所述映射表,并存储于所述浏览器中。

8. 一种应用程序启动设备,其特征在于,包括:

接收模块,用于接收用户输入的第一启动请求,并根据所述第一启动请求确定待启动的第一网页应用的地址;

查询模块,用于根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息;所述映射表中记录有待启动的各网页应用的地址和属性配置信息,以使浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用;

第一启动模块,用于根据所述第一网页应用的属性配置信息启动所述第一网页应用。

9. 一种应用程序启动设备,其特征在于,包括:至少一个处理器和存储器;

所述存储器存储计算机执行指令;

所述至少一个处理器执行所述存储器存储的计算机执行指令,使得所述至少一个处理器执行如权利要求1至7任一项所述的应用程序启动方法。

10. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如权利要求1至7任一项所述的应用程序启动方法。

应用程序启动方法及设备

技术领域

[0001] 本发明实施例涉及语音识别技术领域,尤其涉及一种应用程序启动方法及设备。

背景技术

[0002] 多媒体播放中通常设置多层内容,并将该多层内容进行融合后呈现出内容丰富的界面,以满足用户对视频内容日渐提高的需求。每一层内容可能需要一个或多个单页网页(Web)应用提供。单页Web应用就是指只有一个Web页面作为入口的应用,在浏览器中运行期间不会重新加载页面。

[0003] 现有技术中,浏览器(例如Cobalt浏览器)的实现通常是浏览器的核心通过浏览器的对接层与底层具体硬件进行对接,Cobalt浏览器的对接层中的窗口接口层(Window Application Programming Interface,Window API)的实现仅能创建单一属性的窗口,也即单个浏览器仅能启动一个单页Web应用。若欲启动多层中的多个单页Web应用时需要启动多个独立程序来实现。

[0004] 然而,若每个单页Web应用单独占用一个独立程序的话,启动多个独立程序实现多个单页Web应用需要占用较多内存,增加了硬件成本。并且每个单页Web应用从代码角度来看,均需要单独维护,从而降低了工作效率,增加了人力成本。

发明内容

[0005] 本发明实施例提供一种应用程序启动方法及设备,以提高工作效率,降低成本。

[0006] 第一方面,本发明实施例提供一种应用程序启动方法,包括:

[0007] 浏览器接收用户输入的第一启动请求,并根据所述第一启动请求确定待启动的第一网页应用的地址;

[0008] 所述浏览器根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息;所述映射表中记录有待启动的各网页应用的地址和属性配置信息,以使所述浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用;

[0009] 所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用。

[0010] 在一种可能的设计中,所述待启动的各网页应用中包括用户界面应用;所述浏览器接收用户输入的第一启动请求之前,还包括:

[0011] 所述浏览器启动后在映射表中查询用户界面应用的属性配置信息,并根据所述用户界面应用的属性配置信息启动所述用户界面应用。

[0012] 在一种可能的设计中,所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用之后,还包括:

[0013] 所述浏览器根据所述第一网页应用的地址,对当前应用的记录信息进行更新。

[0014] 在一种可能的设计中,所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用之后,还包括:

- [0015] 所述浏览器的按键管理模块接收应用管理器发送的第一按键信息；
- [0016] 所述浏览器的按键管理模块将所述第一按键信息发送给所述用户界面应用以使所述用户界面应用对所述第一按键信息进行处理，得到第一处理结果并返回；
- [0017] 所述浏览器的按键管理模块接收所述第一处理结果，并对所述第一处理结果进行判断；
- [0018] 若所述第一处理结果为所述第一按键信息已被执行，则将所述第一处理结果返回给所述应用管理器。
- [0019] 在一种可能的设计中，所述浏览器的按键管理模块接收所述第一处理结果，并对所述第一处理结果进行判断之后，还包括：
- [0020] 若所述第一按键信息未被执行，则所述浏览器的按键管理模块将所述第一按键信息发送给当前应用，以使所述当前应用对所述第一按键信息进行处理，得到第二处理结果并返回；
- [0021] 所述浏览器的按键管理模块接收所述第二处理结果，并对所述第二处理结果进行判断；
- [0022] 若所述第二处理结果为所述第一按键信息已被执行，则将所述第二处理结果返回给所述应用管理器。
- [0023] 在一种可能的设计中，所述浏览器的按键管理模块接收所述第二处理结果，并对所述第二处理结果进行判断之后，还包括：
- [0024] 若所述第二处理结果为所述第一按键信息未被执行，则所述浏览器的按键管理模块根据预设应用列表，将所述第一按键信息依次发送给所述预设应用列表中各应用进行处理以使当前对所述第一按键信息进行处理的应用返回处理结果，直至接收到所述预设应用列表中最后一个应用返回的第三处理结果，或者当前对所述第一按键信息进行处理的应用返回所述第一按键信息已被执行的第四处理结果；其中，所述预设应用列表包括已启动的各网页应用的地址；将处理结果发送给应用管理器；
- [0025] 所述浏览器的按键管理模块将所述第三处理结果或所述第四处理结果返回给所述应用管理器。
- [0026] 在一种可能的设计中，所述浏览器接收用户输入的第一启动请求之前，还包括：
- [0027] 根据待启动的各网页应用的地址和属性配置信息，创建所述映射表，并存储于所述浏览器中。
- [0028] 第二方面，本发明实施例提供一种应用程序启动设备，包括：
- [0029] 接收模块，用于接收用户输入的第一启动请求，并根据所述第一启动请求确定待启动的第一网页应用的地址；
- [0030] 查询模块，用于根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息；所述映射表中记录有待启动的各网页应用的地址和属性配置信息，以使浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用；
- [0031] 第一启动模块，用于根据所述第一网页应用的属性配置信息启动所述第一网页应用。
- [0032] 第三方面，本发明实施例提供一种应用程序启动设备，包括：至少一个处理器和存

储器；

[0033] 所述存储器存储计算机执行指令；

[0034] 所述至少一个处理器执行所述存储器存储的计算机执行指令，使得所述至少一个处理器执行如上第一方面以及第一方面各种可能的设计所述的方法。

[0035] 第四方面，本发明实施例提供一种计算机可读存储介质，所述计算机可读存储介质中存储有计算机执行指令，当处理器执行所述计算机执行指令时，实现如上第一方面以及第一方面各种可能的设计所述的方法。

[0036] 本实施例提供的应用程序启动方法及设备，该方法通过浏览器接收用户输入的第一启动请求，并根据所述第一启动请求确定待启动的第一网页应用的地址；所述浏览器根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息；所述映射表中记录有待启动的各网页应用的地址和属性配置信息，以使所述浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用；所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用。本实施例提供的方法通过将待启动的各网页应用的地址和属性配置信息记录在映射表中，并在应用启动之前从所述映射表中获取相应的属性配置信息，能够在单个浏览器中启动多个单页Web应用，从而降低了所占用的内存空间，节约了硬件成本，并且减少了代码维护工作，提高工作效率的同时也降低了人工成本。

附图说明

[0037] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍，显而易见地，下面描述中的附图是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

