



(12) 发明专利

(10) 授权公告号 CN 111782183 B

(45) 授权公告日 2024.03.01

(21) 申请号 202010610352.4

(22) 申请日 2020.06.29

(65) 同一申请的已公布的文献号
申请公布号 CN 111782183 A

(43) 申请公布日 2020.10.16

(73) 专利权人 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 许多凯 李朋飞 郭金

(74) 专利代理机构 北京市汉坤律师事务所
11602
专利代理师 姜浩然 吴丽丽

(51) Int. Cl.
G06F 8/20 (2018.01)
G06F 8/41 (2018.01)

(56) 对比文件

CN 104077140 A, 2014.10.01

CN 104516730 A, 2015.04.15

CN 110377290 A, 2019.10.25

CN 110851139 A, 2020.02.28

CN 111324333 A, 2020.06.23

US 2017344363 A1, 2017.11.30

Ma Liangli; Wang Houxiang; Li

Yongjie. Using Component Metadata based on Dependency Relationships Matrix to improve the Testability of Component-based Software. 《2006 1st International Conference on Digital Information Management》. 2007, 全文.

季永炜; 徐洁洁; 赵浩强; 黄秋霞. 基于依赖关系模型的复杂软件系统架构管理方法的研究. 电脑编程技巧与维护. 2016, (06), 全文.

审查员 焦天栋

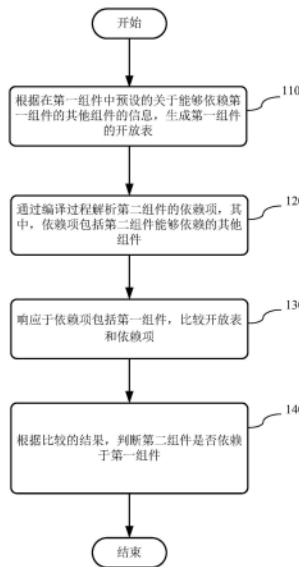
权利要求书2页 说明书6页 附图2页

(54) 发明名称

用于判断组件依赖关系的方法及设备、电子设备和介质

(57) 摘要

本公开提供一种用于判断组件依赖关系的方法及设备、电子设备和介质, 涉及计算机工程架构、云计算技术领域。用于判断组件依赖关系的方法包括: 根据在第一组件中预设的关于能够依赖该第一组件的其他组件的信息, 生成第一组件的开放表, 该开放表表示所述第一组件能够被其他哪些组件依赖; 通过编译过程解析第二组件的依赖项, 其中, 该依赖项包括第二组件能够依赖的其他组件; 响应于依赖项包括第一组件, 比较开放表和依赖项; 以及根据比较的结果, 判断第二组件是否依赖于第一组件。



1. 一种由计算机实现的用于在代码工程中判断组件依赖关系的方法,包括:

根据在第一组件中预设的关于能够依赖所述第一组件的其他组件的信息,生成所述第一组件的开放表,所述开放表表示所述第一组件能够被其他哪些组件依赖,其中,所述第一组件属于所述代码工程的第一作用域,所述第一作用域包括用于实现第一功能的多个组件,所述多个组件被分别配置用于表征组件依赖关系的第一参数值或第二参数值,其中,所述第一参数值用于表示相应的组件只能被所述第一作用域内的其他组件依赖,所述第二参数值用于表示相应的组件能够被所述代码工程中的任一其他组件依赖;

通过编译过程解析第二组件的依赖项,其中,所述依赖项包括所述第二组件能够依赖的其他组件;

响应于所述依赖项包括所述第一组件,比较所述开放表和所述依赖项;以及

根据比较的结果,判断所述第二组件是否依赖于所述第一组件。

2. 如权利要求1所述的方法,其中,所述开放表列出能够依赖所述第一组件的其他组件,在所述代码工程的任一其他组件都能够依赖所述第一组件的情况下,所述开放表为空;并且其中,

所述根据比较的结果,判断所述第二组件是否依赖于所述第一组件包括:

响应于所述开放表不为空并且所述第二组件不在所述开放表中,生成依赖关系错误的提示。

3. 如权利要求1所述的方法,其中,所述根据在第一组件中预设的关于能够依赖所述第一组件的其他组件的信息生成所述第一组件的开放表包括:

根据配置给所述第一组件的参数值,生成所述第一组件的开放表。

4. 如权利要求2或3所述的方法,其中,所述方法还包括:

响应于所述依赖关系错误的提示,从所述第二组件的依赖项中删除所述第一组件。

5. 一种用于在代码工程中判断组件依赖关系的设备,包括:

生成模块,被配置为根据在第一组件中预设的关于能够依赖所述第一组件的其他组件的信息,生成所述第一组件的开放表,所述开放表表示所述第一组件能够被其他哪些组件依赖,其中,所述第一组件属于所述代码工程的第一作用域,所述第一作用域包括用于实现第一功能的多个组件,所述多个组件被分别配置用于表征组件依赖关系的第一参数值或第二参数值,其中,所述第一参数值用于表示相应的组件只能被所述第一作用域内的其他组件依赖,所述第二参数值用于表示相应的组件能够被所述代码工程中的任一其他组件依赖;

解析模块,被配置为通过编译过程解析第二组件的依赖项,其中,所述依赖项包括所述第二组件能够依赖的其他组件;

比较模块,被配置为响应于所述依赖项包括所述第一组件,比较所述开放表和所述依赖项;以及

判断模块,被配置为根据比较的结果,判断所述第二组件是否依赖于所述第一组件。

6. 如权利要求5所述的设备,其中,所述开放表列出能够依赖所述第一组件的其他组件,在所述代码工程的任一其他组件都能够依赖所述第一组件的情况下,所述开放表为空;并且其中,

所述判断模块还被配置为:响应于所述开放表不为空并且所述第二组件不在所述开放

表中,生成依赖关系错误的提示。

7.如权利要求5所述的设备,其中,所述根据在第一组件中预设的关于能够依赖所述第一组件的其他组件的信息生成所述第一组件的开放表包括:

根据配置给所述第一组件的参数值,生成所述第一组件的开放表。

8.如权利要求6或7所述的设备,其中,还包括纠错模块,被配置为响应于所述依赖关系错误的提示,从所述第二组件的依赖项中删除所述第一组件。

9.一种电子设备,包括:

处理器;以及

存储程序的存储器,所述程序包括指令,所述指令在由所述处理器执行时使所述处理器执行根据权利要求1-4中任一项所述的方法。

10.一种存储程序的计算机可读存储介质,所述程序包括指令,所述指令在由电子设备的处理器执行时,致使所述电子设备执行根据权利要求1-4中任一项所述的方法。

用于判断组件依赖关系的方法及设备、电子设备和介质

技术领域

[0001] 本公开涉及计算机工程架构、云计算技术领域,特别涉及用于判断组件依赖关系的方法及设备、电子设备和介质。

背景技术

[0002] 在程序设计中,其代码工程的架构趋势是代码组件化。组件开发需要做到高内聚低耦合,组件之间的依赖关系需规范化,对不合理的依赖关系需要通过解耦技术实现解耦。目前还没有限制组件依赖关系的具体实现,这会导致编译检查依赖关系是否符合架构设计力不从心。组件的依赖关系在多人维护迭代过程中,很容易造成依赖关系的复杂化,网状依赖关系,影响组件可独立输出的能力。对于超级工程而言,只由人工来审核代码组件之间的依赖关系成本偏高。

[0003] 在此部分中描述的方法不一定是之前已经设想到或采用的方法。除非另有指明,否则不应假定此部分中描述的任何方法仅因其包括在此部分中就被认为是现有技术。类似地,除非另有指明,否则此部分中提及的问题不应认为在任何现有技术中已被公认。

发明内容

[0004] 根据本公开的一个方面,提供了一种由计算机实现的用于在代码工程中判断组件依赖关系的方法,包括:根据在第一组件中预设的关于能够依赖第一组件的其他组件的信息,生成第一组件的开放表,该开放表表示所述第一组件能够被其他哪些组件依赖;通过编译过程解析第二组件的依赖项,其中,依赖项包括第二组件能够依赖的其他组件;响应于依赖项包括第一组件,比较开放表和依赖项;以及根据比较的结果,判断第二组件是否依赖于第一组件。

