



(12) 发明专利申请

(10) 申请公布号 CN 112925583 A

(43) 申请公布日 2021.06.08

(21) 申请号 202110212463.4

(22) 申请日 2021.02.25

(71) 申请人 北京百度网讯科技有限公司  
地址 100085 北京市海淀区上地十街10号  
百度大厦2层

(72) 发明人 蔡晓雯 孙微 李佳 朱叶青

(74) 专利代理机构 北京英赛嘉华知识产权代理  
有限责任公司 11204  
代理人 王达佐 马晓亚

(51) Int. Cl.  
G06F 9/445 (2018.01)  
G06F 8/71 (2018.01)

权利要求书2页 说明书9页 附图6页

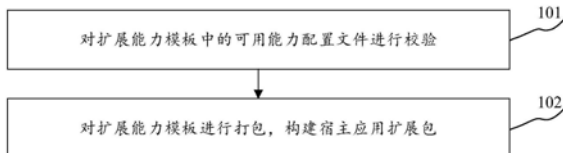
(54) 发明名称

宿主应用能力扩展方法、装置、设备以及存储介质

(57) 摘要

本申请公开了一种宿主应用能力扩展方法、装置、设备以及存储介质,涉及应用平台技术领域。该方法的一具体实施方式包括:对扩展能力模板中的可用能力配置文件进行校验,其中,扩展能力模板是利用方法集配置化生成的,方法集循环利用不同扩展能力模板中的相同源文件,可用能力配置文件包含在宿主应用上可用的能力的信息;对扩展能力模板进行打包,构建宿主应用扩展包。该实施方式为宿主应用开发者提供一种通过自动化配置实现小程序个性化能力扩展的方法集,该方法集可实现配置化生成扩展能力模板、打包构建宿主应用扩展包等功能,使得宿主应用可以无障碍、低成本接入。

100



1. 一种宿主应用能力扩展方法,包括:

对扩展能力模板中的可用能力配置文件进行校验,其中,所述扩展能力模板是利用方法集配置化生成的,所述方法集循环利用不同扩展能力模板中的相同源文件,所述可用能力配置文件包含在宿主应用上可用的能力的信息;

对所述扩展能力模板进行打包,构建宿主应用扩展包。

2. 根据权利要求1所述的方法,其中,所述方法还包括:

将所述宿主应用扩展包发送到服务端进行上线发布。

3. 根据权利要求1所述的方法,其中,在所述利用方法集配置化生成扩展能力模板之后,还包括:

利用所述方法集自动化生成描述表,其中,所述描述表是脚本语言与客户端之间的通信接口表示;

对所述描述表进行校验。

4. 根据权利要求3所述的方法,其中,在所述利用所述方法集自动化生成描述表之后,还包括:

根据所述描述表生成能力文档。

5. 根据权利要求3所述的方法,其中,在所述利用所述方法集自动化生成描述表之后,还包括:

根据所述描述表生成单元测试文件。

6. 根据权利要求1所述的方法,其中,利用方法集配置化生成所述扩展能力模板的步骤包括:

根据交互式命令行的提示,选择扩展能力项目对应的选项,创建所述扩展能力项目;或者

使用交互式命令行自动化生成预设类型的扩展能力项目,并存放在对应的项目路径下,其中,所述方法集支持交互式命令行进行配置。

7. 根据权利要求3所述的方法,其中,所述利用所述方法集自动化生成描述表,包括:

根据交互式命令行的提示,输入命令行字段,其中,所述命令行字段包括以下至少一项:应用程序编程接口API的名称、调起客户端能力的方式和客户端方法的存放路径;

利用所述方法集对所述命令行字段进行组装拼接,生成所述描述表。

8. 根据权利要求5所述的方法,其中,所述根据所述描述表生成单元测试文件,包括:

利用所述方法集,由所述描述表自动生成所述单元测试文件,其中,所述单元测试文件包括正常调用能力的事件和异常调用能力的事件,所述异常调用能力的事件包括以下至少一项:缺少参数、参数类型错误和参数不合要求。

9. 根据权利要求2所述的方法,其中,在所述将所述宿主应用扩展包发送到服务端进行上线发布之后,还包括:

在本地控制台输出所述宿主应用扩展包的二维码和替换工具的二维码。

10. 一种宿主应用能力扩展装置,包括:

第一校验模块,被配置成对扩展能力模板中的可用能力配置文件进行校验,其中,所述扩展能力模板是利用方法集配置化生成的,所述方法集循环利用不同扩展能力模板中的相同源文件,所述可用能力配置文件包含在宿主应用上可用的能力的信息;

构建模块,被配置成对所述扩展能力模板进行打包,构建宿主应用扩展包。

11. 根据权利要求10所述的装置,其中,所述装置还包括:

发布模块,被配置成将所述宿主应用扩展包发送到服务端进行上线发布。

12. 根据权利要求10所述的装置,其中,所述装置还包括:

第二生成模块,被配置成利用所述方法集自动化生成描述表,其中,所述描述表是脚本语言与客户端之间的通信接口表示;

第二校验模块,被配置成对所述描述表进行校验。

13. 根据权利要求12所述的装置,其中,所述装置还包括:

第三生成模块,被配置成根据所述描述表生成能力文档。

14. 根据权利要求12所述的装置,其中,所述装置还包括:

第四生成模块,被配置成根据所述描述表生成单元测试文件。

15. 根据权利要求10所述的装置,其中,所述装置还包括第一生成模块,所述第一生成模块被配置成:

根据交互式命令行的提示,选择扩展能力项目对应的选项,创建所述扩展能力项目;或者

使用交互式命令行自动化生成预设类型的扩展能力项目,并存放在对应的项目路径下,其中,所述方法集支持交互式命令行进行配置。

16. 根据权利要求12所述的装置,其中,所述第二生成模块进一步被配置成:

根据交互式命令行的提示,输入命令行字段,其中,所述命令行字段包括以下至少一项:应用程序编程接口API的名称、调起客户端能力的方式和客户端方法的存放路径;

利用所述方法集对所述命令行字段进行组装拼接,生成所述描述表。

17. 根据权利要求14所述的装置,其中,所述第四生成模块进一步被配置成:

利用所述方法集,由所述描述表自动生成所述单元测试文件,其中,所述单元测试文件包括正常调用能力的事件和异常调用能力的事件,所述异常调用能力的事件包括以下至少一项:缺少参数、参数类型错误和参数不合要求。

18. 根据权利要求11所述的装置,其中,所述装置还包括:

输出模块,被配置成在本地控制台输出所述宿主应用扩展包的二维码和替换工具的二维码。

19. 一种电子设备,包括:

至少一个处理器;以及

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行权利要求1-9中任一项所述的方法。

20. 一种存储有计算机指令的非瞬时计算机可读存储介质,所述计算机指令用于使所述计算机执行权利要求1-9中任一项所述的方法。

21. 一种计算机程序产品,包括计算机程序,所述计算机程序在被处理器执行时实现根据权利要求1-9中任一项所述的方法。

## 宿主应用能力扩展方法、装置、设备以及存储介质

### 技术领域

[0001] 本申请实施例涉及计算机领域,具体涉及应用平台技术领域,尤其涉及宿主应用能力扩展方法、装置、设备以及存储介质。

### 背景技术

[0002] 宿主应用需要安装在客户端上才能运行。宿主应用集成了小程序开源能力。小程序开发者可以使用小程序开源能力开发小程序。小程序无需用户安装即可在宿主应用上运行。

[0003] 目前,在小程序中,针对宿主应用的私有能力,提供extension扩展机制将宿主应用的私有能力注入到swan上供小程序调用。

### 发明内容

[0004] 本申请实施例提出了一种宿主应用能力扩展方法、装置、设备以及存储介质。

[0005] 第一方面,本申请实施例提出了一种宿主应用能力扩展方法,包括:对扩展能力模板中的可用能力配置文件进行校验,其中,扩展能力模板是利用方法集配置化生成的,方法集循环利用不同扩展能力模板中的相同源文件,可用能力配置文件包含在宿主应用上可用的能力的信息;对扩展能力模板进行打包,构建宿主应用扩展包。

[0006] 第二方面,本申请实施例提出了一种宿主应用能力扩展装置,包括:第一校验模块,被配置成对扩展能力模板中的可用能力配置文件进行校验,其中,扩展能力模板是利用方法集配置化生成的,方法集循环利用不同扩展能力模板中的相同源文件,可用能力配置文件包含在宿主应用上可用的能力的信息;构建模块,被配置成对扩展能力模板进行打包,构建宿主应用扩展包。

[0007] 第三方面,本申请实施例提出了一种电子设备,包括:至少一个处理器;以及与至少一个处理器通信连接的存储器;其中,存储器存储有可被至少一个处理器执行的指令,指令被至少一个处理器执行,以使至少一个处理器能够执行如第一方面中任一实现方式描述的方法。

[0008] 第四方面,本申请实施例提出了一种存储有计算机指令的非瞬时计算机可读存储介质,计算机指令用于使计算机执行如第一方面中任一实现方式描述的方法。

[0009] 第五方面,本申请实施例提出了一种计算机程序产品,包括计算机程序,计算机程序在被处理器执行时实现如第一方面中任一实现方式描述的方法。

[0010] 本申请实施例提供的宿主应用能力扩展方法、装置、设备以及存储介质,首先对利用方法集配置化生成的扩展能力模板中的可用能力配置文件进行校验;然后对扩展能力模板进行打包,构建宿主应用扩展包。为宿主应用开发者提供一种通过自动化配置实现小程序个性化能力扩展的方法集,该方法集可实现配置化生成扩展能力模板、打包构建宿主应用扩展包等功能,使得宿主应用可以无障碍、低成本接入。

[0011] 应当理解,本部分所描述的内容并非旨在标识本公开的实施例的关键或重要特

征,也不用于限制本公开的范围。本公开的其它特征将通过以下的说明书而变得容易理解。

### 附图说明

[0012] 通过阅读参照以下附图所作的对非限制性实施例所作的详细描述,本申请的其它特征、目的和优点将会变得更明显。附图用于更好地理解本方案,不构成对本申请的限定。

其中:

[0013] 图1是根据本申请的宿主应用能力扩展方法的一个实施例的流程图;

[0014] 图2是根据本申请的宿主应用能力扩展方法的又一个实施例的流程图;

[0015] 图3是生成扩展能力模板的示意图;

[0016] 图4是生成描述表的示意图;

[0017] 图5是生成能力文档的示意图;

[0018] 图6是生成单元测试文件的示意图;

[0019] 图7是发布宿主应用扩展包的示意图;

[0020] 图8是根据本申请的宿主应用能力扩展装置的一个实施例的结构示意图;

[0021] 图9是用来实现本申请实施例的宿主应用能力扩展方法的电子设备的框图。

### 具体实施方式

[0022] 以下结合附图对本申请的示范性实施例做出说明,其中包括本申请实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本申请的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0023] 需要说明的是,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组合。下面将参考附图并结合实施例来详细说明本申请。

[0024] 图1示出了根据本申请的宿主应用能力扩展方法的一个实施例的流程100。该宿主应用能力扩展方法包括以下步骤:

[0025] 步骤101,对扩展能力模板中的可用能力配置文件进行校验。

[0026] 在本实施例中,宿主应用能力扩展方法的执行主体可以对扩展能力模板中的可用能力配置文件(如canIUse)进行校验。

[0027] 实践中,宿主应用需要安装在客户端上才能运行。宿主应用集成了小程序开源能力。小程序开发者可以使用小程序开源能力开发小程序。小程序无需用户安装即可在宿主应用上运行。通常,宿主应用中可以设置接入小程序的接口,小程序开发者可以将开发完成的小程序通过接口设置于宿主应用中。从而实现无需用户安装即可在宿主应用上使用小程序,减少了客户端的存储开支。

[0028] 通常,宿主应用私有的个性化能力可以通过扩展机制注入到swan上供小程序调用。通过这种机制,对基础端能力和私有端能力进行了有效隔离。宿主应用可扩展的个性化能力的类型多样。例如,API(Application Programming Interface,应用程序编程接口)方法支持纯客户端能力、纯前端能力、复杂能力等等。

[0029] 在实际场景中,宿主应用的不同类型的个性化能力的配置不同,因此,若要扩展宿主应用的不同类型的个性化能力就需要配置化生成不同的扩展能力模板。其中,扩展能力

模板可以为具有扩展能力的模板,宿主应用的一种个性化能力对应一个扩展能力模板。

[0030] 在本实施例中,扩展能力模板可以是利用方法集配置化生成的。其中,方法集可以是包含大量方法的工具,可实现配置化生成扩展能力模板的能力。方法集支持交互式命令行进行配置,宿主应用的开发者只需要根据交互式命令行的提示进行相应的选择,即可生成对应的扩展能力模板。无需宿主应用的开发者主动定位需要扩展的个性化能力的类型,也无需了解不同类型的个性化能力的不同配置,即可采用方法集配置化生成规范化的扩展能力模板,大大降低了宿主应用开发者的理解成本。