[0038] 图1a和图1b为本发明一实施例提供的应用程序启动方法的应用场景图；

[0039] 图2为本发明一实施例提供的应用程序启动方法的流程示意图；

[0040] 图3为本发明又一实施例提供的Cobalt浏览器的架构图；

[0041] 图4为本发明又一实施例提供的应用程序启动方法的流程示意图；

[0042] 图5为本发明又一实施例提供的应用程序启动方法的流程示意图；

[0043] 图6为现有技术的按键处理系统架构示意图；

[0044] 图7为本发明又一实施例提供的按键处理系统架构示意图；

[0045] 图8为本发明又一实施例提供的应用程序启动设备的结构示意图；

[0046] 图9为本发明又一实施例提供的应用程序启动设备的结构示意图；

[0047] 图10为本发明又一实施例提供的应用程序启动设备的硬件结构示意图。

具体实施方式

[0048] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0049] 图1a和图1b为本发明一实施例提供的应用程序启动方法的应用场景图。如图1a所示,在一种多媒体播放的实现中,该多媒体播放内容包括视频播放内容和屏幕菜单(On Screen Display,OSD),其中,该视频播放包括相互独立的主层(main)和从层(sub),OSD包括第0层(Layer0)、第1层(Layer1)和第2层(Layer2)。在播放时,不同层之间具有优先级,高优先级的层会覆盖低优先级的层,将各层融合后得到最终的界面。例如,一种可能的实现是:视频播放层的优先级小于OSD层的优先级,在OSD层中,Layer 2<Layer 0<Layer 1。如图1b,在电视中的一种实现是电视用户界面(User Interface,UI)在最上层,视频网站应用YouTube和YouTube Kids,电子销售点(Electronic Point Of Sale,EPOS)在中间层。

[0050] 在具体实现过程中,上述电视中的4种应用(UI、YouTube、YouTube Kids和EPOS)都属于单页Web应用(SPA),也都基于Cobalt浏览器实现。但Cobalt浏览器的对接层中的“窗口接口层(Window API)”接口的相关实现不支持根据不同应用创建不同属性(比如不同层)的窗口,也就是说,当前Cobalt浏览器在每次启动后,创建的窗口都是同一种属性,仅能支持单个单页Web应用。该4种应用,均为独立的程序。每一个应用单独占用一个独立程序需要的硬件资源以及相关逻辑控制。因此启动多个单页Web应用需要启动多个独立程序,会使得占用内存较多,且使硬件成本增加。另外,从代码角度来看,上述4个应用,需要单独维护,降低了工作效率,增加了人力成本。基于此,本发明实施例提供一种应用程序启动方法,以提高工作效率,降低成本。

[0051] 下面以具体地实施例对本发明的技术方案进行详细说明。下面这几个具体的实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例不再赘述。

[0052] 图2为本发明一实施例提供的应用程序启动方法的流程示意图。如图2所示,该方法包括:

[0053] 201、浏览器接收用户输入的第一启动请求,并根据所述第一启动请求确定待启动的第一网页应用的地址。

[0054] 实际应用中,本实施例的执行主体为安装有浏览器的终端设备或服务器等电子设备。例如计算机、电视、平板、手机等。

[0055] 可选地,用户输入的第一启动请求可以通过与终端设备和服务器设备有线或无线连接的输入部件进行输入,例如通过遥控器、键盘或鼠标等部件。

[0056] 举例来说,用户在观看电视时,通过手机遥控器选中显示界面中的电视购物应用对应的图标,电视机接收到请求信号后,发送给浏览器进行处理。

[0057] 可选地,根据所述第一启动请求确定待启动的第一网页应用的地址的方式有多种,例如,一种实现方式中,第一启动请求可以直接包括待启动的第一网页应用的地址,浏览器对接收到的第一启动请求进行解析即可获得第一网页应用的地址。另一种实现方式中,第一启动请求可以包括待启动的第一网页应用的标识,对该第一启动请求解析获得该第一网页应用的标识,进而再根据该标识从预存的应用标识应用地址的对应表中查询获得第一网页应用的地址。表1为本发明又一实施例提供的应用标识应用地址的对应表;如表1所示,用户界面应用的标识为UI,该标识UI对应的应用地址为file:///html/hisenseUI/index.html;YouTube视频网站应用的应用标识为YouTube,该标识YouTube对应的应用地址为http://www.youtube.com/tv;YouTube Kids视频网站应用的应用标识为YouTube Kids,该标识YouTube Kids对应的应用地址为http://www.youtube.com/tv_kids;电子销售点应

用的应用标识为EPOS,该标识EPOS对应的应用地址为file://html/hisenseUI/webapp/EPOS/epos.html。假设第一网页应用时电子销售点应用,则对第一启动请求解析得到电子销售点应用的标识EPOS,根据EPOS标识从该对应表中获取到的应用地址为file://html/hisenseUI/webapp/EPOS/epos.html。

[0058] 表1应用标识应用地址的对应表

应用标识	应用地址
UI	file://html/hisenseUI/index.html
YouTube	http://www.youtube.com/tv
YouTube Kids	http://www.youtube.com/tv_kids
EPOS	file://html/hisenseUI/webapp/EPOS/epos.html

[0060] 202、所述浏览器根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息;所述映射表中记录有待启动的各网页应用的地址和属性配置信息,以使所述浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用。

[0061] 可选地,所述属性配置信息可以包括应用的窗口的属性信息以及所对应底层的属性信息,例如可以包括应用的窗口的所属层级、分辨率、起始位置,以及底层是否存在数据缓存等内容。

[0062] 具体的,为了使浏览器(例如:Cobalt浏览器)能够支持多个应用,即该多个应用共享一个程序/进程,以减少内存节约成本,本实施例中将多个应用的应用地址和多个属性配置信息存储在一个映射表中,该映射表包括每个应用与其相对应的配置信息之间映射关系。具体的,在运用浏览器支持多个应用之前可以根据待启动的各网页应用的地址和属性配置信息,创建所述映射表,并存储于所述浏览器中。表2为本发明又一实施例提供的映射表;如表2所示,应用地址file://html/hisenseUI/index.html对应的属性配置信息为{Layer1,4k {3840*2140} position {0,0},no buffer},应用地址http://www.youtube.com/tv、应用地址http://www.youtube.com/tv_kids和应用地址file://html/hisenseUI/webapp/EPOS/epos.html分别对应的属性配置信息均为{Layer0,2k {1920*1080} position {0,0}},其中,Layer0,Layer1代表应用的窗口所属层级;2k {1920*1080},4k {3840*2140}代表应用的窗口的分辨率;Position {0,0}代表应用窗口的起始坐标;No buffer代表底层没有窗口数据缓存。

[0063] 表2映射表

应用地址	属性配置信息
file://html/hisenseUI/index.html	{Layer1,4k{3840*2140}position{0,0}, no buffer}
http://www.youtube.com/tv	{Layer0,2k{1920*1080}position{0,0}}
http://www.youtube.com/tv_kids	{Layer0,2k{1920*1080}position{0,0}}
file://html/hisenseUI/webapp/EPOS/e pos.html	{Layer0,2k{1920*1080}position{0,0}}

[0065] 实际应用中,在接收到启动请求后,为待启动的应用创建窗口时,根据由启动请求确定的应用地址从映射表中查询出对应的属性配置信息,根据该属性配置信息创建相应的应用的窗口。图3为本发明又一实施例提供的Cobalt浏览器的架构图;如图3所示,该架构图包括开源代码部分和用户实现部分,开源代码部分包括Cobalt浏览器核心;用户实现部分包括Cobalt对接层和底层平台,其中,Cobalt对接层包括窗口接口层和其他应用接口,以及包括待启动的各应用的应用地址与属性配置信息的映射表。通过从该映射表中获取当前待启动的应用的属性配置信息,来为待启动的应用创建窗口,实现应用的启动。

[0066] 203、所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用。

[0067] 为了便于后续操作中的处理,需要对各应用的状态进行记录,该记录信息可以包括已启动应用的信息和当前应用的信息。具体的,在完成第一网页应用的启动后,第一网页应用即为当前应用,因此,所述浏览器可以根据所述第一网页应用的地址,对当前应用的记录信息进行更新,将新启动的第一网页应用作为当前应用。并且,还可以将该第一网页应用记录在已启动应用的记录信息中。

