



## (12)发明专利申请

(10)申请公布号 CN 107622008 A

(43)申请公布日 2018.01.23

(21)申请号 201610556248.5

(22)申请日 2016.07.14

(71)申请人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼资本大厦一座四  
层847号邮箱

(72)发明人 车薇 边爽

(74)专利代理机构 北京博思佳知识产权代理有  
限公司 11415

代理人 陈蕾

(51)Int.Cl.

G06F 11/36(2006.01)

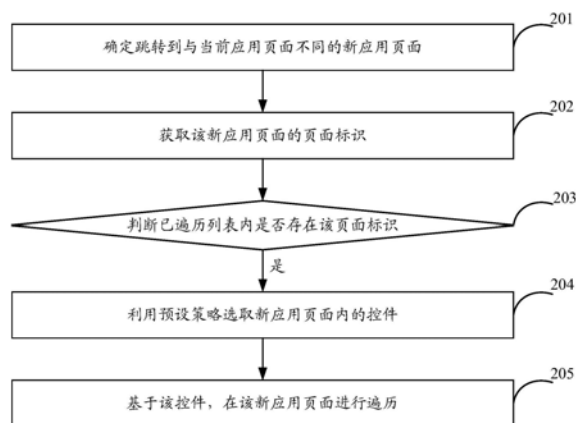
权利要求书3页 说明书12页 附图3页

### (54)发明名称

一种应用页面的遍历方法和装置

### (57)摘要

本申请提供一种应用页面的遍历方法和装置,该方法包括:确定跳转到与当前应用页面不同的新应用页面;获取所述新应用页面的页面标识;判断已遍历列表内是否存在所述页面标识;如果是,则利用预设策略选取所述新应用页面内的控件;基于所述控件,在所述新应用页面进行遍历。通过本申请的技术方案,在跳转到新应用页面后,如果之前已经访问过该新应用页面,就可以直接获取到该新应用页面对应的最后访问控件,继而可以使用最后访问控件在新应用页面进行遍历,而不需要从新应用页面的第一个控件开始,在新应用页面进行遍历,从而去除重复遍历的过程,节省了整个应用遍历的时间,为更高效、更稳定的遍历自动化测试带来可行性。



1. 一种应用页面的遍历方法,其特征在于,所述方法包括:  
确定跳转到与当前应用页面不同的新应用页面;  
获取所述新应用页面的页面标识;  
判断已遍历列表内是否存在所述页面标识;  
如果是,则利用预设策略选取所述新应用页面内的控件;  
基于所述控件,在所述新应用页面进行遍历。
2. 根据权利要求1所述的方法,其特征在于,所述确定跳转到与当前应用页面不同的新应用页面的过程,具体包括:  
如果在当前应用页面发生页面跳转,则比较跳转后页面的控件数量与当前应用页面的控件数量;  
如果二者不同,则确定跳转到与当前应用页面不同的新应用页面;  
或者,  
如果在当前应用页面发生页面跳转,则比较跳转后页面的控件树形结构与当前应用页面的控件树形结构;  
如果二者不完全相同,则确定跳转到与当前应用页面不同的新应用页面。
3. 根据权利要求1所述的方法,其特征在于,所述利用预设策略选取所述新应用页面内的控件的过程,具体包括:  
从所述已遍历列表内选取所述页面标识对应的最后访问控件;或者,  
统计所述新应用页面内的每个控件的访问次数;  
利用每个控件的访问次数,选取所述新应用页面内的控件。
4. 根据权利要求3所述的方法,其特征在于,所述确定跳转到与当前应用页面不同的新应用页面之后,所述方法进一步包括:  
获取当前应用页面的最后访问控件;  
在所述已遍历列表内记录当前应用页面的页面标识与所述当前应用页面的最后访问控件的对应关系。
5. 根据权利要求3或4所述的方法,其特征在于,所述新应用页面或者所述当前应用页面的页面标识,具体包括:所述新应用页面或者所述当前应用页面的页面路径信息,或者基于页面内的控件确定的唯一标识。
6. 根据权利要求5所述的方法,其特征在于,在基于页面内的控件确定所述新应用页面或者所述当前应用页面的页面标识的过程中,所述方法还包括:  
将所述新应用页面或者所述当前应用页面内的控件树形结构转换为一个字符串;  
对所述字符串进行运算;  
从运算结果中选取指定位数的数值;  
将所述指定位数的数值确定为所述新应用页面或者所述当前应用页面的页面标识。
7. 根据权利要求1所述的方法,其特征在于,所述判断已遍历列表内是否存在所述页面标识之后,所述方法进一步包括:  
如果否,则在所述已遍历列表内记录所述新应用页面的页面标识;  
从所述新应用页面的第一个控件开始,在所述新应用页面进行遍历。
8. 根据权利要求1所述的方法,其特征在于,所述基于所述控件,在所述新应用页面进

行遍历的过程,具体包括:

基于所述新应用页面内的各控件的顺序,从所述新应用页面的所述控件的下一个控件开始,在所述新应用页面进行遍历,且不再对所述控件以及所述控件前面的所有控件进行遍历。

9. 根据权利要求1所述的方法,其特征在于,所述确定跳转到与当前应用页面不同的新应用页面之后,所述方法进一步包括:

通过拍照截屏方式或者页面标题判断方式分析所述新应用页面的准确性;

如果分析出所述新应用页面准确,则执行获取所述新应用页面的页面标识的过程;

如果分析出所述新应用页面不准确,则跳转回所述当前应用页面。

10. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

在所述新应用页面进行遍历的过程中,如果在遍历过程中出现异常弹框,则关闭所述异常弹框,并继续在所述新应用页面进行遍历。

11. 一种应用页面的遍历装置,其特征在于,所述装置包括:

确定模块,用于确定跳转到与当前应用页面不同的新应用页面;

第一获取模块,用于获取所述新应用页面的页面标识;

判断模块,用于判断已遍历列表内是否存在所述页面标识;

第二获取模块,用于当判断结果为存在所述页面标识时,则利用预设策略选取所述新应用页面内的控件;

处理模块,用于基于所述控件,在所述新应用页面进行遍历。

12. 根据权利要求11所述的装置,其特征在于,

所述确定模块,具体用于在确定跳转到与当前应用页面不同的新应用页面的过程中,如果在当前应用页面发生页面跳转,则比较跳转后页面的控件数量与当前应用页面的控件数量,如果二者不同,则确定跳转到与当前应用页面不同的新应用页面;或者,如果在当前应用页面发生页面跳转,则比较跳转后页面的控件树形结构与当前应用页面的控件树形结构,如果二者不完全相同,则确定跳转到与当前应用页面不同的新应用页面。

13. 根据权利要求10所述的装置,其特征在于,

所述第二获取模块,具体用于在利用预设策略选取所述新应用页面内的控件的过程中,从所述已遍历列表内选取所述页面标识对应的最后访问控件;或者,统计所述新应用页面内的每个控件的访问次数;利用每个控件的访问次数,选取所述新应用页面内的控件。

14. 根据权利要求13所述的装置,其特征在于,

所述第一获取模块,还用于在确定模块确定跳转到与当前应用页面不同的新应用页面之后,获取当前应用页面的最后访问控件,在所述已遍历列表内记录当前应用页面的页面标识与所述当前应用页面的最后访问控件的对应关系。

