



(12) 发明专利

(10) 授权公告号 CN 108595329 B

(45) 授权公告日 2022. 08. 26

(21) 申请号 201810367819.X

审查员 陈治训

(22) 申请日 2018.04.23

(65) 同一申请的已公布的文献号
申请公布号 CN 108595329 A

(43) 申请公布日 2018.09.28

(73) 专利权人 腾讯科技(深圳)有限公司
地址 518057 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 高源

(74) 专利代理机构 深圳翼盛智成知识产权事务
所(普通合伙) 44300
专利代理师 李玉婷

(51) Int. Cl.
G06F 11/36 (2006.01)

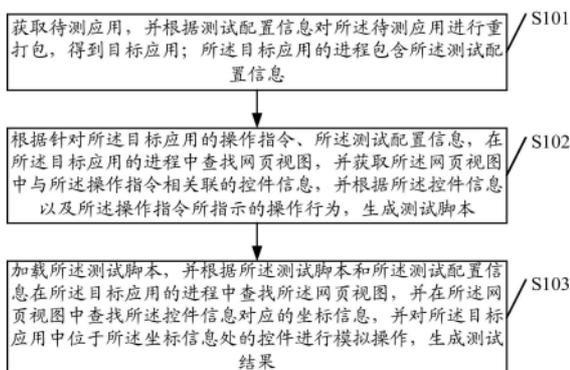
权利要求书3页 说明书21页 附图8页

(54) 发明名称

一种应用测试方法、装置及计算机存储介质

(57) 摘要

本发明实施例公开了一种应用测试方法、装置及计算机存储介质,其中方法包括:获取待测应用,并根据测试包数据对待测应用进行重打包,得到目标应用;目标应用的进程包含测试包数据;根据针对目标应用的操作指令、测试包数据,在目标应用的进程中查找网页视图,并获取网页视图中与操作指令相关联的控件信息,并根据控件信息以及操作指令所指示的操作行为,生成测试脚本;加载测试脚本,并根据测试脚本和测试包数据在目标应用的进程中查找网页视图,并在网页视图中查找控件信息对应的坐标信息,并对目标应用中位于坐标信息处的控件进行模拟操作,生成测试结果。采用本发明,可以对基于自定义的webview内核的移动端网页程序进行自动化测试。



1. 一种应用测试方法,其特征在于,包括:

获取待测应用,并根据测试包数据对所述待测应用进行重打包,得到目标应用;所述目标应用的进程包含所述测试包数据;

根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本;其中,该步骤包括:查找所述操作指令所指示的坐标信息是否有原生webview或X5 webview;若有,则注入查询代码段,并基于所述查询代码段查找所述坐标信息对应的控件,并获取所述控件的控件信息;通过回调函数将所述控件信息以及所述操作指令所指示的操作行为返回测试应用,并在所述测试应用中所述控件信息以及所述操作行为转化为所述测试脚本;

加载所述测试脚本,并根据所述测试脚本和所述测试包数据在所述目标应用的进程中查找所述网页视图,并在所述网页视图中查找所述控件信息对应的坐标信息,并对所述目标应用中位于所述坐标信息处的控件进行模拟操作,生成测试结果,其中,该步骤包括:生成多条测试指令,其中,每条所述测试指令均有自身对应的操作行为、以及操作行为针对的操作对象;查找所述测试指令所包含的操作对象是否有原生webview或X5 webview;若有,则注入查询代码段,并通过所述查询代码段查找所述测试指令中的控件信息对应的控件;获取所述控件在网页页面的坐标信息;基于所述测试指令对应的操作行为,对所述坐标信息处的控件进行模拟操作,生成测试结果。

2. 如权利要求1所述的方法,其特征在于,还包括:

启动所述目标应用,并初始化所述目标应用中的所述测试包数据,并为所述目标应用的进程和测试应用的进程创建进程通信服务;所述测试应用的进程用于加载所述测试脚本。

3. 如权利要求2所述的方法,其特征在于,所述根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本,包括:

根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过查询代码段对所述网页视图的页面内容进行遍历,得到遍历结果;

获取所述操作指令所指示的操作位置的坐标信息,并从所述遍历结果中获取所述坐标信息对应的控件信息;

根据所述目标应用中的回调函数将所述控件信息返回至所述测试应用,并在所述测试应用中所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

4. 如权利要求2所述的方法,其特征在于,所述根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本,包括:

根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过loadURL从所述网页视图的页面内容中,获取所述操作指令所指示的操作位置的控件信息;

拦截由loadURL获取到的所述控件信息,并通过编码语言注册通道将拦截到的所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

5.如权利要求2所述的方法,其特征在于,所述加载所述测试脚本,并根据所述测试脚本和所述测试包数据在所述目标应用的进程中查找所述网页视图,并在所述网页视图中查找所述控件信息对应的坐标信息,并对所述目标应用中位于所述坐标信息处的控件进行模拟操作,生成测试结果,包括:

在所述测试应用的进程中加载所述测试脚本,并通过所述进程通信服务将所述测试脚本中的测试指令发送到所述目标应用的进程;