[0068] 本实施例提供的应用程序启动方法,通过浏览器接收用户输入的第一启动请求,并根据所述第一启动请求确定待启动的第一网页应用的地址;所述浏览器根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息;所述映射表中记录有待启动的各网页应用的地址和属性配置信息,以使所述浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用;所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用。本实施例提供的方法通过将待启动的各网页应用的地址和属性配置信息记录在映射表中,并在应用启动之前从所述映射表中获取相应的属性配置信息,能够在单个浏览器中启动多个单页Web应用,从而降低了所占用的内存空间,节约了硬件成本,并且减少了代码维护工作,提高工作效率的同时也降低了人工成本。

[0069] 可选地,在另一实施例中,在步骤203之后还可以包括:

[0070] 204、浏览器接收用户输入的第二启动请求,并根据所述第二启动请求确定待启动的第二网页应用的地址。

[0071] 205、所述浏览器根据所述第二网页应用的地址在映射表中查询所述第二网页应用的属性配置信息。

- [0072] 206、所述浏览器根据所述第二网页应用的属性配置信息启动所述第二网页应用。
- [0073] 本实施例中,步骤204与步骤206与上述步骤201至步骤203相类似此处不再赘述。
- [0074] 本实施例通过从映射表中获取第二启动请求对应的第二网页应用的属性配置信息,进而根据该配置信息在同一浏览器程序中启动第二网页应用,节省了内存,降低了硬件成本,且相对于将第二网页应用以独立程序启动,减少了代码维护的人工成本,提高了工作效率。
- [0075] 图4为本发明又一实施例提供的应用程序启动方法的流程示意图。在上述实施例的基础上,本实施例中对单个浏览器启动多个应用的过程进行了详细描述,如图4所示,该方法包括:
- [0076] 401、所述浏览器启动后在映射表中查询用户界面应用的属性配置信息,并根据所述用户界面应用的属性配置信息启动所述用户界面应用。
- [0077] 本实施例中,用户界面UI应用可以设置为处理硬件设备的基本操作和/或其他应用的启动,例如电视中的音量调节,频道切换等,因此在浏览器启动后,优先启动该UI应用,以支持其他应用的启动。
- [0078] 402、浏览器接收用户输入的第一启动请求,并根据所述第一启动请求确定待启动的第一网页应用的地址。
- [0079] 403、所述浏览器根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息;所述映射表中记录有待启动的各网页应用的地址和属性配置信息,以使所述浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用。
- [0080] 404、所述浏览器根据所述第一网页应用的属性配置信息启动所述第一网页应用。
- [0081] 本实施例中步骤402至步骤404与本实施例中步骤201至步骤203的步骤相类似,此处不再赘述。
- [0082] 以下以在电视中的应用为例,对上述步骤进行示例说明,
- [0083] 电视启动后,浏览器默认启动运行;
- [0084] 浏览器启动后,默认总是在新网页1中打开UI对应的网页应用,具体的,浏览器从映射表中获取UI应用地址对应的属性配置信息并显示。此时,浏览器中只有1个应用UI(用户界面)在运行,且当前应用也是UI;
- [0085] 设备收到按键信息,例如,用户通过遥控器选择电视频道;按键响应在UI应用中进行处理并执行完毕;此时,因为未启动新的应用,所以在浏览器中仍然只有UI这1个应用在运行,且当前应用仍然是UI。
- [0086] 当设备收到另一个按键信息,比如,用户通过遥控器设置电视成“商场模式”。浏览器会在新页面2中打开EPOS(电子销售终端)应用,具体的,浏览器根据按键信息解析出EPOS应用地址,并从映射表中获取EPOS应用地址对应的属性配置信息并显示。此时,浏览器中有2个应用UI&EPOS在运行,且当前应用是EPOS;
- [0087] 当设备收到另一个按键信息,比如,用户通过遥控器打开“YouTube”。浏览器会在新页面3中打开YouTube应用,具体的,浏览器根据按键信息解析出YouTube应用地址,并从映射表中获取YouTube应用地址对应的属性配置信息并显示。此时,浏览器中有3个应用UI&EPOS&YouTube在运行,且当前应用是YouTube;

[0088] 当设备收到另一个按键,比如,用户通过遥控器设置电视音量。浏览器会显示UI应用中的音量设置条。此时,浏览器中有3个应用UI&EPOS&YouTube在运行,且当前应用是UI;

[0089] 当音量设置完毕后,UI界面隐藏到后端。此时,浏览器中有3个应用UI&EPOS&YouTube在运行,且当前应用是YouTube;

[0090] 当有应用退出后,比如YouTube,浏览器中有2个应用UI&EPOS在运行,且当前应用是EPOS;

[0091] 当再有应用退出后,比如EPOS,浏览器中有1个应用UI在运行,且当前应用是UI;

[0092] 从上述步骤中可知,已启动应用和当前应用的记录信息需要实时更新,以便后续操作中调用该信息。并且,因为要支持多个应用,就将浏览器收到消息后转给浏览器已启动的各单页应用。其中,可以将UI的处理优先级设置为最高,优先对按键信息进行处理。

[0093] 本实施例提供的应用程序启动方法,通过在浏览器启动后,根据映射表启动用户界面应用,为后续实现用户的基本设置操作以及启动其他应用的操作做好了准备。便于用户操作,降低学习成本。

[0094] 图5为本发明又一实施例提供的应用程序启动方法的流程示意图,在上述实施例的基础上,例如在图4所示的实施例的基础上,为了适应于单个浏览器支持多个单页应用的情况,本实施例对按键信息的处理进行了改进,在浏览器中增设了按键管理模块。如图5所示,该方法包括:

[0095] 501、所述浏览器的按键管理模块接收应用管理器发送的第一按键信息。

[0096] 502、所述浏览器的按键管理模块将所述第一按键信息发送给所述用户界面应用以使所述用户界面应用对所述第一按键信息进行处理,得到第一处理结果并返回。

[0097] 503、所述浏览器的按键管理模块接收所述第一处理结果,并对所述第一处理结果进行判断。

[0098] 504、若所述第一处理结果为所述第一按键信息已被执行,则将所述第一处理结果返回给所述应用管理器。

[0099] 505、若所述第一按键信息未被执行,则所述浏览器的按键管理模块将所述第一按键信息发送给当前应用,以使所述当前应用对所述第一按键信息进行处理,得到第二处理结果并返回。

[0100] 506、所述浏览器的按键管理模块接收所述第二处理结果,并对所述第二处理结果进行判断。

[0101] 507、若所述第二处理结果为所述第一按键信息已被执行,则将所述第二处理结果返回给所述应用管理器。

[0102] 508、若所述第二处理结果为所述第一按键信息未被执行,则所述浏览器的按键管理模块根据预设应用列表,将所述第一按键信息依次发送给所述预设应用列表中各应用进行处理以使当前对所述第一按键信息进行处理的应用返回处理结果,直至接收到所述预设应用列表中最后一个应用返回的第三处理结果,或者当前对所述第一按键信息进行处理的应用返回所述第一按键信息已被执行的第四处理结果;其中,所述预设应用列表包括已启动的各网页应用的地址;将处理结果发送给应用管理器。