[0005] 根据本公开的另一个方面,提供了一种用于在代码工程中判断组件依赖关系的设备,包括:生成模块,被配置为根据在第一组件中预设的关于能够依赖第一组件的其他组件的信息,生成第一组件的开放表,该开放表表示所述第一组件能够被其他哪些组件依赖;解析模块,被配置为通过编译过程解析第二组件的依赖项,其中,依赖项包括第二组件能够依赖的其他组件;比较模块,被配置为响应于依赖项包括第一组件,比较开放表和依赖项;以及判断模块,被配置为根据比较的结果,判断第二组件是否依赖于第一组件。

[0006] 根据本公开的另一个方面,提供了一种电子设备,包括:处理器;以及存储程序的存储器,该程序包括指令,该指令在由处理器执行时使处理器执行本公开中所述的方法。

[0007] 根据本公开的另一个方面,提供了一种存储程序的计算机可读存储介质,该程序包括指令,该指令在由电子设备的处理器执行时,致使电子设备执行本公开中所述的方法。

[0008] 根据本公开的另一个方面,提供了一种计算机程序产品,包括计算机程序,该计算机程序在被处理器执行时实现本公开中所述的方法。

[0009] 根据本公开的一个方面,由计算机实现的用于在代码工程中判断组件依赖关系的方法可以自动化检测不合理依赖关系,提升项目管理效率并减少人工代码审核的负担。

[0010] 根据在下文中所描述的实施例,本公开的这些和其它方面将是清楚明白的,并且将参考在下文中所描述的实施例而被阐明。

附图说明

[0011] 附图示例性地示出了实施例并且构成说明书的一部分,与说明书的文字描述一起用于讲解实施例的示例性实施方式。所示出的实施例仅出于例示的目的,并不限制权利要求的范围。在所有附图中,相同的附图标记指代类似但不一定相同的要素。

[0012] 图1是示出根据示例性实施例的用于在代码工程中判断组件依赖关系的方法的流程图;

[0013] 图2是示出根据示例性实施例的作用域内和外的组件及其之间的依赖关系的示意图;

[0014] 图3是示出根据示例性实施例的用于在代码工程中判断组件依赖关系的设备的示意图;以及

[0015] 图4是示出能够应用于示例性实施例的示例性计算设备的结构框图。

具体实施方式

[0016] 在本公开中,除非另有说明,否则使用术语“第一”、“第二”等来描述各种要素不意图限定这些要素的位置关系、时序关系或重要性关系,这种术语只是用于将一个元件与另一元件区分开。在一些示例中,第一要素和第二要素可以指向该要素的同一实例,而在某些情况下,基于上下文的描述,它们也可以指代不同实例。

[0017] 在本公开中对各种所述示例的描述中所使用的术语只是为了描述特定示例的目的,而并非旨在进行限制。除非上下文另外明确地表明,如果不特意限定要素的数量,则该要素可以是一个也可以是多个。此外,本公开中所使用的术语“和/或”涵盖所列出的项目中的任何一个以及全部可能的组合方式。

[0018] 在例如安卓Android代码工程中,没有限制其组件依赖关系的具体实现。例如,某一功能由三个组件A、B、C组成,组件A属于对外的接口模块,组件B、C属于实现接口内容的实现组件,实际开发中,其他组件如果要使用该功能,只需依赖组件A,如果额外依赖了组件B、C,那么这些依赖就是不合理的。安卓工程目前尚无控制机制,组件的依赖关系在多人维护迭代过程中,很容易造成依赖关系的复杂化,影响组件可独立输出的能力。

[0019] 图1是示出根据示例性实施例的用于在代码工程中判断组件依赖关系的方法的流程图。如图1所示,该方法包括:根据在第一组件中预设的关于能够依赖第一组件的其他组件的信息,生成第一组件的开放表(步骤110);通过编译过程解析第二组件的依赖项,其中,依赖项包括第二组件能够依赖的其他组件(步骤120);响应于依赖项包括第一组件,比较开放表和依赖项(步骤130);以及根据比较的结果,判断第二组件是否依赖于第一组件(步骤140)。