根据所述测试包数据和所述测试指令,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

若根据所述视图结构数组在所述测试指令所指示的待操作对象中查找到网页视图,则通过查询代码段在所述网页视图中,查找所述测试脚本中的所述控件信息所对应的控件,并获取所述控件的坐标信息;

通过回调函数将所述控件的坐标信息返回至系统测试框架;

基于所述系统测试框架,对所述目标应用中位于所述坐标信息处的所述控件进行模拟操作,生成测试结果。

6.如权利要求1所述的方法,其特征在于,还包括:

获取检测指令,并根据所述检测指令设置自定义网页回调接口服务;

若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测和记录所述目标应用的界面中的网页页面检测信息;

将所述网页页面检测信息添加至所述测试结果,并对添加后的测试结果进行显示。

7.如权利要求6所述的方法,其特征在于,所述获取检测指令,并根据所述检测指令设置自定义网页回调接口服务,包括:

获取检测指令;

若所述网页视图包含网页回调接口服务,则根据所述检测指令获取所述网页回调接口服务,并对所述网页回调接口服务中的功能函数进行重写,得到自定义网页回调接口服务;

若所述网页视图未包含网页回调接口服务,则创建所述自定义网页回调接口服务。

8.如权利要求6所述的方法,其特征在于,所述若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测和记录所述目标应用的界面中的网页页面检测信息,包括:

若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测所述目标应用中的网页页面;

若检测到所述网页页面处于开始加载状态,则通过所述自定义网页回调接口服务中的

第一功能函数记录网页加载起始时间信息；

若检测到所述网页页面处于结束加载状态，则通过所述自定义网页回调接口服务中的第二功能函数记录网页加载结束时间信息；

若检测到所述网页页面处于页面跳转状态，则通过所述自定义网页回调接口服务中的第三功能函数记录页面跳转信息；

若检测到所述网页页面处于页面加载异常状态，则通过所述自定义网页回调接口服务中的第四功能函数记录页面加载异常信息；

若检测到所述网页页面处于资源请求状态，则通过所述自定义网页回调接口服务中的第五功能函数统计资源加载数量；

将所述网页加载起始时间信息、网页加载结束时间信息、页面跳转信息、页面加载异常信息、资源加载数量，确定为所述网页页面检测信息。

9. 一种应用测试装置，其特征在于，包括：

重打包模块，用于获取待测应用，并根据测试包数据对所述待测应用进行重打包，得到目标应用；所述目标应用的进程包含所述测试包数据；

脚本生成模块，用于根据针对所述目标应用的操作指令、所述测试包数据，在所述目标应用的进程中查找网页视图，并获取所述网页视图中与所述操作指令相关联的控件信息，并根据所述控件信息以及所述操作指令所指示的操作行为，生成测试脚本；其中，所述脚本生成模块，具体用于查找所述操作指令所指示的坐标信息是否有原生webview或X5 webview；若有，则注入查询代码段，并基于所述查询代码段查找所述坐标信息对应的控件，并获取所述控件的控件信息；通过回调函数将所述控件信息以及所述操作指令所指示的操作行为返回测试应用，并在所述测试应用中将所述控件信息以及所述操作行为转化为所述测试脚本；

测试模块，用于加载所述测试脚本，并根据所述测试脚本和所述测试包数据在所述目标应用的进程中查找所述网页视图，并在所述网页视图中查找所述控件信息对应的坐标信息，并对所述目标应用中位于所述坐标信息处的控件进行模拟操作，生成测试结果，其中，所述测试模块，具体用于生成多条测试指令，其中，每条所述测试指令均有自身对应的操作行为、以及操作行为针对的操作对象；查找所述测试指令所包含的操作对象是否有原生webview或X5 webview；若有，则注入查询代码段，并通过所述查询代码段查找所述测试指令中的控件信息对应的控件；获取所述控件在网页页面的坐标信息；基于所述测试指令对应的操作行为，对所述坐标信息处的控件进行模拟操作，生成测试结果。

10. 一种终端设备，其特征在于，包括：处理器和存储器；

所述处理器和存储器相连，其中，所述存储器用于存储程序代码，所述处理器用于调用所述程序代码，以执行如权利要求1-8任一项所述的方法。

11. 一种计算机存储介质，其特征在于，所述计算机存储介质存储有计算机程序，所述计算机程序包括程序指令，所述程序指令当被处理器执行时，执行如权利要求1-8任一项所述的方法。

一种应用测试方法、装置及计算机存储介质

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种应用测试方法、装置及计算机存储介质。

背景技术

[0002] 目前所开发的应用程序的数量是越来越多,为了保证应用程序的性能和质量,通常都需要对应用程序进行测试,而测试的内容通常都比较多且比较苛刻,若用人工进行一步步的测试,则需要非常高的人工成本,因此,自动化测试则显得尤为重要。