[0103] 509、所述浏览器的按键管理模块将所述第三处理结果或所述第四处理结果返回给所述应用管理器。

[0104] 在本实施例中,所述预设应用列表可以为动态注册创建的应用列表。具体的,当一个应用启动并被应用管理器注册后即加入该预设应用列表。所述预设应用列表中各应用的排序可以是根据注册的先后顺序确定,还可以根据各应用的使用频繁程度进行排序确定的。本实施例对此不做限定。

[0105] 参照图6和图7,图6为现有技术的按键处理系统架构示意图,图7为本发明又一实施例提供的按键处理系统架构示意图。

[0106] 如图6所示,现有的按键处理系统包括浏览器603、应用管理器602、硬件平台601和应用604。其中,应用管理器602为与浏览器603同样级别的应用。在具体的按键处理过程中,应用管理器602监听底层的硬件平台601收到的按键信息,应用管理器602将接收到的按键信息首先发送给浏览器603,而浏览器603中当前只有用户界面应用UI,UI如果无法处理该按键信息,那么应用管理器602再将该按键信息发送给应用603中的各应用,例如YouTube、EPOS、YouTube Kids或其他应用Others。在此过程中浏览器602仅支持一个应用(用户界面应用UI),应用604内也均为单网页应用,需要启动多个独立程序,会使得占用内存较多,且使硬件成本增加。基于此,为了解决上述问题,如图7所示,本实施例提供的按键处理系统增设了按键管理模块7031,具体的,该按键处理系统包括硬件平台701、应用管理器702、浏览器703和应用704,其中,浏览器703包括按键管理器7031。在具体的按键管理过程中,应用管理器702监听底层的硬件平台701收到的按键信息,应用管理器702将接收到的按键信息首先发送给浏览器703中的按键管理模块7031,按键管理模块7031将该按键信息首先发送给用户界面应用UI,按键管理模块7031判断该按键信息是否被处理,若未处理则依次发送给浏览器703内的其他应用(YouTube、YouTube Kids和EPOS),直至遍历完所有浏览器Browser中的所有应用或者判定按键被某一应用处理。若遍历完浏览器703中的所有应用,按键信息均未被处理则将按键信息再转发给应用704中的其他应用Others。由此,通过在浏览器703中增设了对按键处理结果的处理,而不是直接返回给应用管理器702。浏览器703能够支持多个应用,实现了浏览器703中多个应用对按键信息的处理。有利于实现单个浏览器703对多个应用的支持,从而达到提高工作效率降低成本的目的。

[0107] 具体的,以下以在电视中的应用为例,对上述本实施例提供的按键处理过程进行示例说明:

[0108] 电视机通过底层硬件设备收到遥控器的按键信息后,监听按键事件的应用管理器(APplication Manager,APM)会收到该按键信息;

[0109] APM首先将按键信息发送给给UI所在的浏览器处理;

[0110] 浏览器首先将按键信息发送给UI应用;

[0111] UI应用对该按键信息进行处理,获得处理结果,并返回该处理结果给浏览器的按键管理模块;

[0112] 浏览器的按键管理模块对返回的处理结果进行判断;

[0113] 当该按键信息已被执行,则返回处理结果给APM;

[0114] 当该按键信息未被执行,则判断是否已经将按键信息转发给当前应用;

[0115] 若没有发给当前应用,则将按键信息转发给浏览器中的当前应用,比如YouTube,当前应用对该按键信息进行处理,返回处理结果给浏览器的按键管理模块;当该按键信息已被执行,则返回处理结果给APM;

[0116] 若之前已经转发给当前应用,且该按键信息未被执行,则将该按键按照预设的应用列表的顺序依次转发给浏览器中的其他应用,比如EPOS;其它应用对该按键信息进行处理,返回处理结果给按键管理模块,按键管理模块对该按键信息是否已被执行进行判断,若已被执行则将处理结果返回APM;若遍历至最后一个应用对该按键信息执行,则直接将接收到的该最后一个应用的处理结果返回APM;

[0117] APM对接收到的处理结果进行判断;

[0118] 若按键信息已被执行,则按键处理结束;

[0119] 若按键信息未被执行,换言之,当前按键信息不属于任何浏览器应用可执行的范畴:APM转发该按键信息给浏览器以外的当前应用,当前应用对该按键信息进行处理,若按键信息已被执行返回处理结果给APM;

[0120] 若按键信息未被执行则按照应用列表顺序转发按键信息给浏览器以外的其它应用,其它应用处理该按键,返回处理结果给APM。

[0121] 本实施例提供的应用程序启动方法,通过在浏览器中设置按键管理模块,实现了浏览器中多个应用对按键信息的处理。有利于实现单个浏览器对多个应用的支持,从而达到提高工作效率降低成本的目的。

[0122] 图8为本发明又一实施例提供的应用程序启动设备的结构示意图。如图8所示,该应用程序启动设备80包括:接收模块801、查询模块802和第一启动模块803。

[0123] 接收模块801,用于接收用户输入的第一启动请求,并根据所述第一启动请求确定待启动的第一网页应用的地址;

[0124] 查询模块802,用于根据所述第一网页应用的地址在映射表中查询所述第一网页应用的属性配置信息;所述映射表中记录有待启动的各网页应用的地址和属性配置信息,以使所述浏览器根据所述映射表中的所述待启动的各网页应用的属性配置信息分别启动所述待启动的各网页应用;

[0125] 第一启动模块803,用于根据所述第一网页应用的属性配置信息启动所述第一网页应用。

[0126] 本发明实施例提供的应用程序启动设备,通过将待启动的各网页应用的地址和属性配置信息记录在映射表中,并在应用启动之前从所述映射表中获取相应的属性配置信息,能够在单个浏览器中启动多个单页Web应用,从而降低了所占用的内存空间,节约了硬件成本,并且减少了代码维护工作,提高工作效率的同时也降低了人工成本。

[0127] 图9为本发明又一实施例提供的应用程序启动设备的结构示意图。如图9所示,该应用程序启动设备80还包括:第二启动模块804、更新模块805、按键管理模块806和创建模块807。

[0128] 所述待启动的各网页应用中包括用户界面应用;所述设备还包括:

[0129] 第二启动模块804,用于所述浏览器启动后在映射表中查询用户界面应用的属性配置信息,并根据所述用户界面应用的属性配置信息启动所述用户界面应用。

[0130] 可选地,所述设备还包括:

[0131] 更新模块805,用于根据所述第一网页应用的地址,对当前应用的记录信息进行更新。

[0132] 可选地,所述设备还包括:按键管理模块806用于

- [0133] 按键管理模块接收应用管理器发送的第一按键信息；
- [0134] 将所述第一按键信息发送给所述用户界面应用以使所述用户界面应用对所述第一按键信息进行处理，得到第一处理结果并返回；
- [0135] 接收所述第一处理结果，并对所述第一处理结果进行判断；
- [0136] 若所述第一处理结果为所述第一按键信息已被执行，则将所述第一处理结果返回给所述应用管理器。
- [0137] 可选地，所述按键管理模块806还用于：
- [0138] 若所述第一按键信息未被执行，则将所述第一按键信息发送给当前应用，以使所述当前应用对所述第一按键信息进行处理，得到第二处理结果并返回；
- [0139] 接收所述第二处理结果，并对所述第二处理结果进行判断；
- [0140] 若所述第二处理结果为所述第一按键信息已被执行，则将所述第二处理结果返回给所述应用管理器。
- [0141] 可选地，所述按键管理模块806还用于：
- [0142] 若所述第二处理结果为所述第一按键信息未被执行，则根据预设应用列表，将所述第一按键信息依次发送给所述预设应用列表中各应用进行处理以使当前对所述第一按键信息进行处理的应用返回处理结果，直至接收到所述预设应用列表中最后一个应用返回的第三处理结果，或者当前对所述第一按键信息进行处理的应用返回所述第一按键信息已被执行的第四处理结果；其中，所述预设应用列表包括已启动的各网页应用的地址；将处理结果发送给应用管理器；
- [0143] 将所述第三处理结果或所述第四处理结果返回给所述应用管理器。
- [0144] 可选地，所述设备还包括：
- [0145] 创建模块807，用于根据待启动的各网页应用的地址和属性配置信息，创建所述映射表，并存储于所述浏览器中。
- [0146] 本发明实施例提供的应用程序启动设备，可用于执行上述的方法实施例，其实现原理和技术效果类似，本实施例此处不再赘述。
- [0147] 图10为本发明又一实施例提供的应用程序启动设备的硬件结构示意图。如图10所示，本实施例提供的应用程序启动设备100包括：至少一个处理器1001和存储器1002。该应用程序启动设备100还包括通信部件1003。其中，处理器1001、存储器1002以及通信部件1003通过总线1004连接。
- [0148] 在具体实现过程中，至少一个处理器1001执行所述存储器1002存储的计算机执行指令，使得至少一个处理器1001执行如上应用程序启动设备100所执行的应用程序启动方法。
- [0149] 当本实施例的方法由服务器执行时，该通信部件1003可以将启动请求和按键信息发送给服务器。
- [0150] 处理器1001的具体实现过程可参见上述方法实施例，其实现原理和技术效果类似，本实施例此处不再赘述。
- [0151] 在上述的图10所示的实施例中，应理解，处理器可以是中央处理单元（英文：Central Processing Unit，简称：CPU），还可以是其他通用处理器、数字信号处理器（英文：Digital Signal Processor，简称：DSP）、专用集成电路（英文：Application Specific

Integrated Circuit,简称:ASIC)等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合发明所公开的方法的步骤可以直接体现为硬件处理器执行完成,或者用处理器中的硬件及软件模块组合执行完成。

[0152] 存储器可能包含高速RAM存储器,也可能还包括非易失性存储NVM,例如至少一个磁盘存储器。

[0153] 总线可以是工业标准体系结构(Industry Standard Architecture,ISA)总线、外部设备互连(Peripheral Component,PCI)总线或扩展工业标准体系结构(Extended Industry Standard Architecture,EISA)总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示,本申请附图中的总线并不限定仅有一根总线或一种类型的总线。

[0154] 本申请还提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如上应用程序启动设备执行的应用程序启动方法。

[0155] 上述的计算机可读存储介质,上述可读存储介质可以是由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。可读存储介质可以是通用或专用计算机能够存取的任何可用介质。

[0156] 一种示例性的可读存储介质耦合至处理器,从而使处理器能够从该可读存储介质读取信息,且可向该可读存储介质写入信息。当然,可读存储介质也可以是处理器的组成部分。处理器和可读存储介质可以位于专用集成电路(Application Specific Integrated Circuits,简称:ASIC)中。当然,处理器和可读存储介质也可以作为分立组件存在于设备中。

[0157] 本领域普通技术人员可以理解:实现上述各方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成。前述的程序可以存储于一计算机可读取存储介质中。该程序在执行时,执行包括上述各方法实施例的步骤;而前述的存储介质包括:ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0158] 最后应说明的是:以上各实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述各实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分或者全部技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的范围。