[0020] 所述第一组件可以是代码工程中能够被其他组件依赖的任意组件。在一些实施例中,第一组件的开放表为对外接口的形式,例如设置对外接口expose,expose是一个列表,可以指定当前组件只能被指定的组件依赖,即只能被expose列表中的组件依赖。

[0021] 根据一些实施例,开放表中列出能够依赖该第一组件的其他组件,用以表示该第

一组件能够被哪些其他组件依赖,即相当于该第一组件对于哪些其他组件是“开放”的。在代码工程的任一其他组件都能够依赖第一组件的情况下,该开放表为空;并且其中,根据比较的结果,判断第二组件是否依赖于第一组件包括:响应于开放表不为空并且第二组件不在该开放表中,生成依赖关系错误的提示。

[0022] 根据一些实施例,代码工程可以是代码组件化的需规范其组件依赖关系的任意工程,包括安卓代码工程等。

[0023] 根据一些实施例,生成依赖关系错误的提示包括:输出提示信息以提示该第二组件不能依赖该第一组件或者该第二组件依赖该第一组价为不合理的依赖关系。

[0024] 根据一些实施例,第一组件属于代码工程的第一作用域,该第一作用域包括用于实现第一功能的多个组件,该多个组件被分别配置用于表征组件依赖关系的第一参数值或第二参数值;其中,第一参数值用于表示相应的组件只能被该第一作用域内的其他组件依赖;以及第二参数值,用于表示相应的组件能够被代码工程中的任一其他组件依赖。

[0025] 根据一些实施例,首先可以在该代码工程中定义一个或多个作用域,每一个作用域可包括一个或多个组件以用于实现其相应的某一个功能。根据一些实施例,设置对外接口access,其中,当access被配置为第一参数值private时,则表示该组件只能被该作用域内的组件依赖,当access被配置为第二参数值public时,则表示该组件可以被任意组件依赖。

[0026] 根据一些实施例,根据在第一组件中预设的关于能够依赖第一组件的其他组件的信息生成第一组件的开放表包括:根据配置给第一组件的参数值,生成第一组件的开放表。

[0027] 根据一些实施例,依据access可以确定哪些组件可以依赖当前组件,所以可以将这些组件添加到当前组件的expose列表中。

[0028] 例如,如图2所示,定义了功能A的一个作用域,功能A的各个组件都属于该作用域。其中,作用域内组件a-api声明access为public、组件a-impl声明access为private;作用域外组件c声明expose为a-impl;其中例如在组件的迭代开发过程中添加了组件M的依赖关系以使得组件M依赖组件a-api、a-impl和c。根据本公开的方法,将access的声明转换为组件具体的expose列表可以得到:组件a-api的expose为空、组件a-impl的expose为a-api。在编译阶段,依次遍历每个组件以解析所述组件的依赖项。例如,在编译阶段解析组件M的所有依赖:组件M依赖组件a-api,而组件a-api的expose列表为空,可以被任意组件依赖,所以允许组件M依赖a-api;组件M依赖组件a-impl,组件a-impl的expose列表里只包含组件a-api,所以不允许组件M依赖组件a-impl,报错;组件M依赖组件c,组件c的expose里只包含组件a-impl,所以不允许组件M依赖组件c,报错。又例如,在编译阶段解析组件a-impl的所有依赖,组件a-impl依赖组件c,组件c的expose里包含组件a-impl,所以允许组件a-impl依赖c。由此直到遍历完所有组件,并解耦报错的不合理的依赖关系。

[0029] 根据一些实施例,该方法还包括:响应于该依赖关系错误的提示,从第二组件的依赖项中删除该第一组件。即,响应于该依赖关系错误的提示,利用解耦技术解耦第二组件依赖第一组件的依赖关系。