15. 根据权利要求13或14所述的装置,其特征在于,所述第一获取模块获取的所述新应用页面或者当前应用页面的页面标识包括:所述新应用页面或者当前应用页面的页面路径信息,或者基于页面内的控件确定的唯一标识。

16. 根据权利要求15所述的装置,其特征在于,所述第一获取模块在基于页面内的控件确定所述新应用页面或者当前应用页面的页面标识的过程中,将所述新应用页面或者所述当前应用页面内的控件树形结构转换为一个字符串,并对所述字符串进行运算,并从运算

结果中选取指定位数的数值,并将所述指定位数的数值确定为所述新应用页面或者所述当前应用页面的页面标识。

17.根据权利要求11所述的装置,其特征在于,所述处理模块,还用于在所述判断模块判断已遍历列表内是否存在所述页面标识之后,当判断结果为不存在所述页面标识时,则在所述已遍历列表内记录所述新应用页面的页面标识,并从所述新应用页面的第一个控件开始,在所述新应用页面进行遍历。

18.根据权利要求11所述的装置,其特征在于,

所述处理模块,具体用于在基于所述控件,在所述新应用页面进行遍历的过程中,基于所述新应用页面内的各控件的顺序,从所述新应用页面的所述控件的下一个控件开始,在所述新应用页面进行遍历,且不再对所述控件以及所述控件前面的所有控件进行遍历。

19.根据权利要求11所述的装置,其特征在于,所述处理模块,还用于在所述确定模块确定跳转到与当前应用页面不同的新应用页面之后,通过拍照截屏方式或者页面标题判断方式分析所述新应用页面的准确性;如果分析出所述新应用页面准确,则通知所述第一获取模块执行获取所述新应用页面的页面标识的过程;如果分析出所述新应用页面不准确,则跳转回所述当前应用页面。

20.根据权利要求11所述的装置,其特征在于,所述处理模块,还用于在所述新应用页面进行遍历的过程中,如果在遍历过程中出现异常弹框,则关闭所述异常弹框,并继续在所述新应用页面进行遍历。

## 一种应用页面的遍历方法和装置

### 技术领域

[0001] 本申请涉及互联网技术领域,尤其涉及一种应用页面的遍历方法和装置。

### 背景技术

[0002] 应用(即APP)测试是指针对应用的软件测试工作,包括功能测试和性能测试,功能测试包括测试应用的所有功能是否可用,是否出现bug现象等,性能测试包括测试用户使用流程是否顺畅,如支付流程,注册流程等。

[0003] 目前,应用页面中包含大量的控件,控件是对数据和方法的封装,每个控件均可以有自身的属性和方法,属性是控件数据的访问者,方法是控件的简单可见的功能。为了测试应用页面的所有功能是否可用,是否出现bug现象等,则需要依次点击应用页面的每个控件,完成每个控件的测试,继而测试出每个控件的功能是否可用,是否出现bug现象等。如果有控件的功能不可用,或者出现bug现象,则基于测试结果,对该控件的软件代码进行修复。

### 发明内容

[0004] 本申请实施例提供一种应用页面的遍历方法,所述方法包括:

[0005] 确定跳转到与当前应用页面不同的新应用页面;

[0006] 获取所述新应用页面的页面标识;

[0007] 判断已遍历列表内是否存在所述页面标识;

[0008] 如果是,则利用预设策略选取所述新应用页面内的控件;

[0009] 基于所述控件,在所述新应用页面进行遍历。

[0010] 本申请实施例提供一种应用页面的遍历装置,所述装置包括:

[0011] 确定模块,用于确定跳转到与当前应用页面不同的新应用页面;

[0012] 第一获取模块,用于获取所述新应用页面的页面标识;

[0013] 判断模块,用于判断已遍历列表内是否存在所述页面标识;

[0014] 第二获取模块,用于当判断结果为存在所述页面标识时,则利用预设策略选取所述新应用页面内的控件;

[0015] 处理模块,用于基于所述控件,在所述新应用页面进行遍历。

[0016] 基于上述技术方案,本申请实施例中,在应用页面的遍历过程中,在跳转到新应用页面后,如果之前已经访问过该新应用页面,就可以获取到该新应用页面内的控件,并基于该控件在新应用页面进行遍历,而不需要从新应用页面的第一个控件开始,在新应用页面进行遍历,从而去除重复遍历的过程,节省了整个应用遍历的时间,为更高效、更稳定的遍历自动化测试带来可行性。

### 附图说明

[0017] 为了更加清楚地说明本申请实施例或者现有技术中的技术方案,下面将对本申请实施例或者现有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

附图仅仅是本申请中记载的一些实施例,对于本领域普通技术人员来讲,还可以根据这些附图获得其他的附图。

[0018] 图1是本申请一种实施方式中的应用页面的树结构的示意图;

[0019] 图2是本申请一种实施方式中的应用页面的遍历方法的流程图;

[0020] 图3是本申请一种实施方式中的控件树形结构的示例;

[0021] 图4是本申请一种实施方式中的移动终端的硬件结构图;

[0022] 图5是本申请一种实施方式中的应用页面的遍历装置的结构图。

## 具体实施方式

[0023] 在本申请使用的术语仅仅是出于描述特定实施例的目的,而非限制本申请。本申请和权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式,除非上下文清楚地表示其它含义。还应当理解,本文中使用的术语“和/或”是指包含一个或多个相关联的列出项目的任何或所有可能组合。

[0024] 应当理解,尽管在本申请可能采用术语第一、第二、第三等来描述各种信息,但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如,在不脱离本申请范围的情况下,第一信息也可以被称为第二信息,类似地,第二信息也可以被称为第一信息。取决于语境,此外,所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

[0025] 本申请实施例中提出一种应用页面的遍历方法,用于测试应用页面的所有功能是否可用,是否出现bug现象等。在应用页面的遍历过程中,由于应用页面包含大量控件,因此需要依次点击应用页面的每个控件,完成每个控件的测试,测试出每个控件的功能是否可用,是否出现bug现象。如果有控件的功能不可用或者出现bug现象,则基于测试结果,对该控件的软件代码进行修复。

[0026] 在一个例子中,对于当前应用页面的某些控件来说,在点击该控件后,会跳转到与当前应用页面不同的新应用页面。在从新应用页面跳转回当前应用页面后,由于并不知道是否已经对当前应用页面的部分或者全部控件进行过测试,因此从当前应用页面的第一个控件开始,重新在当前应用页面进行遍历,从而造成某些控件的重复遍历,而重复遍历将非常耗时。例如,应用页面1包括控件1、控件2、控件3、控件4,在完成控件1、控件2的遍历后,在点击控件3时,跳转到与应用页面1不同的应用页面2。在从应用页面2跳转回应用页面1后,从应用页面1的控件1开始,重新在应用页面1进行遍历,即会重新遍历控件1、控件2、控件3,从而造成控件1、控件2、控件3的重复遍历。