[0031] 并且,为了实现配置化生成扩展能力模板,还可以提前分析不同类型的个性化能力的扩展能力模板的异同点。对于相同点,方法集可以生成源文件。由于在配置不同扩展能力模板时均会使用到这些源文件,因此这些源文件被循环多次利用,从而提高了源文件的利用率。例如,多个扩展能力模板中均包括一个模块或组件。在首次利用方法集配置化生成其中一个扩展能力模板时,会保存这个模块或组件。待配置化生成其余扩展能力模板时,只需要获取这个模块或组件进行使用即可,从而实现模板或组件的循环利用。

[0032] 此外,扩展能力模板中包括一个可用能力配置文件。可用能力配置文件可以包含在宿主应用上可用的能力的信息,用于告知小程序开发者哪些能力在宿主应用上可用。对于小程序运行时而言,可用能力配置文件的内容格式规范都有强要求,格式不正确,或参数不正确,都可能造成能力不可用。方法集支持对可用能力配置文件的格式、参数等的校验,可提前校验宿主应用开发的能力内容,提高宿主应用的个性化能力的可用性。

[0033] 步骤102,对扩展能力模板进行打包,构建宿主应用扩展包。

[0034] 在本实施例中,上述执行主体可以对扩展能力模板进行打包,构建宿主应用扩展包(如swan.js)。其中,宿主应用的个性化能力最终需要打包构建,输出一个宿主应用扩展包。方法集还可以实现打包构建宿主应用扩展包的能力。利用方法集构建宿主应用扩展包,能够避免因打包方式不合规造成的能力不可用,极大的降低了宿主应用的接入成本。

[0035] 本申请实施例提供的宿主应用能力扩展方法,首先对利用方法集配置化生成的扩展能力模板中的可用能力配置文件进行校验;然后对扩展能力模板进行打包,构建宿主应用扩展包。为宿主应用开发者提供一种通过自动化配置实现小程序个性化能力扩展的方法集,该方法集可实现配置化生成扩展能力模板、打包构建宿主应用扩展包等功能,使得宿主应用可以无障碍、低成本接入。利用方法集配置化生成扩展能力模板,对宿主应用的开发者要求较低,既无需主动定位需要扩展的个性化能力的类型,又无需了解不同类型的个性化能力的不同配置,即可采用方法集配置化生成规范化的扩展能力模板,大大降低了宿主应用开发者的理解成本。由于对宿主应用的开发者要求较低,无法保证方法集直接配置化生成的扩展能力模板可用,因此,在生成扩展能力模板之后,对扩展能力模板进行校验,能够提高宿主应用的个性化能力的可用性。并且,在配置不同扩展能力模板时,方法集循环利用其中都包括的相同的源文件,提高了源文件的利用率。

[0036] 继续参考图2,其示出了根据本申请的宿主应用能力扩展方法的又一个实施例的流程200。该宿主应用能力扩展方法包括以下步骤:

[0037] 步骤201,利用方法集配置化生成扩展能力模板。

[0038] 在本实施例中,宿主应用能力扩展方法的执行主体可以利用方法集配置化生成扩展能力模板。

[0039] 实践中,宿主应用需要安装在客户端上才能运行。宿主应用集成了小程序开源能力。小程序开发者可以使用小程序开源能力开发小程序。小程序无需用户安装即可在宿主应用上运行。通常,宿主应用中可以设置接入小程序的接口,小程序开发者可以将开发完成的小程序通过接口设置于宿主应用中。从而实现无需用户安装即可在宿主应用上使用小程序,减少了客户端的存储开支。

[0040] 通常,宿主应用私有的个性化能力可以通过扩展机制注入到swan上供小程序调用。通过这种机制,对基础端能力和私有端能力进行了有效隔离。宿主应用可扩展的个性化能力的类型多样。例如,API (Application Programming Interface,应用程序编程接口) 方法支持纯客户端能力、纯前端能力、复杂能力等等。

[0041] 在实际场景中,宿主应用的不同类型的个性化能力的配置不同,因此,若要扩展宿主应用的不同类型的个性化能力就需要配置化生成不同的扩展能力模板。其中,扩展能力模板可以为具有扩展能力的模板,宿主应用的一种个性化能力对应一个扩展能力模板。

[0042] 在本实施例中,扩展能力模板可以是利用方法集配置化生成的。其中,方法集可以是包含大量方法的工具,可实现配置化生成扩展能力模板的能力。方法集支持交互式命令行进行配置,宿主应用的开发者只需要根据交互式命令行的提示进行相应的选择,即可生成对应的扩展能力模板。无需宿主应用的开发者主动定位需要扩展的个性化能力的类型,也无需了解不同类型的个性化能力的不同配置,即可采用方法集配置化生成规范化的扩展能力模板,大大降低了宿主应用开发者的理解成本。

[0043] 并且,为了实现配置化生成扩展能力模板,还可以提前分析不同类型的个性化能力的扩展能力模板的异同点。对于相同点,方法集可以生成源文件。由于在配置不同扩展能力模板时均会使用到这些源文件,因此这些源文件被循环多次利用,从而提高了源文件的利用率。例如,多个扩展能力模板中均包括一个模块或组件。在首次利用方法集配置化生成其中一个扩展能力模板时,会保存这个模块或组件。待配置化生成其余扩展能力模板时,只需要获取这个模块或组件进行使用即可,从而实现模板或组件的循环利用。

[0044] 在本实施例的一些可选的实现方式中,方法集主要通过以下两种方式配置化生成扩展能力模板:

[0045] 其一,方法集支持交互式命令行进行模板配置,生成规范化的扩展能力模板。具体地,根据交互式命令行的提示,开发者只需要选择扩展能力项目对应的选项,即可创建规范的扩展能力项目。

[0046] 其二,方法集支持使用交互式命令行自动化生成预设类型的单个扩展能力项目。具体地,使用交互式命令行自动化生成预设类型的扩展能力项目,并存放在对应的项目路径下,方便查找。例如,宿主应用开发者选择生成有复杂上下文API的模板,则方法集会去复制模板目录下的文件,并替换里面的内容,最终生成新的文件存入宿主应用开发者的项目里。

[0047] 为了便于理解,图3示出了生成扩展能力模板的示意图。如图3所示,extension模板中的API/组件模板包括纯前端API的模板、有复杂上下文的API的模板someContext等。宿主应用开发者选择生成有复杂上下文的API的模板abcContext,方法集复制有复杂上下文的API的模板someContext目录下的文件,并替换里面的内容,生成新的文件存入宿主应用开发者的项目里。