[0030] 根据本公开的另一方面,如图3所示,还提供一种用于在代码工程中判断组件依赖关系的设备300,包括:生成模块310,被配置为根据在第一组件中预设的关于能够依赖该第一组件的其他组件的信息,生成该第一组件的开放表,该开放表表示所述第一组件能够被

其他哪些组件依赖;解析模块320,被配置为通过编译过程解析第二组件的依赖项,其中,该依赖项包括该第二组件能够依赖的其他组件;比较模块330,被配置为响应于该依赖项包括第一组件,比较该开放表和该依赖项;以及判断模块340,被配置为根据比较的结果,判断该第二组件是否依赖于该第一组件。

[0031] 根据一些实施例,开放表列出能够依赖所述第一组件的其他组件,在所述代码工程的任一其他组件都能够依赖所述第一组件的情况下,所述开放表为空;并且其中,所述判断模块还被配置为:响应于所述开放表不为空并且所述第二组件不在所述开放表中,生成依赖关系错误的提示。

[0032] 根据一些实施例,第一组件属于所述代码工程的第一作用域,所述第一作用域包括用于实现第一功能的多个组件,所述多个组件被分别配置用于表征组件依赖关系的第一参数值或第二参数值;其中,所述第一参数值,用于表示相应的组件只能被所述第一作用域内的其他组件依赖;以及所述第二参数值,用于表示相应的组件能够被所述代码工程中的任一其他组件依赖。

[0033] 根据一些实施例,根据在第一组件中预设的关于能够依赖所述第一组件的其他组件的信息生成所述第一组件的开放表包括:根据配置给所述第一组件的参数值,生成所述第一组件的开放表。

[0034] 根据一些实施例,还包括纠错模块,被配置为响应于所述依赖关系错误的提示,从所述第二组件的依赖项中删除所述第一组件。

[0035] 根据本公开的另一方面,还提供一种电子设备,可以包括:处理器;以及存储程序的存储器,所述程序包括指令,所述指令在由所述处理器执行时使所述处理器执行根据上述的用于在代码工程中判断组件依赖关系的方法。

[0036] 根据本公开的另一方面,还提供一种存储程序的计算机可读存储介质,所述程序包括指令,所述指令在由电子设备的处理器执行时,致使所述电子设备执行根据上述的用于在代码工程中判断组件依赖关系的方法。

[0037] 参见图4所示,现将描述计算设备2000,其是可以应用于本公开的各方面的硬件设备(电子设备)的示例。计算设备2000可以是配置为执行处理和/或计算的任何机器,可以是但不限于工作站、服务器、台式计算机、膝上型计算机、平板计算机、个人数字助理、机器人、智能电话、车载计算机或其任何组合。上述用于在代码工程中判断组件依赖关系的方法可以全部或至少部分地由计算设备2000或类似设备或系统实现。

[0038] 计算设备2000可以包括(可能经由一个或多个接口)与总线2002连接或与总线2002通信的元件。例如,计算设备2000可以包括总线2002、一个或多个处理器2004、一个或多个输入设备2006以及一个或多个输出设备2008。一个或多个处理器2004可以是任何类型的处理器,并且可以包括但不限于一个或多个通用处理器和/或一个或多个专用处理器(例如特殊处理芯片)。输入设备2006可以是能向计算设备2000输入信息的任何类型的设备,并且可以包括但不限于鼠标、键盘、触摸屏、麦克风和/或遥控器。输出设备2008可以是能呈现信息的任何类型的设备,并且可以包括但不限于显示器、扬声器、视频/音频输出终端、振动器和/或打印机。计算设备2000还可以包括非暂时性存储设备2010或者与非暂时性存储设备2010连接,非暂时性存储设备可以是非暂时性的并且可以实现数据存储的任何存储设备,并且可以包括但不限于磁盘驱动器、光学存储设备、固态存储器、软盘、柔性盘、硬盘、磁