[0027] 针对上述发现,本申请实施例中提出的应用页面的遍历方法,在应用页面的遍历过程中,在从一个应用页面跳转到另一个应用页面时,可以知道是否已经对另一个应用页面的部分或者全部控件进行过测试,在对部分或者全部控件进行过测试时,不需要从第一个控件开始,重新在另一个应用页面进行遍历,从而避免控件的重复遍历。例如,应用页面1包括控件1、控件2、控件3、控件4,在完成控件1、控件2的点击后,在点击控件3时,跳转到与应用页面1不同的应用页面2,并记录应用页面1的页面标识与控件3的对应关系。在从应用页面2跳转回应用页面1后,基于上述对应关系,可以获知已经对应用页面1的部分控件进行过测试,并从应用页面1的控件3的下一个控件4开始,重新在应用页面1进行遍历,因此不会

造成控件1、控件2、控件3的重复遍历。

[0028] 为了对用户进行个性化推荐,可以根据不同地域的用户人群的行为、习惯、偏好等因素,在应用页面部署大量控件,如可以达到几百甚至上千个控件。如果采用上述重复遍历的方式,则测试时间会很长,已经不能满足要求。而采用本申请实施例的技术方案后,通过避免重复遍历,可以节省大量的测试时间。

[0029] 在一个例子中,应用页面是运行在终端设备(如PC(Personal Computer,个人计算机)、移动终端、笔记本电脑等)上的客户端显示的,以终端设备为移动终端(如手机、平板电脑、ipad等)为例,则应用页面的遍历方法,可以应用在无线自动化测试的应用场景下。其中,应用页面的遍历,是指通过遍历算法模拟用户沿着某条搜索路线,依次对应用页面的每个控件,均做一次且仅做一次访问(即点击操作),以测试控件的功能是否可用,是否出现bug现象。

[0030] 其中,上述的终端设备或者移动终端,可以为真实的终端设备或者移动终端,也可以为能够模拟出终端设备功能或者移动终端功能的服务器。

[0031] 在一个例子中,针对每个应用页面,均可以维护一个树结构,如图1所示,为一个应用页面的树结构的示例,该树结构中的每个节点均表示一个控件。在图1中,应用页面的根节点(root)可以为控件A,该控件A下存在三个子节点,分别为控件A.1、控件A.2、控件A.3,且控件A是这三个子节点的父节点。控件A.1下存在三个子节点,分别为控件A.1.1、控件A.1.2、控件A.1.3,且控件A.1是这三个子节点的父节点。控件A.1.2下存在两个子节点,分别为控件A.1.2.1、控件A.1.2.2,且控件A.1.2是这两个子节点的父节点。

[0032] 在一个例子中,应用页面中的控件,具体可以包括但不限于以下之一或者任意组合:UIA(User Interface Automation,用户界面自动化)window(窗口)、UIA Scrollview(滚动视图)、UIA navigationbar(导航栏)、UIA tableview(表格视图)、UIA button(按钮)、UIA toolbar(工具栏)、UIA Image(图像)、UIA webview(网页视图)、UIA Element(元素)、UIA tablecell(表格单元)、UIA textfield(文本框)、UIA statictext(静态文本)等。

[0033] 例如,控件A可以为UIA window,控件A.1可以为UIA Scrollview,控件A.2可以为UIA navigationbar,控件A.3可以为UIA tableview,控件A.1.1可以为UIA button,控件A.1.2可以为UIA button,控件A.1.3可以为UIA toolbar,控件A.1.2.1可以为UIA tablecell,控件A.1.2.2可以为UIA statictext等。

[0034] 在一个例子中,在应用页面的遍历过程中,应用页面的每个控件之间是有顺序关系的,如图1所示,每个控件之间的顺序关系可以是:控件A-控件A.1-控件A.1.1-控件A.1.2-控件A.1.2.1-控件A.1.2.2-控件A.1.3-控件A.2-控件A.3。

[0035] 在一个例子中,应用页面中各控件的名称就可以体现出相应的顺序关系,继而可以通过各控件的名称,获知应用页面的每个控件之间的顺序关系。例如,控件A的名称为UIA window,控件A.1的名称为UIA window.UIA Scrollview,控件A.2的名称为UIA window.UIA navigationbar,控件A.3的名称为UIA window.UIA tableview,控件A.1.1的名称为UIA window.UIA Scrollview.UIA button1,控件A.1.2的名称可以为UIA window.UIA Scrollview.UIA button2,控件A.1.3的名称可以为UIA window.UIA Scrollview.UIA toolbar,控件A.1.2.1的名称为UIA window.UIA Scrollview.UIA button2.UIA tablecell,控件A.1.2.2的名称为UIA window.UIA Scrollview.UIA button2.UIA

statictext等。

[0036] 基于上述名称可以获知,控件A下存在三个子节点,分别为控件A.1、控件A.2、控件A.3,控件A.1下存在三个子节点,分别为控件A.1.1、控件A.1.2、控件A.1.3,控件A.1.2下存在两个子节点,分别为控件A.1.2.1、控件A.1.2.2。基于此,可以建立图1所示的树结构,并确定控件顺序为:控件A-控件A.1-控件A.1.1-控件A.1.2-控件A.1.2.1-控件A.1.2.2-控件A.1.3-控件A.2-控件A.3。

[0037] 对于树结构的建立方式,以及控件顺序的确定方式,在此不再详加赘述,本申请实施例中,将基于图1所示的树结构以及控件顺序进行相关处理。

[0038] 在一个例子中,每个可以被访问的控件都可以使用一个Javascript对象来描述,因此,每个控件就是一个UIAElement。基于此,还可以将控件称为元素、或者控件元素等,为了方便描述,以控件的表述为例进行说明。

[0039] 在上述应用场景下,在应用页面(假设为应用页面1)的遍历过程中,需要依次遍历控件A、控件A.1、控件A.1.1、控件A.1.2、控件A.1.2.1、控件A.1.2.2、控件A.1.3、控件A.2、控件A.3,在上述控件的遍历过程中,参见图2所示,为本申请实施例中提出的应用页面的遍历方法的流程图,该方法包括以下步骤:

[0040] 步骤201,确定跳转到与当前应用页面不同的新应用页面。

[0041] 在一个例子中,确定跳转到与当前应用页面不同的新应用页面的过程,具体可以包括但不限于如下方式:如果在当前应用页面发生页面跳转,则比较跳转后页面的控件数量与当前应用页面的控件数量,如果二者不同,则可以确定跳转到与当前应用页面不同的新应用页面;或者,如果在当前应用页面发生页面跳转,比较跳转后页面的控件树形结构与当前应用页面的控件树形结构,如果二者不完全相同,则可以确定跳转到与当前应用页面不同的新应用页面。

[0042] 例如,当前应用页面为应用页面1时,需要依次遍历控件A、控件A.1、控件A.1.1、控件A.1.2、控件A.1.2.1、控件A.1.2.2、控件A.1.3、控件A.2、控件A.3。首先点击控件A,在应用页面1未发生页面跳转,并测试出控件A的功能是否可用,是否出现bug现象。点击控件A.1,在应用页面1未发生页面跳转,并测试出控件A.1的功能是否可用,是否出现bug现象。点击控件A.1.1,在应用页面1未发生页面跳转,并测试出控件A.1.1的功能是否可用,是否出现bug现象。点击控件A.1.2,假设在应用页面1发生页面跳转。