[0048] 步骤202,利用方法集自动化生成描述表。

[0049] 在本实施例中,宿主应用能力扩展方法的执行主体可以利用方法集自动化生成描述表(如json文件)。其中,方法集还可以实现自动化生成描述表的能力。描述表可以是脚本语言(如JavaScript)与客户端之间的通信接口表示。脚本语言需要通过表示通信接口的描述表,去生成调用API的方案。

[0050] 在本实施例的一些可选的实现方式中,方法集支持交互式命令行生成描述表。具体地,根据交互式命令行的提示,输入命令行字段;利用方法集对命令行字段进行组装拼接,生成描述表。其中,描述表包括命令行字段。命令行字段可以包括但不限于以下至少一项:API的名称、调起客户端能力的方式和客户端方法的存放路径等等。对于外部宿主应用而言,理解这套通信机制并不容易。而利用方法集,宿主应用开发者不需要理解原理,只需要简单的输入交互式命令行字段,通信调起方式和描述表参数等都交由方法集进行组装拼接,便可以自动生成一份标准化的端能力描述表。此方法不止针对前端开发者,也针对客户端开发者,客户端开发者只需将生成的描述表文件配置在客户端即可。

[0051] 为了便于理解,图4示出了生成描述表的示意图。如图4所示,宿主应用的开发者根据交互式命令行的提示输入操作系统iOS/Andorid、调起方式和apiname等。方法集对命令行字段进行组装拼接,生成描述表。

[0052] 步骤203,根据描述表生成能力文档。

[0053] 在本实施例中,上述执行主体可以根据描述表生成能力文档。其中,方法集还可以提供根据描述表自动生成对应能力文档的能力。描述表可以理解成是API的一份配置文件。一份能力描述表可以映射出该API在调用时所必需的参数、回调。根据描述表可以生成对应能力文档,供宿主应用开发者使用参考,降低了人力维护成本。

[0054] 为了便于理解,图5示出了生成能力文档的示意图。如图5所示,方法集从描述表中抽取属性名、类型、必填和默认值等字段,生成能力文档。

[0055] 步骤204,根据描述表生成单元测试文件。

[0056] 在本实施例中,上述执行主体可以根据描述表生成单元测试文件。其中,方法集还可以提供根据描述表自动生成单元测试文件(如单测case)的能力。

[0057] 在本实施例的一些可选的实现方式中,为了保证线上能力的质量,为宿主应用提供单元测试框架,供宿主应用开发者在线下先检验出能力的可用性。由于描述表可以理解成是API的一份配置文件,利用方法集,可实现由描述表自动生成单元测试文件的能力。其中,利用方法集生成的单元测试文件可以包括正常调用能力的事件和异常调用能力的事件。异常调用能力的事件可以包括但不限于以下至少一项:缺少参数、参数类型错误和参数不合要求等。通常,单元测试文件可以包括一种正常调用能力的事件和缺少参数、参数类型错误和参数不合要求三种异常调用能力的事件,能够拦截住最基本的API不可用情况。其余异常调用能力的事件可以由宿主应用开发者自行补充。

[0058] 为了便于理解,图6示出了生成单元测试文件的示意图。如图6所示,方法集从描述表中抽取一种正常调用能力的事件和三种异常调用能力的事件。其中,一种正常调用能力的事件是:swan.testAPI的正常调用。三种异常调用能力的事件是:swan.testAPI的异常调用,缺少参数;swan.testAPI的异常调用,参数类型错误;swan.testAPI的异常调用,参数不合要求。



[0059] 步骤205,对扩展能力模板中的可用能力配置文件和描述表进行校验。

[0060] 在本实施例中,上述执行主体可以对扩展能力模板中的可用能力配置文件(如canIUse)和描述表进行校验。其中,扩展能力模板中包括一个可用能力配置文件。可用能力配置文件可以包含在宿主应用上可用的能力的信息,用于告知小程序开发者哪些能力在宿主应用上可用。对于小程序运行时而言,描述表和可用能力配置文件的内容格式规范都有强要求,格式不正确,或参数不正确,都可能造成能力不可用。方法集支持对描述表和可用能力配置文件的格式、参数等的校验,可提前校验宿主应用开发的能力内容,提高宿主应用的个性化能力的可用性。

[0061] 步骤206,对扩展能力模板进行打包,构建宿主应用扩展包。

[0062] 在本实施例中,上述执行主体可以对扩展能力模板进行打包,构建宿主应用扩展包(如swan.js)。其中,宿主应用的个性化能力最终需要打包构建,输出一个宿主应用扩展包。方法集还可以实现打包构建宿主应用扩展包的能力。利用方法集构建宿主应用扩展包,能够避免因打包方式不合规造成的能力不可用,极大的降低了宿主应用的接入成本。

[0063] 步骤207,将宿主应用扩展包发送到服务端进行上线发布。

[0064] 在本实施例中,上述执行主体可以将宿主应用扩展包发送到服务端。服务端可以对宿主应用扩展包进行上线发布。宿主应用可以获取上线发布的宿主应用扩展包,以供宿主应用使用。利用方法集构建的宿主应用扩展包,也能避免因打包方式不合规造成的能力不可用,极大地降低了宿主应用的接入成本。

[0065] 步骤208,在本地控制台输出宿主应用扩展包的二维码和替换工具的二维码。

[0066] 在本实施例中,上述执行主体可以在本地控制台输出宿主应用扩展包的二维码和替换工具的二维码。宿主应用可以获取上线发布的宿主应用扩展包,以供宿主应用使用。方法集还可以实现打包构建宿主应用扩展包的能力,可在本地控制台输出宿主应用扩展包的二维码和替换工具的二维码,便于宿主应用开发调试、测试验证。其中,替换工具是开源宿主应用调试工具小程序。同时,利用方法集构建的宿主应用扩展包,也能避免因打包方式不合规造成的能力不可用,极大地降低了宿主应用的接入成本。

[0067] 为了便于理解,图7示出了发布宿主应用扩展包的示意图。如图7所示,宿主应用开发者利用方法集打包构建宿主应用扩展包,对宿主应用扩展包进行上线发布。同时,利用方法集在本地控制台输出宿主应用扩展包的二维码和替换工具的二维码,便于宿主应用本地测试。