带或任何其他磁介质,光盘或任何其他光学介质、ROM(只读存储器)、RAM(随机存取存储器)、高速缓冲存储器和/或任何其他存储器芯片或盒、和/或计算机可从其读取数据、指令和/或代码的任何其他介质。非暂时性存储设备2010可以从接口拆卸。非暂时性存储设备2010可以具有用于实现上述方法和步骤的数据/程序(包括指令)/代码。计算设备2000还可以包括通信设备2012。通信设备2012可以是使得能够与外部设备和/或与网络通信的任何类型的设备或系统,并且可以包括但不限于调制解调器、网卡、红外通信设备、无线通信设备和/或芯片组,例如蓝牙TM设备、1302.11设备、WiFi设备、WiMax设备、蜂窝通信设备和/或类似物。

[0039] 计算设备2000还可以包括工作存储器2014,其可以是存储对处理器2004的工作有用的程序(包括指令)和/或数据的任何类型的工作存储器,并且可以包括但不限于随机存取存储器和/或只读存储器设备。

[0040] 软件要素(程序)可以位于工作存储器2014中,包括但不限于操作系统2016、一个或多个应用程序2018、驱动程序和/或其他数据和代码。用于执行上述方法和步骤的指令可以被包括在一个或多个应用程序2018中,并且上述用于在代码工程中判断组件依赖关系的方法可以通过由处理器2004读取和执行一个或多个应用程序2018的指令来实现。更具体地,上述用于在代码工程中判断组件依赖关系的方法中,步骤110~步骤140可以例如通过处理器2004执行具有步骤110~步骤140的指令的应用程序2018而实现。此外,上述用于在代码工程中判断组件依赖关系的方法中的其它步骤可以例如通过处理器2004执行具有执行相应步骤中的指令的应用程序2018而实现。软件要素(程序)的指令的可执行代码或源代码可以存储在非暂时性计算机可读存储介质(例如上述存储设备2010)中,并且在执行时可以被存入工作存储器2014中(可能被编译和/或安装)。软件要素(程序)的指令的可执行代码或源代码也可以从远程位置下载。

[0041] 还应该理解,可以根据具体要求而进行各种变型。例如,也可以使用定制硬件,和/或可以用硬件、软件、固件、中间件、微代码,硬件描述语言或其任何组合来实现特定元件。例如,所公开的方法和设备中的一些或全部可以通过使用根据本公开的逻辑和算法,用汇编语言或硬件编程语言(诸如VERILOG, VHDL, C++)对硬件(例如,包括现场可编程门阵列(FPGA)和/或可编程逻辑阵列(PLA)的可编程逻辑电路)进行编程来实现。

[0042] 还应该理解,前述方法可以通过服务器-客户端模式来实现。例如,客户端可以接收用户输入的数据并将所述数据发送到服务器。客户端也可以接收用户输入的数据,进行前述方法中的一部分处理,并将处理所得到的数据发送到服务器。服务器可以接收来自客户端的数据,并且执行前述方法或前述方法中的另一部分,并将执行结果返回给客户端。客户端可以从服务器接收到方法的执行结果,并例如可以通过输出设备呈现给用户。

[0043] 还应该理解,计算设备2000的组件可以分布在网络上。例如,可以使用一个处理器执行一些处理,而同时可以由远离该一个处理器的另一个处理器执行其他处理。计算系统2000的其他组件也可以类似地分布。这样,计算设备2000可以被解释为在多个位置执行处理的分布式计算系统。例如,计算设备2000可以实现为云平台的一部分。该云平台抽象云的硬件(例如,服务器)和软件资源的底层功能。资源可以包括在远离计算设备2000的服务器上执行计算机处理时可以使用的应用和/或数据。资源还可以包括通过因特网和/或通过诸如蜂窝或Wi-Fi网络的订户网络提供的服务。

[0044] 虽然已经参照附图描述了本公开的实施例或示例,但应理解,上述的方法、系统和设备仅仅是示例性的实施例或示例,本发明的范围并不由这些实施例或示例限制,而是仅由授权后的权利要求书及其等同范围来限定。实施例或示例中的各种要素可以被省略或者可由其等同要素替代。此外,可以通过不同于本公开中描述的次序来执行各步骤。进一步地,可以以各种方式组合实施例或示例中的各种要素。重要的是随着技术的演进,在此描述的很多要素可以由本公开之后出现的等同要素进行替换。