[0043] 由于页面跳转后的应用页面可能还是应用页面1(即重新加载了应用页面1,可以类似于点击刷新操作),也可能是一个与应用页面1不同的新应用页面,如应用页面2,因此需要确定是否跳转到与应用页面1不同的新应用页面。例如,由于应用页面1的控件数量为9,假设跳转后的应用页面的控件数量为5,则说明跳转到与应用页面1不同的新应用页面2。又例如,由于应用页面1的控件包括控件A、控件A.1、控件A.1.1、控件A.1.2、控件A.1.2.1、控件A.1.2.2、控件A.1.3、控件A.2、控件A.3,假设跳转后的应用页面的控件包括控件A、控件A.1、控件A.1.1、控件A.1.2、控件A.1.4、控件A.1.5,则二者不完全相同,说明跳转到与应用页面1不同的新应用页面2。

[0044] 在一个例子中,如果在当前应用页面发生页面跳转,则比较跳转后页面的控件数量与当前应用页面的控件数量,如果二者不同,则可以确定跳转到与应用页面1不同的新应用页面,如果二者相同,则进一步比较跳转后页面的控件树形结构与当前应用页面的



控件树形结构,如果二者不完全相同,则可以确定跳转到与当前应用页面不同的新应用页面,如果二者完全相同,则可以确定未跳转到与当前应用页面不同的新应用页面,还停留在当前应用页面。

[0045] 在一个例子中,控件树形结构可以包括应用页面内的所有控件的控件名称和控件坐标等信息,当然,控件树形结构还可以包含其它内容,在此不再赘述。如图3所示,为一个控件树形结构的示例。其中,控件坐标是指:控件在应用页面的位置,例如,移动终端的像素大小为980\*1198,则应用页面的像素大小为980\*1198,则基于控件在应用页面的位置,控件坐标可以为500\*500等,其唯一表示了应用页面的一个位置,即通过该位置可以确定控件的位置。

[0046] 在一个例子中,在确定跳转到与当前应用页面不同的新应用页面前,在进入当前应用页面时,即依次遍历控件A、控件A.1、控件A.1.1、控件A.1.2、控件A.1.2.1、控件A.1.2.2、控件A.1.3、控件A.2、控件A.3之前,就可获取当前应用页面的页面标识,并在已遍历列表内记录当前应用页面的页面标识。

[0047] 如表1所示,为已遍历列表的一个示例,在当前应用页面为应用页面1时,则可以记录应用页面1的页面标识,在后续过程中介绍该页面标识。

[0048] 表1

[0049]

页面标识	控件
应用页面1的页面标识	

[0050] 情况一:

[0051] 在一个例子中,在确定跳转到与当前应用页面不同的新应用页面后,还可以获取当前应用页面的最后访问控件,并在已遍历列表内记录当前应用页面的页面标识与当前应用页面的最后访问控件的对应关系,即在当前应用页面的页面标识下记录当前应用页面的最后访问控件。例如,由于点击控件A.1.2后,跳转到与当前应用页面不同的新应用页面,因此最后访问控件为控件A.1.2。

[0052] 在一个例子中,可以在当前应用页面的页面标识下,只记录当前应用页面的最后访问控件,如表2所示。在另一个例子中,每次点击控件后,就可以在当前应用页面的页面标识下记录该控件,并在跳转到与当前应用页面不同的新应用页面后,在当前应用页面的页面标识下记录最后访问控件,如表3所示。

[0053] 表2

[0054]

页面标识	控件
应用页面1的页面标识	控件A.1.2

[0055] 表3

[0056]

页面标识	控件
应用页面1的页面标识	控件A、控件A.1、控件A.1.1、控件A.1.2

[0057] 在一个例子中,在已遍历列表内记录的控件可以为控件的全路径名称,如最后访问控件的全路径名称。例如,控件A的名称可以为UIA window,控件A.1.2的全路径名称可以

为UIA window.UIA Scrollview.UIA button2。

[0058] 情况二：

[0059] 在一个例子中，在确定跳转到与当前应用页面不同的新应用页面后，还可以统计当前应用页面内的每个控件的访问次数，并在已遍历列表内记录当前应用页面的页面标识与当前应用页面的每个控件的访问次数的对应关系，即在当前应用页面的页面标识下记录每个控件及其访问次数，如表4所示。

[0060] 表4

[0061]

页面标识	控件
应用页面 1 的页面标识	控件 A: 3 次、控件 A.1: 3 次、控件 A.1.1: 2 次、控件 A.1.2: 2 次、控件 A.1.2.1: 2 次、控件 A.1.2.2: 1 次、控件 A.1.3: 1 次、控件 A.2: 0 次、控件 A.3: 0 次

[0062] 在一个例子中，在已遍历列表内记录的控件可以为控件的全路径名称，如控件的全路径名称。例如，控件A的名称可以为UIA window，控件A.1.2的全路径名称可以为UIA window.UIA Scrollview.UIA button2等。

[0063] 又例如，在跳转到应用页面2之后，则当前应用页面为应用页面2，且需要依次遍历应用页面2下的控件B、控件B.1、控件B.1.1、控件B.1.2、控件B.1.2.1。针对应用页面2的处理过程，与应用页面1的处理过程类似，在此不再赘述，此处先介绍从应用页面2跳转回应用页面1的过程。

[0064] 在一个例子中，在跳转到应用页面2后，不遍历应用页面2下的控件B、控件B.1、控件B.1.1、控件B.1.2、控件B.1.2.1，而是记录需要在后续过程遍历应用页面2下的控件B、控件B.1、控件B.1.1、控件B.1.2、控件B.1.2.1的信息，并跳转回应用页面1。待应用页面1的所有控件均遍历完成后，再遍历应用页面2下的控件B、控件B.1、控件B.1.1、控件B.1.2、控件B.1.2.1。

[0065] 在另一个例子中，在跳转到应用页面2后，直接遍历应用页面2下的控件B、控件B.1、控件B.1.1、控件B.1.2、控件B.1.2.1，具体遍历过程在此不再赘述，假设在点击应用页面2下的控件B.1.2时，跳转回应用页面1。

[0066] 其中，在从应用页面1跳转到应用页面2时，则当前应用页面为应用页面1，且新应用页面为应用页面2。在从应用页面2跳转回应用页面1时，则当前应用页面为应用页面2，且新应用页面为应用页面1。

[0067] 步骤202，获取该新应用页面的页面标识。

[0068] 针对上述过程中涉及到的新应用页面的页面标识以及当前应用页面的页面标识，具体可以包括但不限于：新应用页面或者当前应用页面的页面路径信息，或者基于页面（新应用页面或者当前应用页面）内的控件确定的唯一标识。

[0069] 其中，页面路径信息就是地址栏中的地址，基于该页面路径信息，可以唯一确定出一个页面，因此，可以将页面路径信息确定为页面标识。

[0070] 其中，在基于页面内的控件确定新应用页面或者当前应用页面的页面标识的过程

中,可以将新应用页面或者当前应用页面内的控件树形结构转换为一个字符串,并对该字符串进行运算,并从运算结果中选取指定位数的数值,并将该指定位数的数值确定为新应用页面或者当前应用页面的页面标识。