[0068] 从图2中可以看出,与图1对应的实施例相比,本实施例中的宿主应用能力扩展方法增加了利用方法集自动化生成描述表、根据描述表生成能力文档和单元测试文件的步骤。由此,本实施例描述的方案中的方法集还可以实现自动化生成描述表的能力,宿主应用开发者不需要理解原理,只需要简单的输入交互式命令行字段,通信调起方式和描述表参数都交由方法集进行组装拼接,便可以自动生成一份标准化的端能力描述表。并且,根据描述表自动生成对应能力文档,供宿主应用开发者使用参考,降低了人力维护成本。根据描述表生成单元测试文件,供宿主应用开发者在线下先检验出能力的可用性。利用方法集扩展宿主能力,规范化宿主应用私有能力的扩展方式;提升宿主应用开发者的开发效率,优化开发者体验;提供本地打包构建功能,便于宿主应用开发者验证功能,从而提高扩展能力的可用性;保证宿主应用扩展包的编译产出正确,保证线上扩展能力的质量。

[0069] 进一步参考图8,作为对上述各图所示方法的实现,本申请提供了一种宿主应用能力扩展装置的一个实施例,该装置实施例与图1所示的方法实施例相对应,该装置具体可以应用于各种电子设备中。

[0070] 如图8所示,本实施例的宿主应用能力扩展装置800可以包括:第一校验模块801和构建模块802。其中,第一校验模块801,被配置成对扩展能力模板中的可用能力配置文件进行校验,其中,扩展能力模板是利用方法集配置化生成的,方法集循环利用不同扩展能力模板中的相同源文件,可用能力配置文件包含在宿主应用上可用的能力的信息;构建模块802,被配置成对扩展能力模板进行打包,构建宿主应用扩展包。

[0071] 在本实施例中,宿主应用能力扩展装置800中:第一校验模块801和构建模块802的具体处理及其所带来的技术效果可分别参考图1对应实施例中的步骤101-102的相关说明,在此不再赘述。

[0072] 在本实施例的一些可选的实现方式中,宿主应用能力扩展装置800还包括:发布模块,被配置成将宿主应用扩展包发送到服务端进行上线发布。

[0073] 在本实施例的一些可选的实现方式中,宿主应用能力扩展装置800还包括:第二生成模块,被配置成利用方法集自动化生成描述表,其中,描述表是脚本语言与客户端之间的通信接口表示;第二校验模块,被配置成对描述表进行校验。

[0074] 在本实施例的一些可选的实现方式中,第三生成模块,被配置成根据描述表生成能力文档。

[0075] 在本实施例的一些可选的实现方式中,第四生成模块,被配置成根据描述表生成单元测试文件。

[0076] 在本实施例的一些可选的实现方式中,宿主应用能力扩展装置800还包括第一生成模块,第一生成模块被配置成:根据交互式命令行的提示,选择扩展能力项目对应的选项,创建扩展能力项目;或者使用交互式命令行自动化生成预设类型的扩展能力项目,并存放在对应的项目路径下,其中,方法集支持交互式命令行进行配置。

[0077] 在本实施例的一些可选的实现方式中,第二生成模块进一步被配置成:根据交互式命令行的提示,输入命令行字段,其中,命令行字段包括以下至少一项:应用程序编程接口API的名称、调起客户端能力的方式和客户端方法的存放路径;利用方法集对命令行字段进行组装拼接,生成描述表。

[0078] 在本实施例的一些可选的实现方式中,第四生成模块进一步被配置成:利用方法集,由描述表自动生成单元测试文件,其中,单元测试文件包括正常调用能力的事件和异常调用能力的事件,异常调用能力的事件包括以下至少一项:缺少参数、参数类型错误和参数不合要求。

[0079] 在本实施例的一些可选的实现方式中,宿主应用能力扩展装置800还包括:输出模块,被配置成在本地控制台输出宿主应用扩展包的二维码和替换工具的二维码。

[0080] 根据本申请的实施例,本申请还提供了一种电子设备、一种可读存储介质和一种计算机程序产品。

[0081] 图9示出了可以用来实施本公开的实施例的示例电子设备900的示意性框图。电子设备旨在表示各种形式的数字计算机,诸如,膝上型计算机、台式计算机、工作台、个人数字助理、服务器、刀片式服务器、大型计算机、和其它适合的计算机。电子设备还可以表示各种

形式的移动装置,诸如,个人数字处理、蜂窝电话、智能电话、可穿戴设备和其它类似的计算装置。本文所示的部件、它们的连接和关系、以及它们的功能仅仅作为示例,并且不意在限制本文中描述的和/或者要求的本公开的实现。

[0082] 如图9所示,设备900包括计算单元901,其可以根据存储在只读存储器 (ROM) 902中的计算机程序或者从存储单元908加载到随机访问存储器 (RAM) 903中的计算机程序,来执行各种适当的动作和处理。在RAM 903中,还可存储设备900操作所需的各种程序和数据。计算单元901、ROM 902以及RAM 903通过总线904彼此相连。输入/输出 (I/O) 接口905也连接至总线904。

[0083] 设备900中的多个部件连接至I/O接口905,包括:输入单元906,例如键盘、鼠标等;输出单元907,例如各种类型的显示器、扬声器等;存储单元908,例如磁盘、光盘等;以及通信单元909,例如网卡、调制解调器、无线通信收发机等。通信单元909允许设备900通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0084] 计算单元901可以是各种具有处理和计算能力的通用和/或专用处理组件。计算单元901的一些示例包括但不限于中央处理单元 (CPU)、图形处理单元 (GPU)、各种专用的人工智能 (AI) 计算芯片、各种运行机器学习模型算法的计算单元、数字信号处理器 (DSP)、以及任何适当的处理器、控制器、微控制器等。计算单元901执行上文所描述的各个方法和处理,例如宿主应用能力扩展方法。例如,在一些实施例中,宿主应用能力扩展方法可被实现为计算机软件程序,其被有形地包含于机器可读介质,例如存储单元908。在一些实施例中,计算机程序的部分或者全部可以经由ROM 902和/或通信单元909而被载入和/或安装到设备900上。当计算机程序加载到RAM 903并由计算单元901执行时,可以执行上文描述的宿主应用能力扩展方法的一个或多个步骤。备选地,在其他实施例中,计算单元901可以通过其他任何适当的方式(例如,借助于固件)而被配置为执行宿主应用能力扩展方法。

[0085] 本文中以上描述的系统和技术和各种实施方式可以在数字电子电路系统、集成电路系统、场可编程门阵列 (FPGA)、专用集成电路 (ASIC)、专用标准产品 (ASSP)、芯片上系统的系统 (SOC)、负载可编程逻辑设备 (CPLD)、计算机硬件、固件、软件、和/或它们的组合中实现。这些各种实施方式可以包括:实施在一个或者多个计算机程序中,该一个或者多个计算机程序可在包括至少一个可编程处理器的可编程系统上执行和/或解释,该可编程处理器可以是专用或者通用可编程处理器,可以从存储系统、至少一个输入装置、和至少一个输出装置接收数据和指令,并且将数据和指令传输至该存储系统、该至少一个输入装置、和该至少一个输出装置。