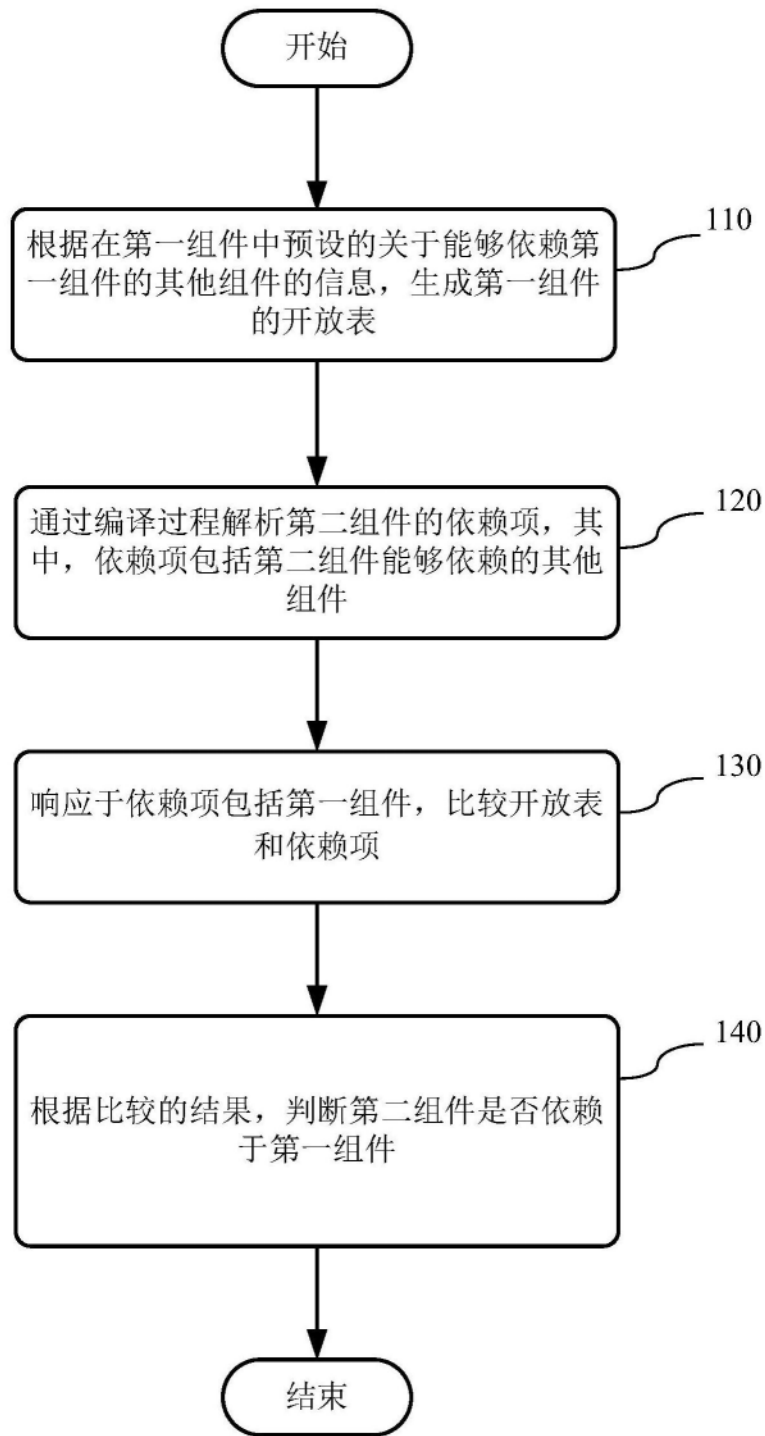


图1

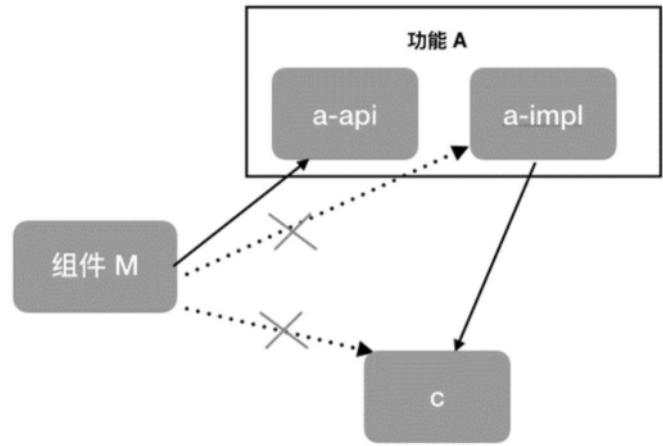


图2