[0071] 在一个例子中,由于不同应用页面的控件树形结构不会完全相同,因此可以基于应用页面的控件树形结构,确定出该应用页面的唯一的页面标识。在确定应用页面的页面标识的过程中,可以将应用页面内的控件树形结构转换为一个字符串(string),如通过logElementTree()方法打印应用页面的控件树形结构,打印结果为该字符串。之后,可以对该字符串进行运算,如进行hash运算、MD5加密运算等,具体运算方式可以根据实际经验选择,本申请实施例中对此不作限制。之后,从运算结果中选取指定位数的数值,如取运算结果的后5位,取运算结果的前6位,取运算结果的第2位-第8位等,具体应该如何选取,以及选取几位,均可以根据实际经验进行选择,本申请实施例中对此不作限制。之后,就可以将该指定位数的数值,确定为应用页面的页面标识。

[0072] 步骤203,判断已遍历列表内是否存在该页面标识。如果是,执行步骤204。

[0073] 步骤204,利用预设策略选取新应用页面内的控件。

[0074] 步骤205,基于该控件,在该新应用页面进行遍历。

[0075] 其中,预设策略可以为控件不重复策略,即使用该预设策略选取的控件,尽可能保证选取的控件是一个未遍历过的控件,保证控件不重复遍历。

[0076] 针对上述情况一,利用预设策略选取新应用页面内的控件的过程,可以包括但不限于如下方式:从已遍历列表内选取该页面标识对应的最后访问控件。

[0077] 在一个例子中,该页面标识就是该新应用页面的页面标识,且该页面标识对应的最后访问控件是新应用页面内的控件。

[0078] 本申请实施例中,基于该控件(即该最后访问控件),在该新应用页面进行遍历的过程,具体可以包括但不限于如下方式:基于新应用页面内的各控件的顺序,从新应用页面的所述控件的下一个控件开始,在新应用页面进行遍历,且不再对所述控件以及所述控件前面的所有控件进行遍历。

[0079] 本申请实施例中,在判断已遍历列表内是否存在该页面标识之后,如果判断结果是否为否,则在一个例子中,在已遍历列表内记录新应用页面的页面标识,并从新应用页面的第一个控件开始,在新应用页面进行遍历。或者,在另一个例子中,在跳转到新应用页面之后,先不在新应用页面进行遍历,而是记录需要在后续过程遍历新应用页面的信息,并跳转回当前应用页面,待当前应用页面的所有控件均遍历完成之后,才在已遍历列表内记录新应用页面的页面标识,并从新应用页面的第一个控件开始,在新应用页面进行遍历。

[0080] 例如,在从应用页面1跳转到应用页面2时,则当前应用页面为应用页面1,且新应用页面为应用页面2,在步骤202中,获取到的页面标识为应用页面2的页面标识。如表2或者表3所示,在已遍历列表内不存在该页面标识,因此,可以在已遍历列表内记录应用页面2的页面标识,并从应用页面2的第一个控件(控件B)开始,在应用页面2进行遍历,即依次遍历应用页面2下的控件B、控件B.1、控件B.1.1、控件B.1.2、控件B.1.2.1,假设在点击应用页面2下的控件B.1.2时,跳转回应用页面1,则可以在已遍历列表内记录应用页面2的最后访问控件(即控件B.1.2),如表5或者表6所示。

[0081] 表5

[0082]

页面标识	控件
应用页面1的页面标识	控件A.1.2
应用页面2的页面标识	控件B.1.2

[0083] 表6

[0084]

页面标识	控件
应用页面1的页面标识	控件A、控件A.1、控件A.1.1、控件A.1.2
应用页面2的页面标识	控件B、控件B.1、控件B.1.1、控件B.1.2

[0085] 又例如,在从应用页面2跳转到应用页面1时,则当前应用页面为应用页面2,且新应用页面为应用页面1,在步骤202中,获取到的页面标识为应用页面1的页面标识。如表2或者表3所示,在已遍历列表内已经存在该页面标识,因此,从已遍历列表内获取该页面标识对应的最后访问控件,即控件A.1.2。基于应用页面1内的各控件的顺序:控件A、控件A.1、控件A.1.1、控件A.1.2、控件A.1.2.1、控件A.1.2.2、控件A.1.3、控件A.2、控件A.3,从应用页面1的最后访问控件A.1.2的下一个控件A.1.2.1开始,在应用页面1进行遍历,且不再对控件A、控件A.1、控件A.1.1、控件A.1.2进行遍历,因此,只需要遍历控件A.1.2.1、控件A.1.2.2、控件A.1.3、控件A.2、控件A.3即可,从而避免重复遍历控件A、控件A.1、控件A.1.1、控件A.1.2。

[0086] 基于上述技术方案,本申请实施例中,在应用页面的遍历过程中,通过维护页面标识与最后访问控件的对应关系,在跳转到新应用页面后,如果之前已经访问过该新应用页面,就可以直接获取到该新应用页面对应的最后访问控件,继而可以使用最后访问控件在新应用页面进行遍历,而不需要从新应用页面的第一个控件开始,在新应用页面进行遍历,从而去除重复遍历的过程,节省了整个应用遍历的时间,为更高效、更稳定的遍历自动化测试带来可行性。

[0087] 针对上述情况二,在一个例子中,利用预设策略选取新应用页面内的控件的过程,可以包括但不限于如下方式:统计新应用页面内的每个控件的访问次数;利用每个控件的访问次数,选取所述新应用页面内的控件。

[0088] 其中,基于表4所示的已遍历列表,可以统计出每个控件的访问次数。

[0089] 其中,针对“利用每个控件的访问次数,选取所述新应用页面内的控件”的过程,可以选取访问次数少的控件为所述新应用页面内的控件。例如,选取访问次数最少的控件为所述新应用页面内的控件,当访问次数最少的控件为多个时,则可以从多个访问次数最少的控件中任意选取一个。又例如,设置一个访问次数阈值,且选取访问次数小于该次数阈值的控件为所述新应用页面内的控件,而不能选取访问次数大于等于该次数阈值的控件为所述新应用页面内的控件。又例如,基于每个控件的访问次数,为每个控件设置访问权重,访问次数越多的控件,其访问权重越小;而访问权重越小的控件,被选取为所述新应用页面内的控件的概率越小。上述过程给出了几个“利用每个控件的访问次数,选取所述新应用页面内的控件”的方式,实际应用中还可以采用其它方式,只要可以选取访问次数少的控件为所述新应用页面内的控件即可。

[0090] 本申请实施例中,基于该控件(即该选取的控件),在该新应用页面进行遍历的过

程,具体可以包括但不限于如下方式:基于新应用页面内的各控件的顺序,从新应用页面的该控件开始,在新应用页面进行遍历,且不再对该控件前面的所有控件进行遍历,本申请实施例中对此控件遍历方式不再赘述。

[0091] 本申请实施例中,在判断已遍历列表内是否存在该页面标识之后,如果判断结果为否,则在一个例子中,在已遍历列表内记录新应用页面的页面标识,并从新应用页面的第一个控件开始,在新应用页面进行遍历。或者,在另一个例子中,在跳转到新应用页面之后,先不在新应用页面进行遍历,而是记录需要在后续过程遍历新应用页面的信息,并跳转回当前应用页面,待当前应用页面的所有控件均遍历完成之后,才在已遍历列表内记录新应用页面的页面标识,并从新应用页面的第一个控件开始,在新应用页面进行遍历。