[0086] 用于实施本公开的方法的程序代码可以采用一个或多个编程语言的任何组合来编写。这些程序代码可以提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器或控制器,使得程序代码当由处理器或控制器执行时使流程图和/或框图中所规定的功能/操作被实施。程序代码可以完全在机器上执行、部分地在机器上执行,作为独立软件包部分地在机器上执行且部分地在远程机器上执行或完全在远程机器或服务器上执行。

[0087] 在本公开的上下文中,机器可读介质可以是有形的介质,其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备,或者上述内容的任何合

适组合。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、便携式计算机盘、硬盘、随机存取存储器 (RAM)、只读存储器 (ROM)、可擦除可编程只读存储器 (EPROM 或快闪存储器)、光纤、便捷式紧凑盘只读存储器 (CD-ROM)、光学储存设备、磁储存设备、或上述内容的任何合适组合。

[0088] 为了提供与用户的交互,可以在计算机上实施此处描述的系统和技术,该计算机具有:用于向用户显示信息的显示装置(例如,CRT(阴极射线管)或者LCD(液晶显示器)监视器);以及键盘和指向装置(例如,鼠标或者轨迹球),用户可以通过该键盘和该指向装置来将输入提供给计算机。其它种类的装置还可以用于提供与用户的交互;例如,提供给用户的反馈可以是任何形式的传感反馈(例如,视觉反馈、听觉反馈、或者触觉反馈);并且可以用任何形式(包括声输入、语音输入或者、触觉输入)来接收来自用户的输入。

[0089] 可以将此处描述的系统和技术实施在包括后台部件的计算系统(例如,作为数据服务器)、或者包括中间件部件的计算系统(例如,应用服务器)、或者包括前端部件的计算系统(例如,具有图形用户界面或者网络浏览器的用户计算机,用户可以通过该图形用户界面或者该网络浏览器来与此处描述的系统和技术实施方式交互)、或者包括这种后台部件、中间件部件、或者前端部件的任何组合的计算系统中。可以通过任何形式或者介质的数字数据通信(例如,通信网络)来将系统的部件相互连接。通信网络的示例包括:局域网(LAN)、广域网(WAN)和互联网。

[0090] 计算机系统可以包括客户端和服务端。客户端和服务端一般远离彼此并且通常通过通信网络进行交互。通过在相应的计算机上运行并且彼此具有客户端-服务端关系的计算机程序来产生客户端和服务端的关系。

[0091] 应该理解,可以使用上面所示的各种形式的流程,重新排序、增加或删除步骤。例如,本发公开中记载的各步骤可以并行地执行也可以顺序地执行也可以不同的次序执行,只要能够实现本公开公开的技术方案所期望的结果,本文在此不进行限制。

[0092] 上述具体实施方式,并不构成对本公开保护范围的限制。本领域技术人员应该明白的是,根据设计要求和因素,可以进行各种修改、组合、子组合和替代。任何在本公开的精神和原则之内所作的修改、等同替换和改进等,均应包含在本公开保护范围之内。