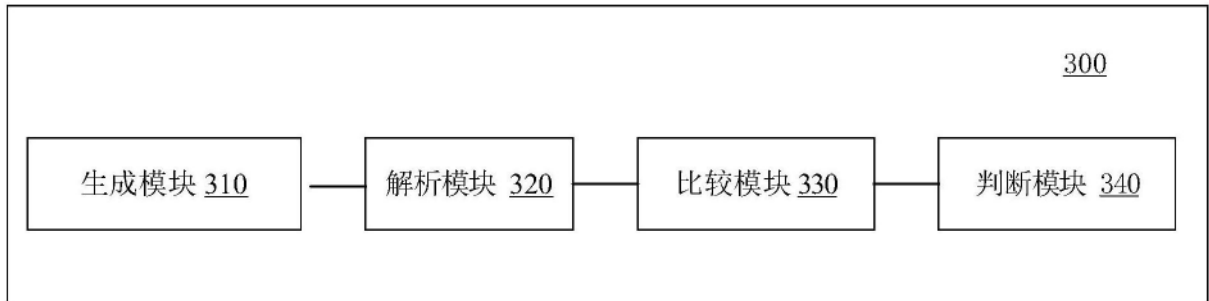


图3

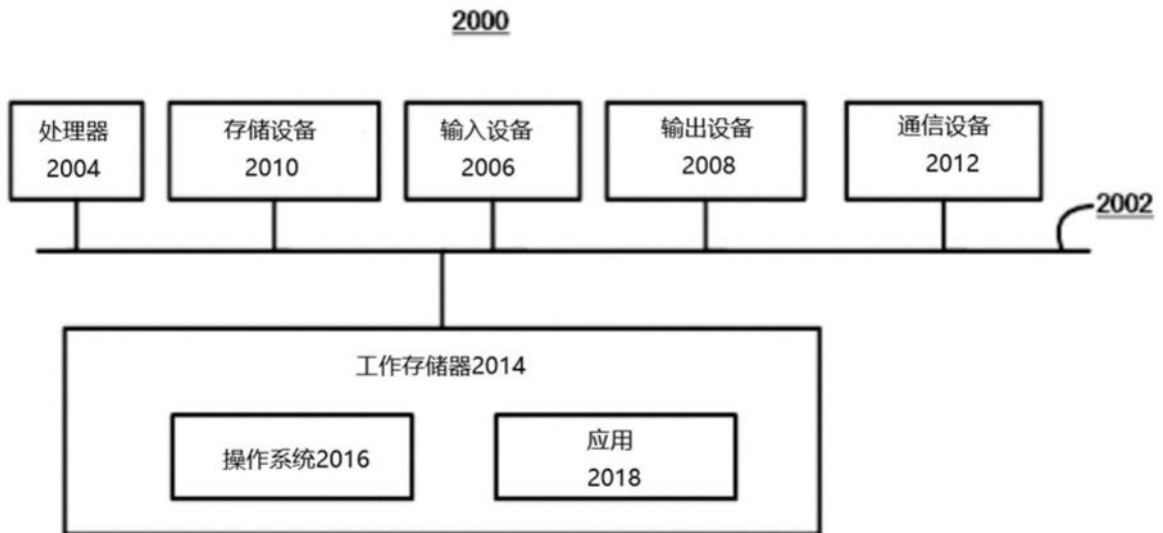


图4