[0092] 例如,在从应用页面1跳转到应用页面2时,则当前应用页面为应用页面1,且新应用页面为应用页面2,在步骤202中,获取到的页面标识为应用页面2的页面标识。如表4所示,在已遍历列表内不存在该页面标识,因此,可以在已遍历列表内记录应用页面2的页面标识,并从应用页面2的第一个控件(控件B)开始,在应用页面2进行遍历,即依次遍历应用页面2下的控件B、控件B.1、控件B.1.1、控件B.1.2、控件B.1.2.1,假设在点击应用页面2下的控件B.1.2时,跳转回应用页面1,则可以在已遍历列表内记录应用页面2的页面标识,以及每个控件的访问次数,如表7所示。

[0093] 表7

[0094]

页面标识	控件
应用页面 1 的页面标识	控件 A: 3 次、控件 A.1: 3 次、控件 A.1.1: 2 次、控件 A.1.2: 2 次、控件 A.1.2.1: 2 次、控件 A.1.2.2: 1 次、控件 A.1.3: 1 次、控件 A.2: 0 次、控件 A.3: 0 次
应用页面 2 的页面标识	控件 B: 1 次、控件 B.1: 1 次、控件 B.1.1: 1 次、控件 B.1.2: 1 次、控件 B.1.2.1: 0 次

[0095] 又例如,在从应用页面2跳转到应用页面1时,则当前应用页面为应用页面2,且新应用页面为应用页面1,在步骤202中,获取到的页面标识为应用页面1的页面标识。如表4所示,在已遍历列表内已经存在该页面标识,因此,统计每个控件的访问次数,且发现控件A.2的访问次数为0,控件A.3的访问次数为0,因此,可以选取控件A.2或者控件A.3,假设选取控件A.2。基于应用页面1内的各控件的顺序:控件A、控件A.1、控件A.1.1、控件A.1.2、控件A.1.2.1、控件A.1.2.2、控件A.1.3、控件A.2、控件A.3,从控件A.2开始,在应用页面1进行遍历,且不再对控件A.2之前的控件进行遍历,因此,只需要遍历控件A.2和控件A.3即可,从而避免重复遍历控件A.2之前的各控件。

[0096] 基于上述技术方案,本申请实施例中,在应用页面的遍历过程中,通过维护页面标识与每个控件的访问次数的对应关系,在跳转到新应用页面后,如果之前已经访问过该新应用页面,就可以基于每个控件的访问次数选取该应用页面内的控件进行遍历,而不需要从新应用页面的第一个控件开始,在新应用页面进行遍历,从而去除重复遍历的过程,节省

了整个应用遍历的时间,为更高效、更稳定的遍历自动化测试带来可行性。

[0097] 本申请实施例中,在确定跳转到与当前应用页面不同的新应用页面之后,还可以通过拍照截屏方式或者页面标题判断方式分析新应用页面的准确性;如果分析出新应用页面准确,则执行上述步骤202-步骤205等过程;如果分析出新应用页面不准确,则直接跳转回当前应用页面,在当前应用页面继续遍历。

[0098] 其中,拍照截屏方式是指:在确定跳转到新应用页面之后,可以通过对新应用页面进行拍照或者截屏,分析出新应用页面是否准确。例如,如果新应用页面存在无法加载、无法打开等问题,则分析出新应用页面不准确。

[0099] 其中,页面标题判断方式是指:在确定跳转到新应用页面之后,可以通过对新应用页面的页面标题进行判断,分析出新应用页面是否准确。例如,若页面标题存在不完整、不准确、不符合要求等问题,分析出新应用页面不准确。

[0100] 本申请实施例中,在对新应用页面或者当前应用页面进行遍历的过程中,如果在遍历过程中出现异常弹框,则可以关闭异常弹框,并继续在新应用页面或者当前应用页面进行遍历,以防止异常弹框导致的错误跳出。其中,异常弹框可以是应用内部的异常alert(报警)或者外部系统级别的异常alert等。

[0101] 在从一个应用页面(如应用页面1)跳转到另一个应用页面(如应用页面2)时,需要先执行建立树结构的过程,而建立树结构的过程通常比较耗时。本申请实施例中,在建立树结构后,在从应用页面1跳转到应用页面2时,不会删除该树结构,而是将该树结构存储在本地存储介质中。当后续在跳回到该应用页面1时,就可以直接从本地存储介质中得到应用页面1的树结构,而不再需要建立应用页面1的树结构,从而该可以省去树结构的建立时间。

[0102] 基于与上述方法同样的申请构思,本申请实施例还提供一种应用页面的遍历装置120,该应用页面的遍历装置120应用在移动终端10上。该应用页面的遍历装置120可以通过软件实现,也可以通过硬件或者软硬件结合的方式实现。以软件实现为例,作为一个逻辑意义上的装置,是通过其所在的移动终端10的处理器11,读取非易失性存储器12中对应的计算机程序指令形成的。从硬件层面而言,如图4所示,为本申请提出的应用页面的遍历装置120所在的移动终端10的一种硬件结构图,除图4所示的处理器11、非易失性存储器12外,移动终端10还可以包括其他硬件,如负责处理报文的转发芯片、网络接口、内存等;从硬件结构上来讲,移动终端10还可能是分布式设备,可能包括多个接口卡,以便在硬件层面进行报文处理的扩展。

[0103] 如图5所示,为本申请提出的应用页面的遍历装置的结构图,该装置包括:

[0104] 确定模块121,用于确定跳转到与当前应用页面不同的新应用页面;

[0105] 第一获取模块122,用于获取所述新应用页面的页面标识;

[0106] 判断模块123,用于判断已遍历列表内是否存在所述页面标识;

[0107] 第二获取模块124,用于当判断结果为存在所述页面标识时,则利用预设策略选取所述新应用页面内的控件;

[0108] 处理模块125,用于基于所述控件,在所述新应用页面进行遍历。

[0109] 所述确定模块121,具体用于在确定跳转到与当前应用页面不同的新应用页面的过程中,如果在当前应用页面发生页面跳转,则比较跳转后页面的控件数量与当前应用页面的控件数量,如果二者不同,则确定跳转到与当前应用页面不同的新应用页面;或者,如

果在当前应用页面发生页面跳转,则比较跳转后页面的控件树形结构与当前应用页面的控件树形结构,如果二者不完全相同,则确定跳转到与当前应用页面不同的新应用页面。

[0110] 所述第二获取模块124,具体用于在利用预设策略选取所述新应用页面内的控件的过程中,从所述已遍历列表内选取所述页面标识对应的最后访问控件;或者,统计所述新应用页面内的每个控件的访问次数;利用每个控件的访问次数,选取所述新应用页面内的控件。

[0111] 所述第一获取模块122,还用于在确定模块确定跳转到与当前应用页面不同的新应用页面后,获取当前应用页面的最后访问控件,在所述已遍历列表内记录当前应用页面的页面标识与所述当前应用页面的最后访问控件的对应关系。