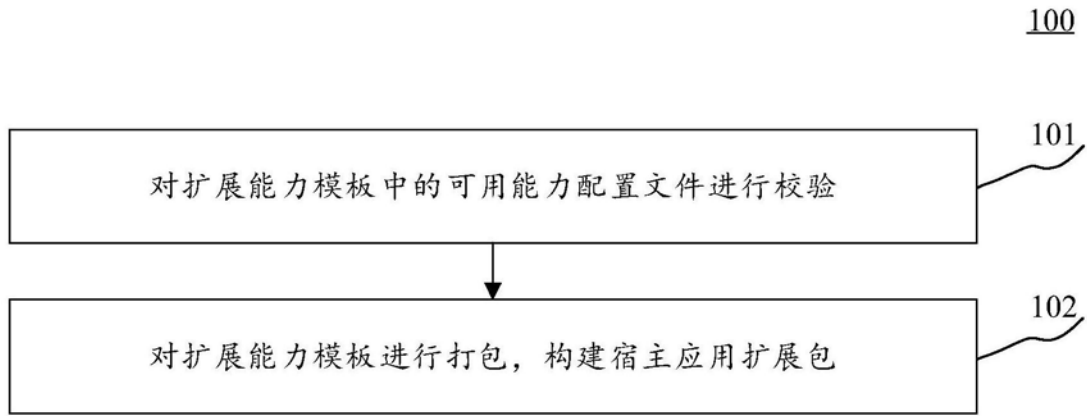


图1

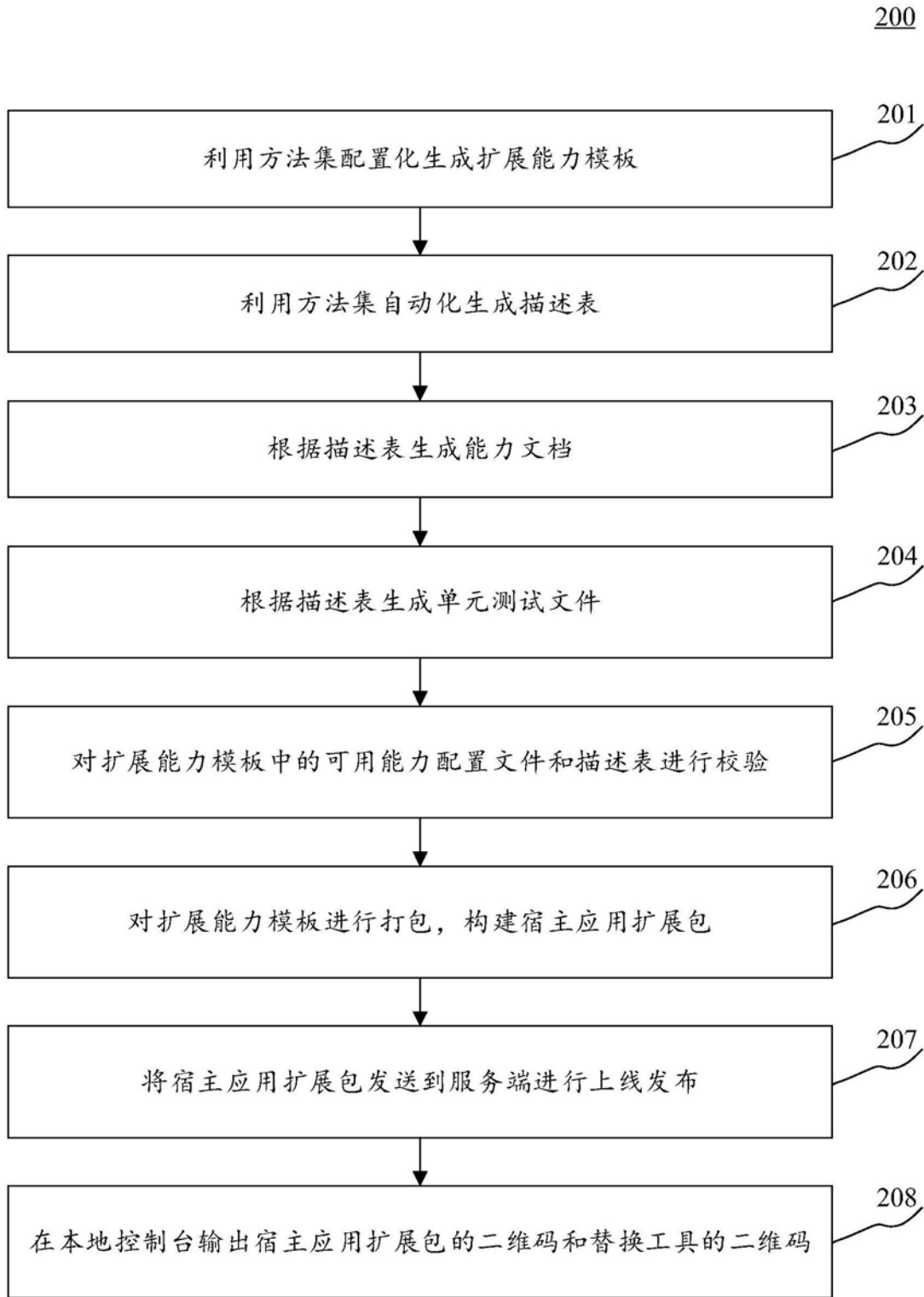


图2

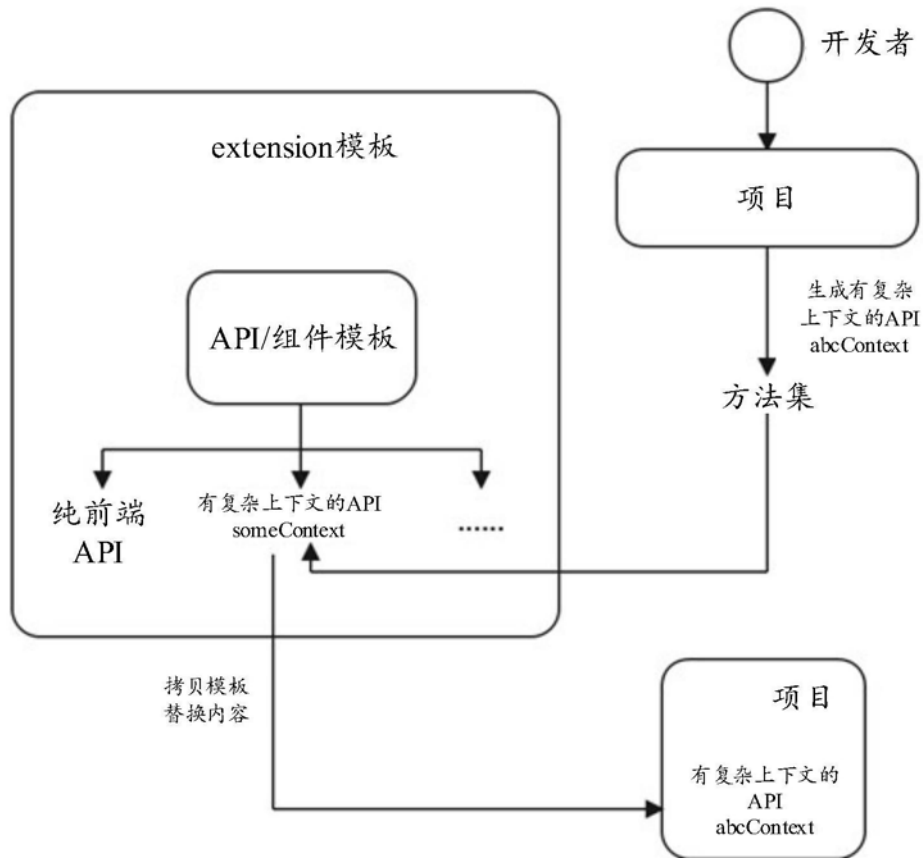


图3

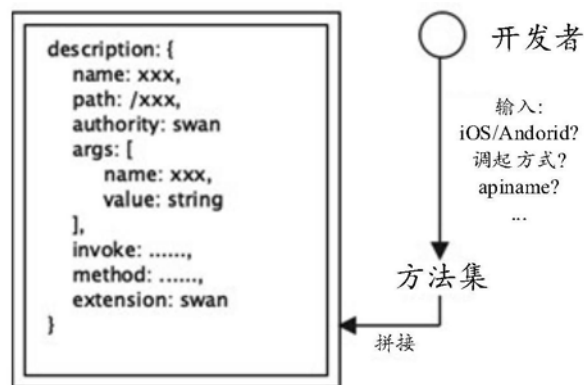


图4

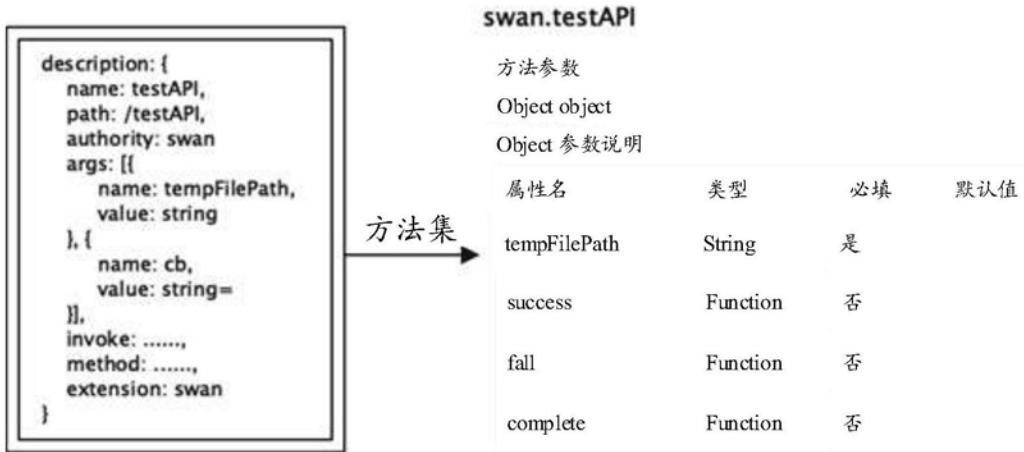