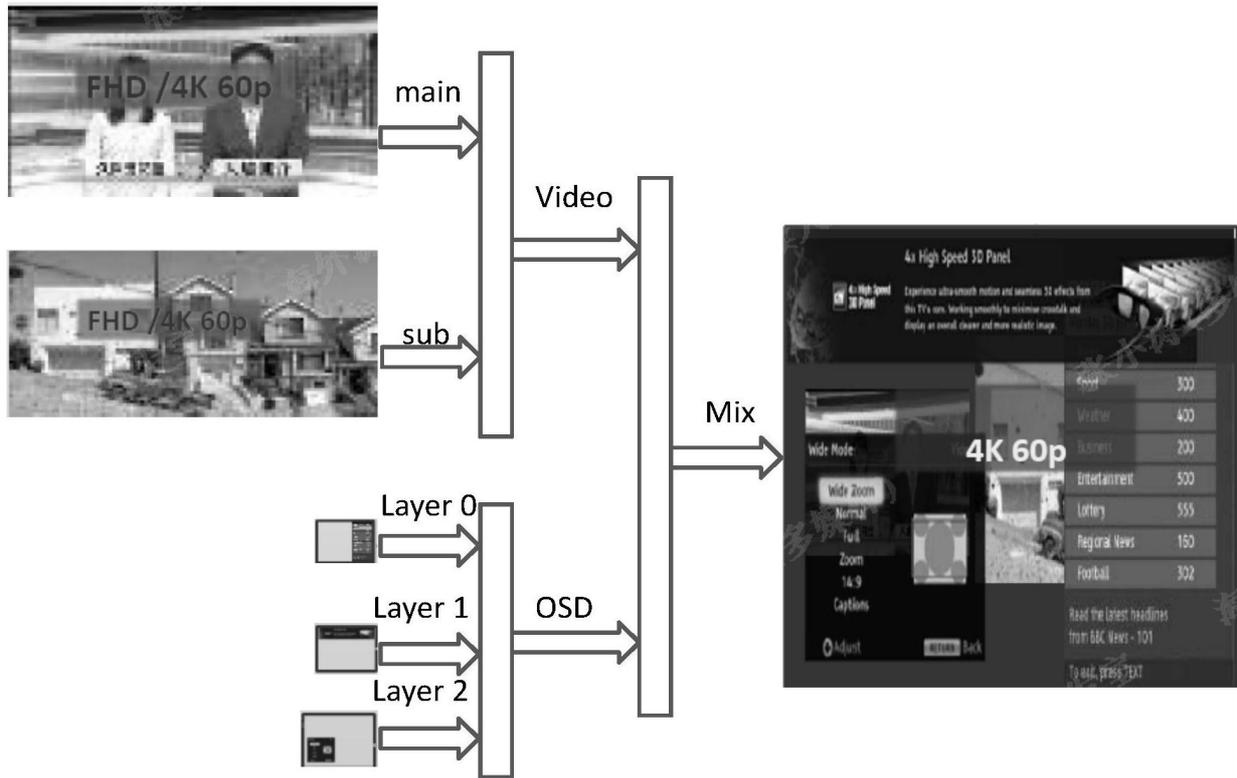


图1a

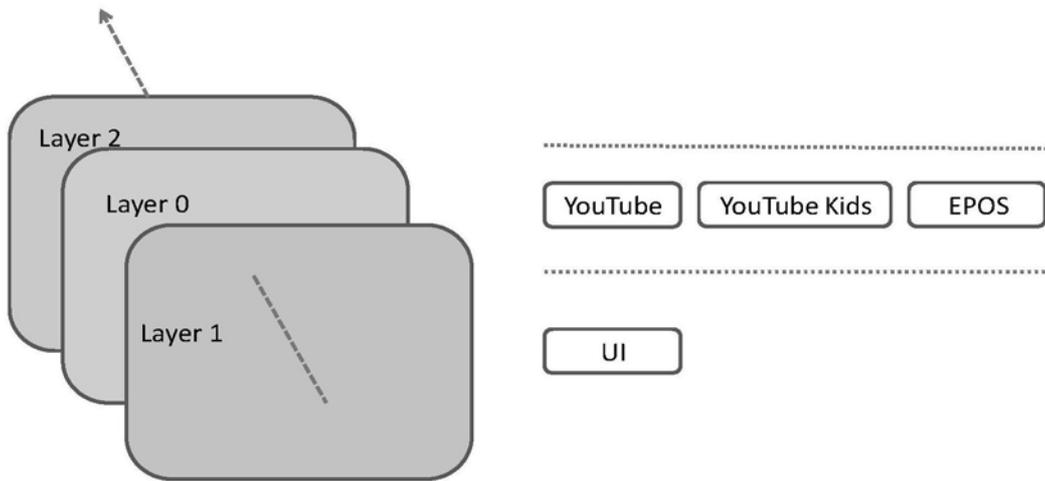


图1b

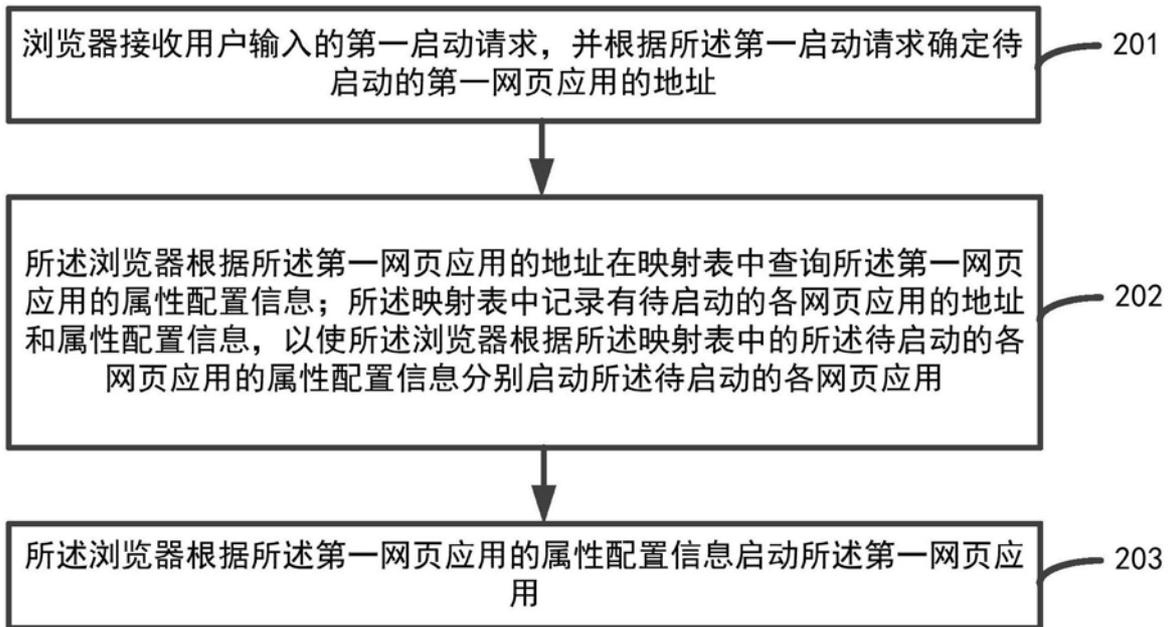


图2

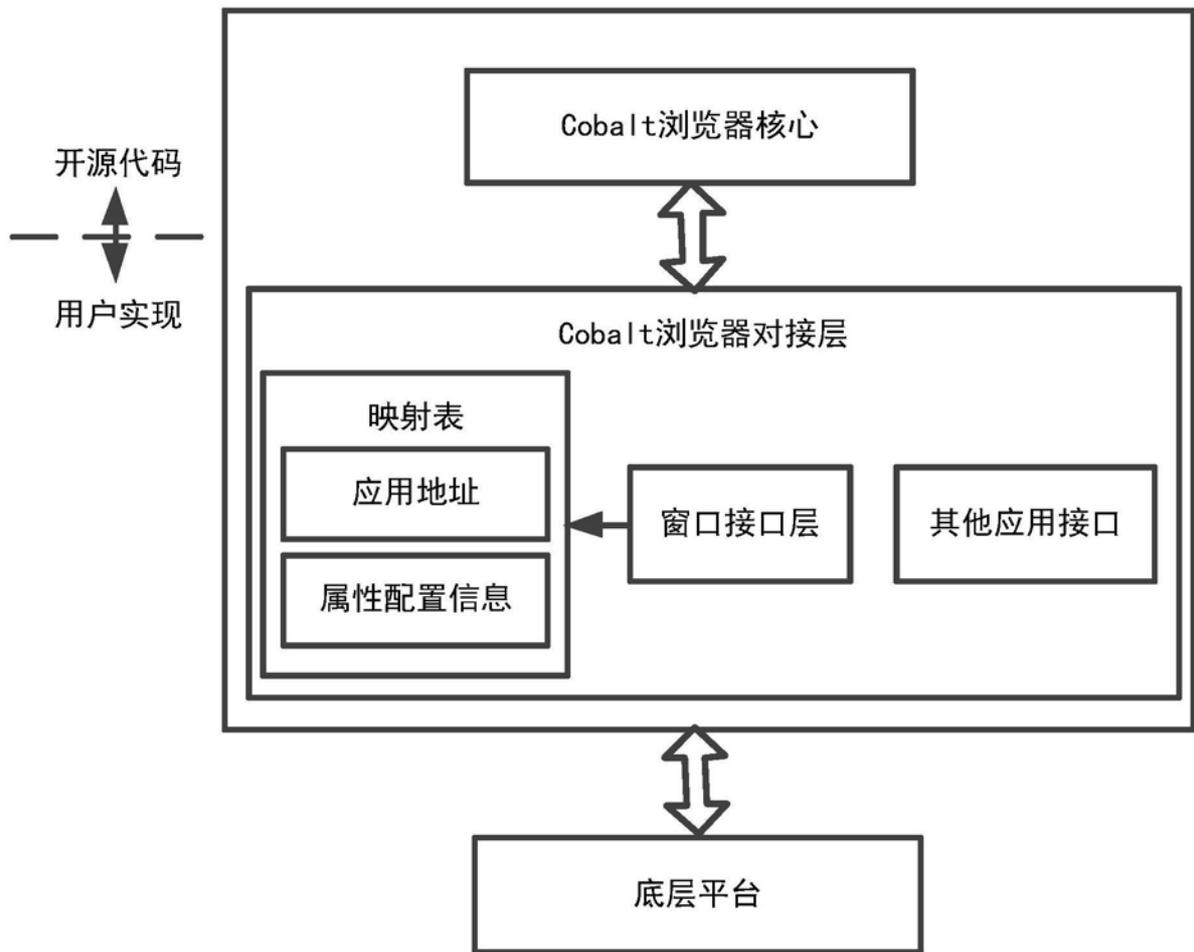