[0112] 在一个例子中,所述第一获取模块122获取的所述新应用页面或者当前应用页面的页面标识包括:所述新应用页面或者当前应用页面的页面路径信息,或者基于页面内的控件确定的唯一标识。

[0113] 进一步的,所述第一获取模块122在基于页面内的控件确定所述新应用页面或者当前应用页面的页面标识的过程中,将所述新应用页面或者所述当前应用页面内的控件树形结构转换为一个字符串,并对所述字符串进行运算,并从运算结果中选取指定位数的数值,并将所述指定位数的数值确定为所述新应用页面或者所述当前应用页面的页面标识。

[0114] 在一个例子中,所述处理模块125,还用于在所述判断模块123判断已遍历列表内是否存在所述页面标识之后,当判断结果为不存在所述页面标识时,则在所述已遍历列表内记录所述新应用页面的页面标识,并从所述新应用页面的第一个控件开始,在所述新应用页面进行遍历。

[0115] 所述处理模块125,具体用于在基于所述控件,在所述新应用页面进行遍历的过程中,基于所述新应用页面内的各控件的顺序,从所述新应用页面的所述控件的下一个控件开始,在所述新应用页面进行遍历,且不再对所述控件以及所述控件前面的所有控件进行遍历。

[0116] 在一个例子中,所述处理模块125,还用于在所述确定模块121确定跳转到与当前应用页面不同的新应用页面之后,通过拍照截屏方式或者页面标题判断方式分析所述新应用页面的准确性;如果分析出所述新应用页面准确,则通知所述第一获取模块执行获取所述新应用页面的页面标识的过程;如果分析出所述新应用页面不准确,则跳转回所述当前应用页面。

[0117] 在一个例子中,所述处理模块125,还用于在所述新应用页面进行遍历的过程中,如果在遍历过程中出现异常弹框,则关闭所述异常弹框,并继续在所述新应用页面进行遍历。

[0118] 其中,本申请装置的各个模块可以集成于一体,也可以分离部署。上述模块可以合并为一个模块,也可以进一步拆分成多个子模块。

[0119] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到本申请可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本申

请各个实施例所述的方法。本领域技术人员可以理解附图只是一个优选实施例的示意图，附图中的模块或流程并不一定是实施本申请所必须的。

[0120] 本领域技术人员可以理解实施例中的装置中的模块可以按照实施例描述进行分布于实施例的装置中，也可以进行相应变化位于不同于本实施例的一个或多个装置中。上述实施例的模块可以合并为一个模块，也可进一步拆分成多个子模块。上述本申请实施例序号仅仅为了描述，不代表实施例的优劣。