图5

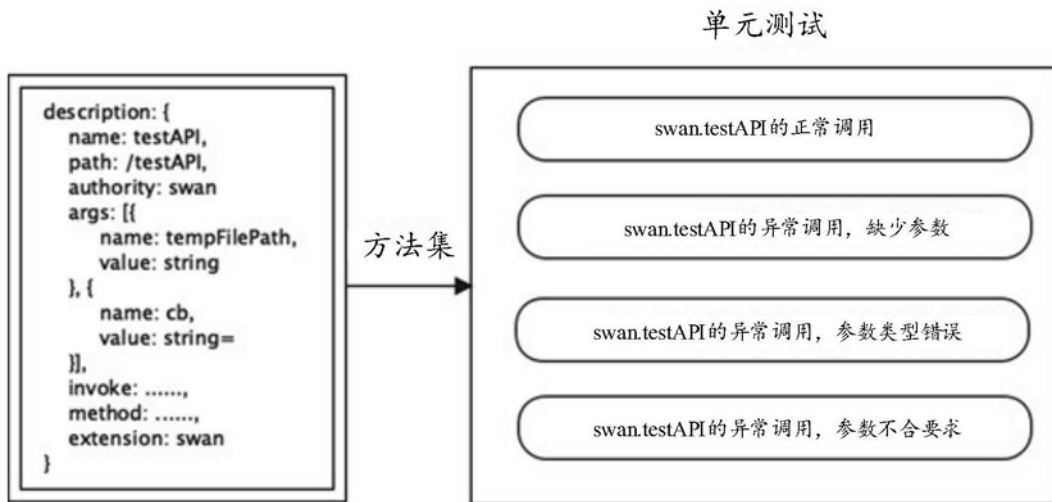


图6



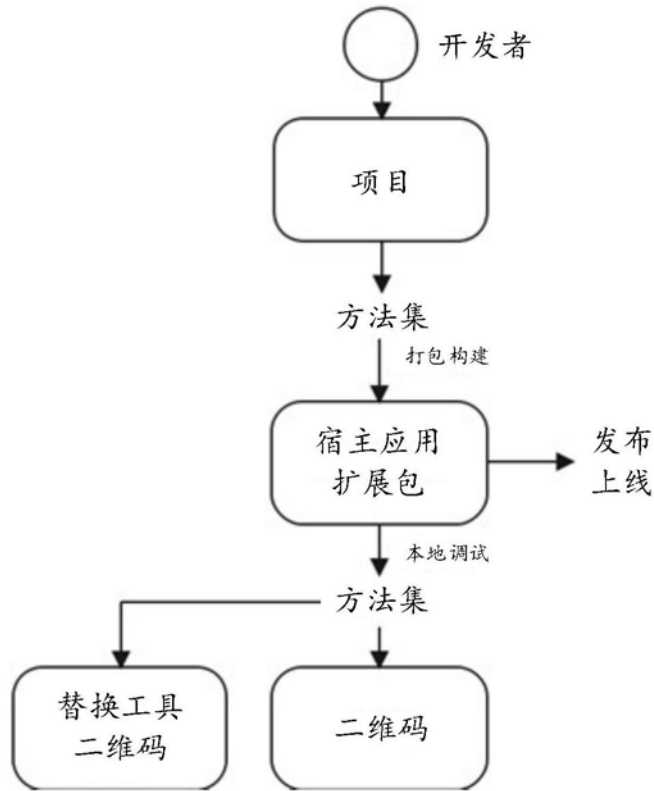


图7

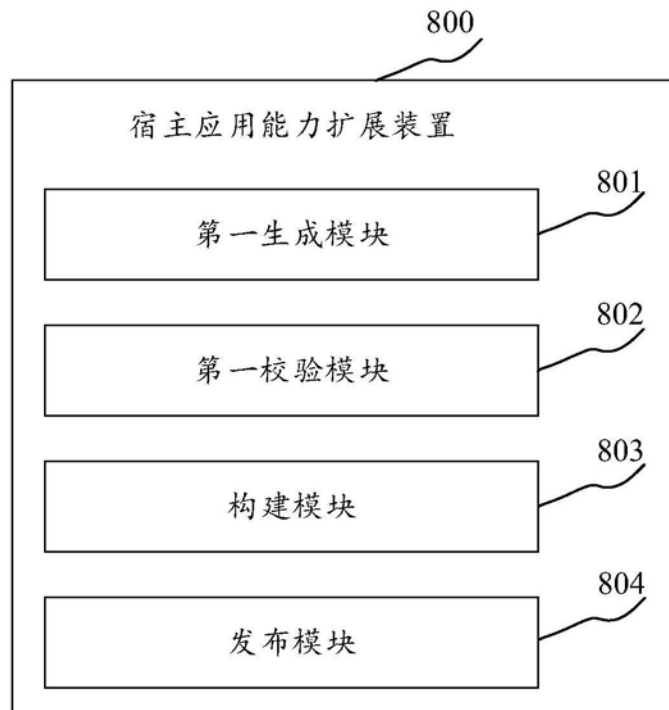


图8

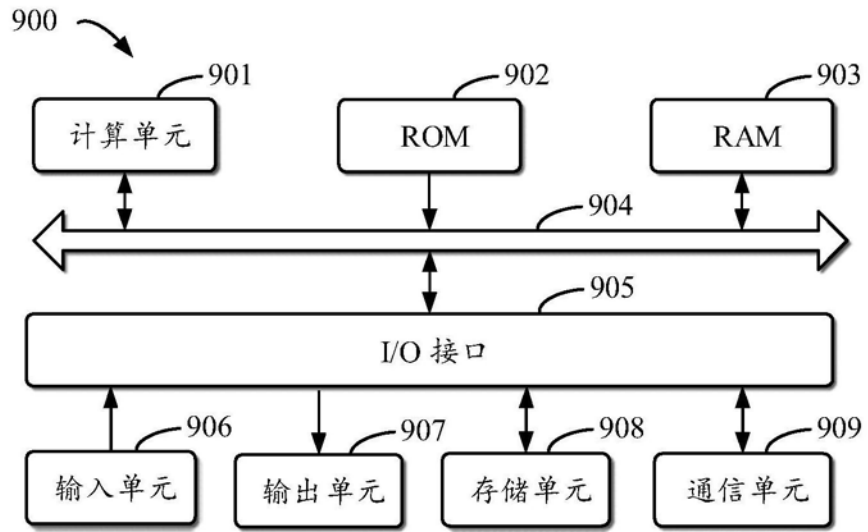


图9