图3

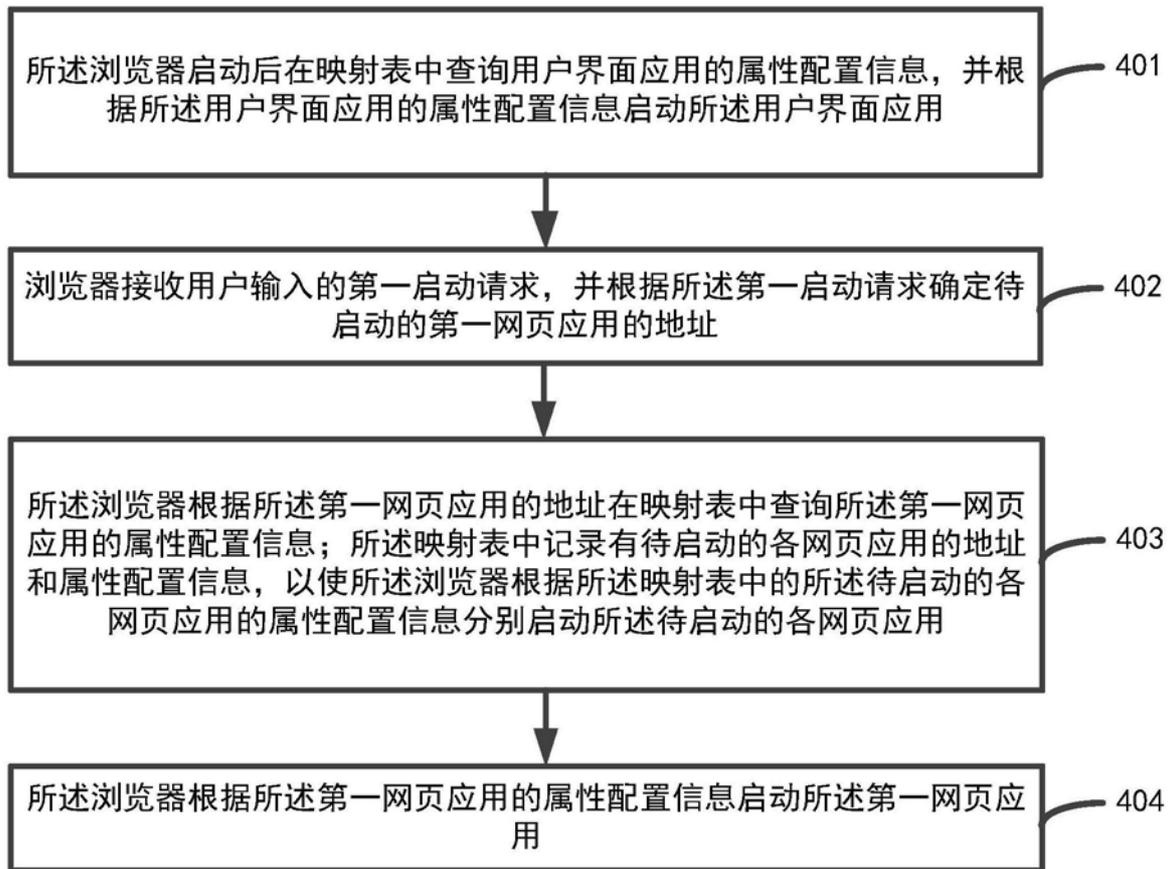


图4

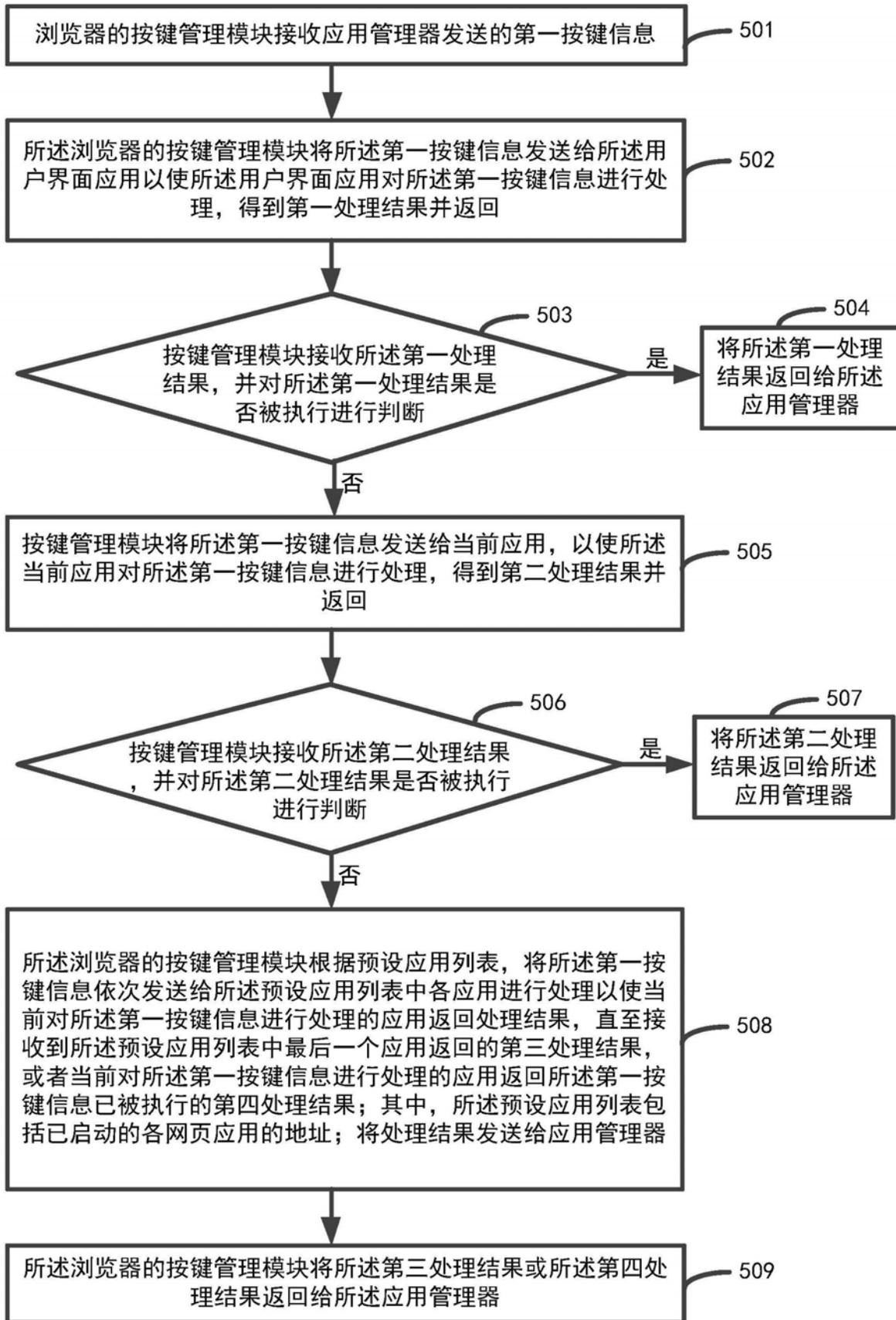


图5

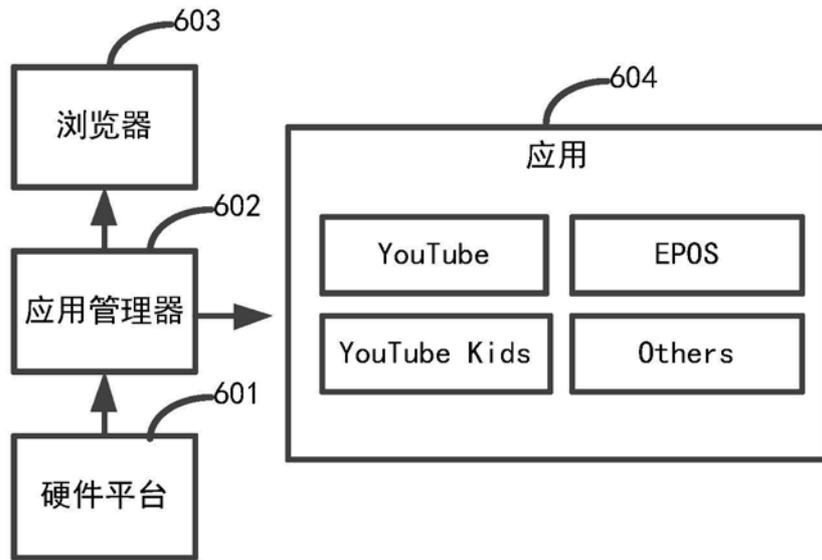


图6