[0121] 以上公开的仅为本申请的几个具体实施例，但是，本申请并非局限于此，任何本领域的技术人员能思之的变化都应落入本申请的保护范围。



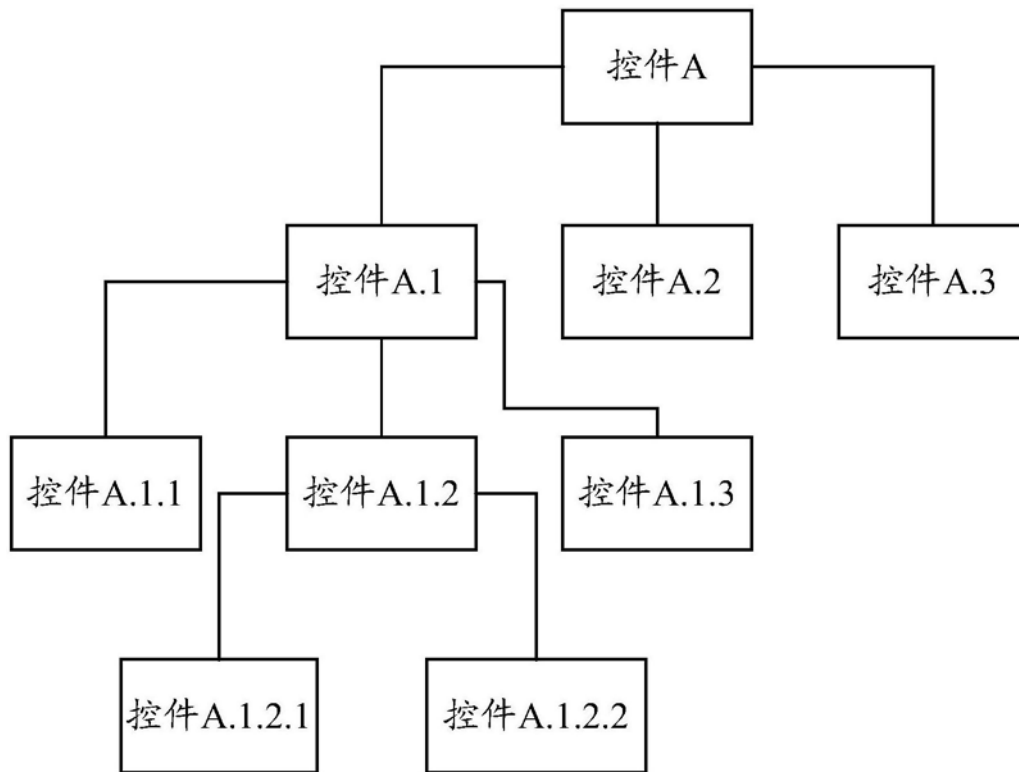


图1

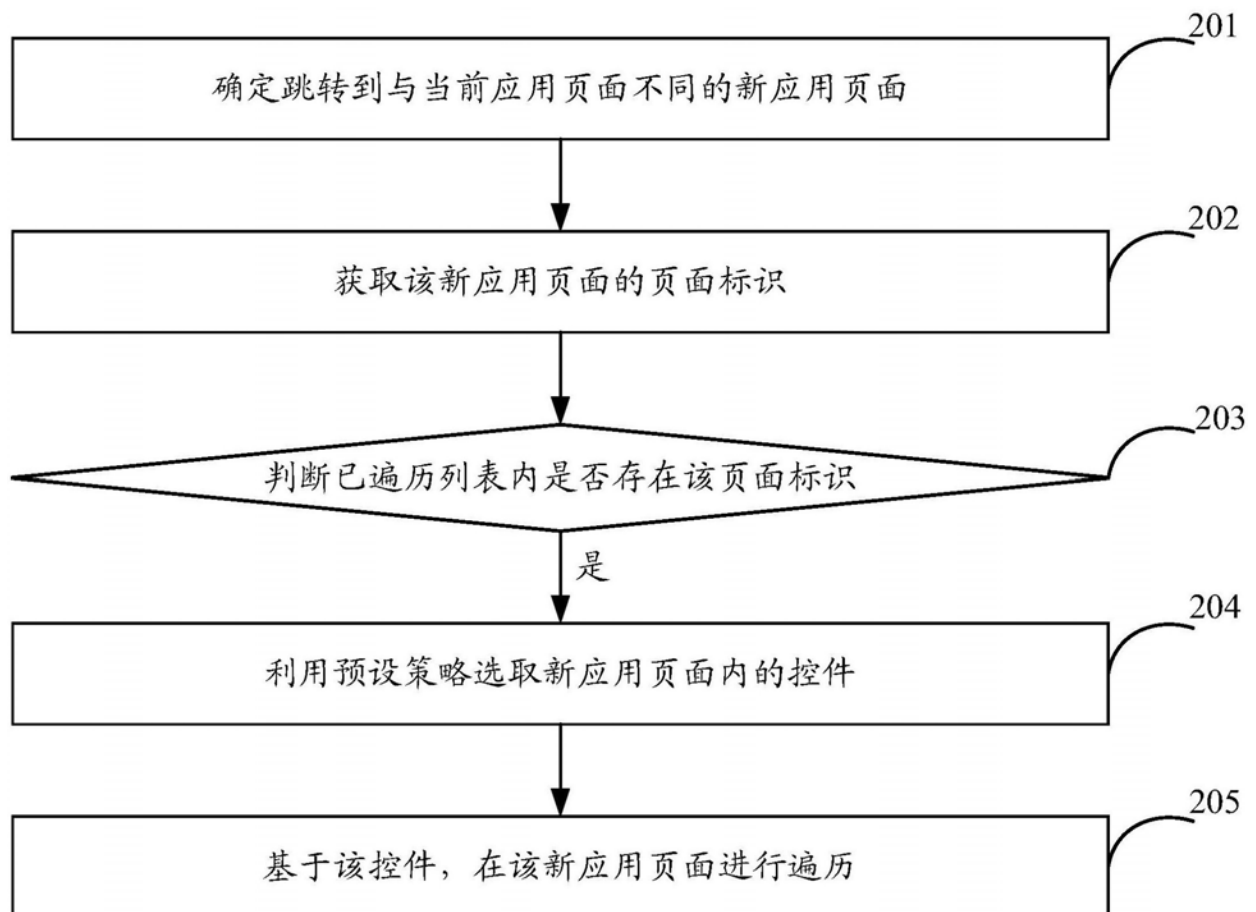


图2

```
+ - UIATarget: name:iPhone Simulator rect:{{0,0},{320,480}}
| +- UIAApplication: name:TestAutomation rect:{{0,20},{320,460}}
| | +- UIWindow: rect:{{0,0},{320,480}}
| | | +- UIAStaticText: name:First View value:First View rect:{{54,52},{212,43}}
| | | +- UITextField: name:User Text value:Tap Some Text Here | rect:{{20,179},{280,31}}
| | | +- UIAStaticText: name:The text is: value:The text is: rect:{{20,231},{112,21}}
| | | +- UIAStaticText: value: rect:{{145,231},{155,21}}
| | | +- UITabBar: rect:{{0,431},{320,49}}
| | | | +- UIImage: rect:{{0,431},{320,49}}
| | | | +- UIButton: name:First value:1 rect:{{2,432},{156,48}}
| | | | +- UIButton: name:Second rect:{{162,432},{156,48}}
```

图3

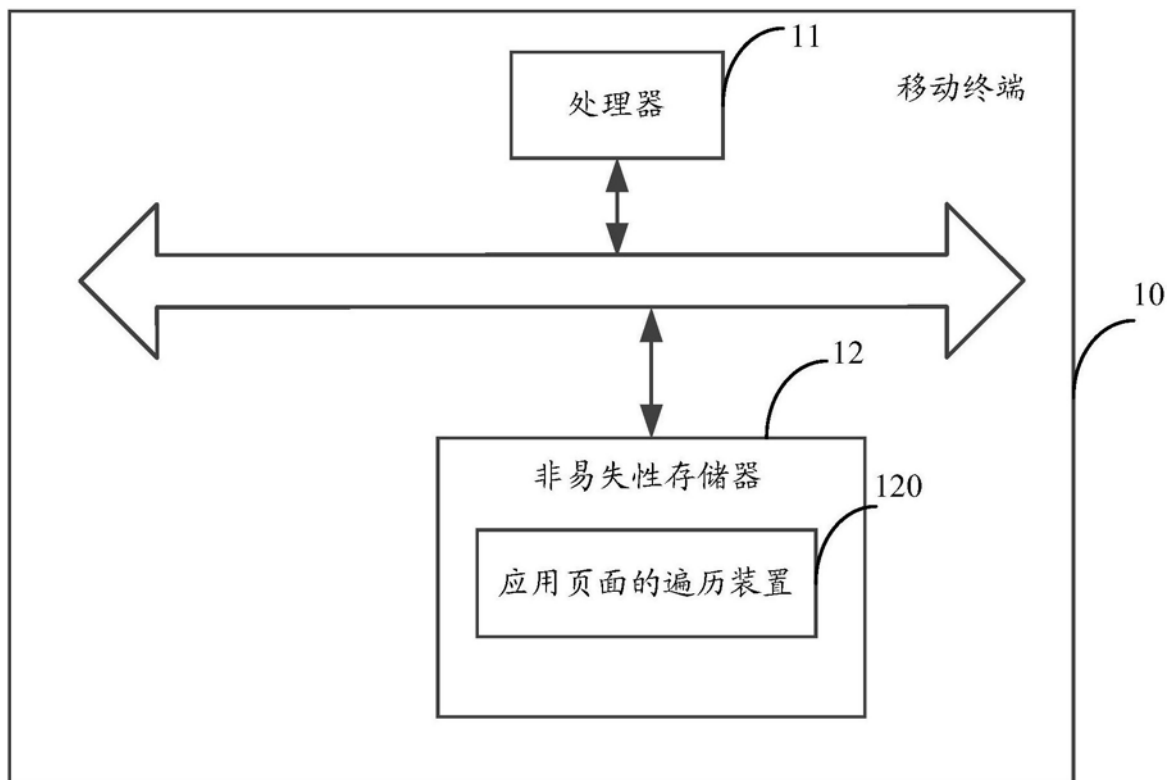


图4

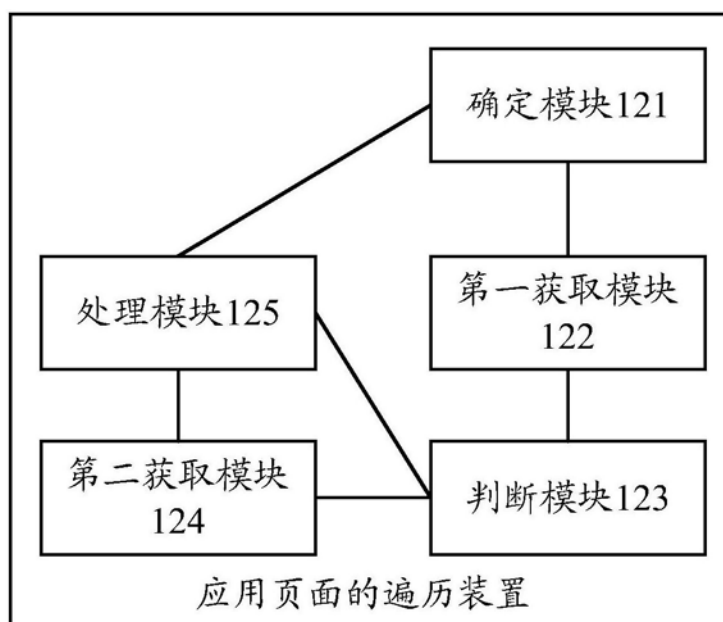


图5