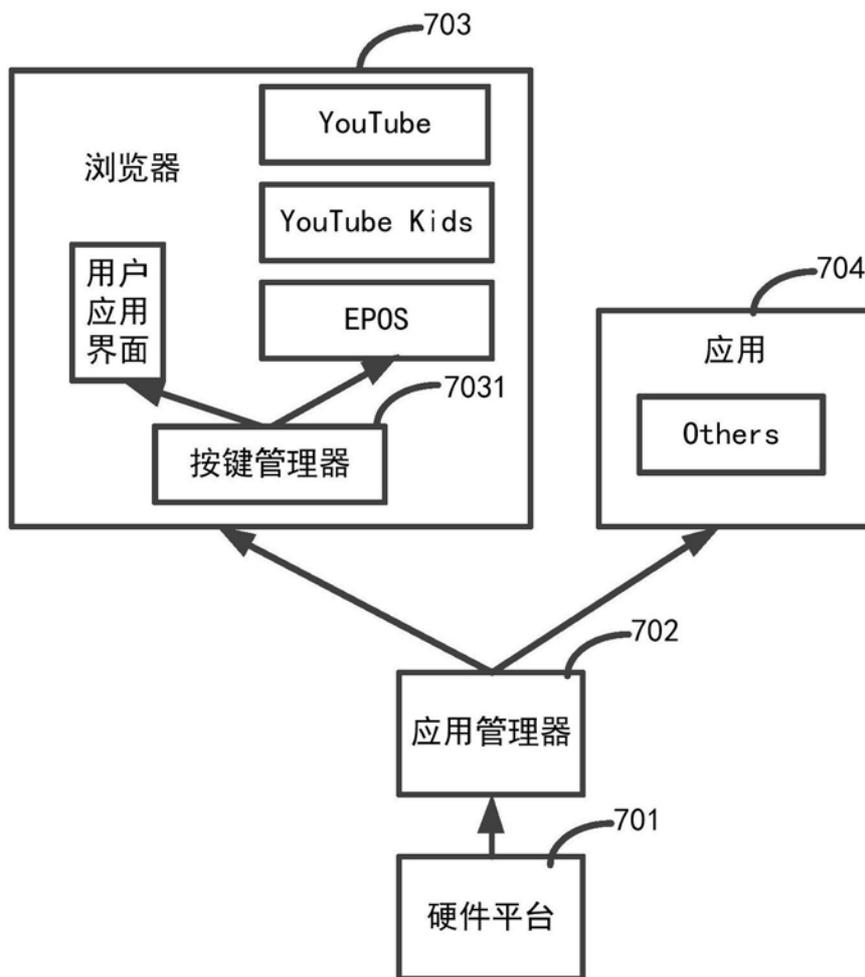


图7

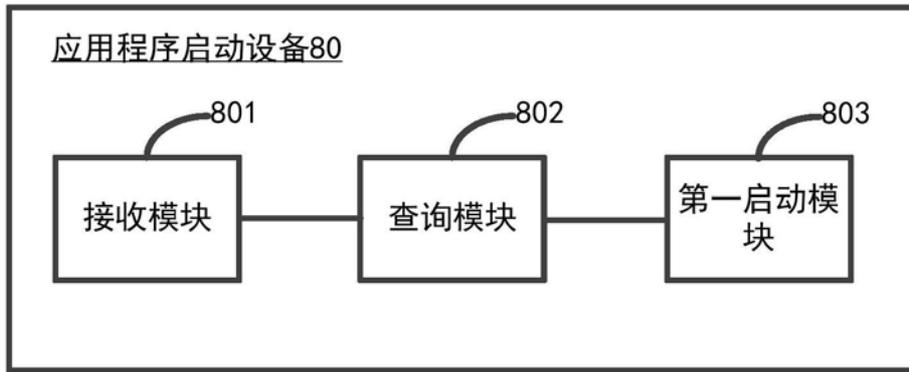


图8

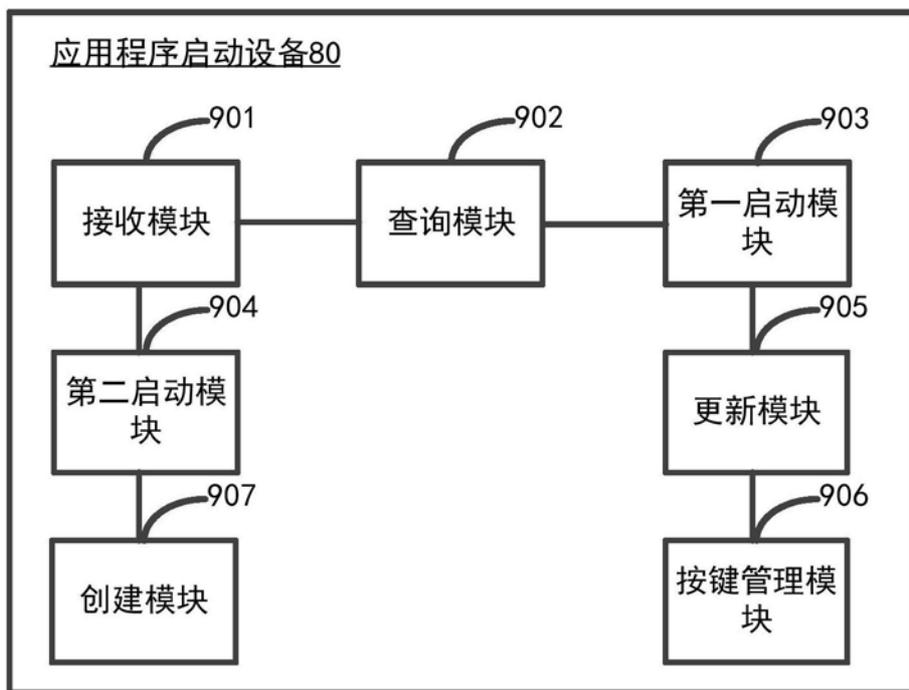


图9

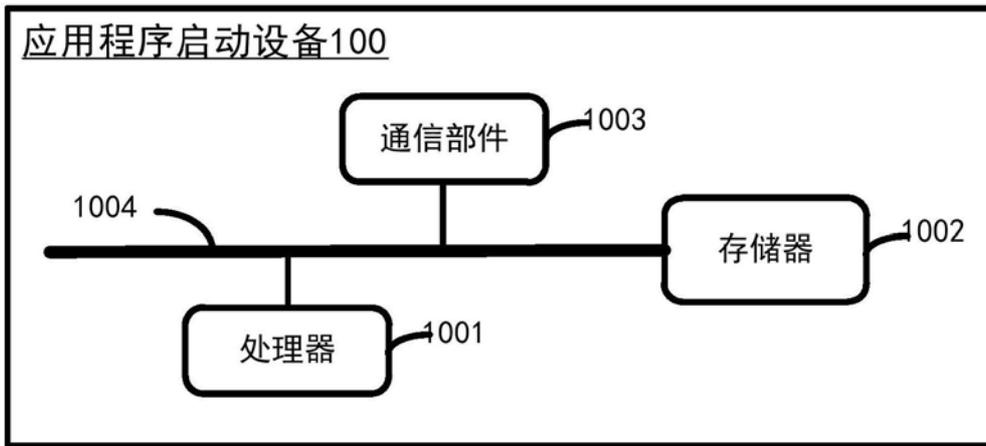


图10