[0003] 目前还可以对移动端网页程序进行自动化测试,目前的主流测试方式可以为:通过Uiautomator (是谷歌推出的用于用户界面自动化测试的工具)提供的API (Application Programming Interface,应用程序编程接口)可以获得移动端网页程序中原生webview (网页视图,用于网页的控件)里的页面内容,进而根据获得的页面内容实现自动化测试。但是目前也存在许多的移动端网页程序是基于自定义的webview内核(如腾讯开发的X5 webview)实现的,此时,是无法通过uiautomator在自定义的webview里抓取到有效的页面元素,进而导致无法对基于自定义的webview内核的移动端网页程序进行自动化测试。

发明内容

[0004] 本发明实施例提供一种应用测试方法、装置及计算机存储介质,可以对基于自定义的webview内核的移动端网页程序进行自动化测试。

[0005] 本发明实施例一方面提供了一种应用测试方法,包括:

[0006] 获取待测应用,并根据测试包数据对所述待测应用进行重打包,得到目标应用;所述目标应用的进程包含所述测试包数据;

[0007] 根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本;

[0008] 加载所述测试脚本,并根据所述测试脚本和所述测试包数据在所述目标应用的进程中查找所述网页视图,并在所述网页视图中查找所述控件信息对应的坐标信息,并对所述目标应用中位于所述坐标信息处的控件进行模拟操作,生成测试结果。

[0009] 其中,还包括:

[0010] 启动所述目标应用,并初始化所述目标应用中的所述测试包数据,并为所述目标应用的进程和测试应用的进程创建进程通信服务;所述测试应用的进程用于加载所述测试脚本。

[0011] 其中,所述根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本,包括:

[0012] 根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的

视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0013] 接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过查询代码段对所述网页视图的页面内容进行遍历,得到遍历结果;

[0014] 获取所述操作指令所指示的操作位置的坐标信息,并从所述遍历结果中获取所述坐标信息对应的控件信息;

[0015] 根据所述目标应用中的回调函数将所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

[0016] 其中,所述根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本,包括:

[0017] 根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0018] 接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过loadURL从所述网页视图的页面内容中,获取所述操作指令所指示的操作位置的控件信息;

[0019] 拦截由loadURL获取到的所述控件信息,并通过编码语言注册通道将拦截到的所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

[0020] 其中,所述加载所述测试脚本,并根据所述测试脚本和所述测试包数据在所述目标应用的进程中查找所述网页视图,并在所述网页视图中查找所述控件信息对应的坐标信息,并对所述目标应用中位于所述坐标信息处的控件进行模拟操作,生成测试结果,包括:

[0021] 在所述测试应用的进程中加载所述测试脚本,并通过所述进程通信服务将所述测试脚本中的测试指令发送到所述目标应用的进程;

[0022] 根据所述测试包数据和所述测试指令,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0023] 若根据所述视图结构数组在所述测试指令所指示的待操作对象中查找到网页视图,则通过查询代码段在所述网页视图中,查找所述测试脚本中的所述控件信息所对应的控件,并获取所述控件的坐标信息;

[0024] 通过回调函数将所述控件的坐标信息返回至系统测试框架;

[0025] 基于所述系统测试框架,对所述目标应用中位于所述坐标信息处的所述控件进行模拟操作,生成测试结果。

[0026] 其中,还包括:

[0027] 获取检测指令,并根据所述检测指令设置自定义网页回调接口服务;

[0028] 若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测和记录所述目标应用的界面中的网页页面检测信息;

[0029] 将所述网页页面检测信息添加至所述测试结果,并对添加后的测试结果进行显示。

[0030] 其中,所述获取检测指令,并根据所述检测指令设置自定义网页回调接口服务,包

括：

[0031] 获取检测指令；

[0032] 若所述网页视图包含网页回调接口服务，则根据所述检测指令获取所述网页回调接口服务，并对所述网页回调接口服务中的功能函数进行重写，得到自定义网页回调接口服务；

[0033] 若所述网页视图未包含网页回调接口服务，则创建所述自定义网页回调接口服务。

[0034] 其中，所述若检测到所述目标应用的界面包含所述网页视图，则触发所述自定义网页回调接口服务检测和记录所述目标应用的界面中的网页页面检测信息，包括：

[0035] 若检测到所述目标应用的界面包含所述网页视图，则触发所述自定义网页回调接口服务检测所述目标应用中的网页页面；

[0036] 若检测到所述网页页面处于开始加载状态，则通过所述自定义网页回调接口服务中的第一功能函数记录网页加载起始时间信息；

[0037] 若检测到所述网页页面处于结束加载状态，则通过所述自定义网页回调接口服务中的第二功能函数记录网页加载结束时间信息；

[0038] 若检测到所述网页页面处于页面跳转状态，则通过所述自定义网页回调接口服务中的第三功能函数记录页面跳转信息；

[0039] 若检测到所述网页页面处于页面加载异常状态，则通过所述自定义网页回调接口服务中的第四功能函数记录页面加载异常信息；

[0040] 若检测到所述网页页面处于资源请求状态，则通过所述自定义网页回调接口服务中的第五功能函数统计资源加载数量；

[0041] 将所述网页加载起始时间信息、网页加载结束时间信息、页面跳转信息、页面加载异常信息、资源加载数量，确定为所述网页页面检测信息。

[0042] 本发明实施例一方面提供了一种应用测试装置，包括：

[0043] 重打包模块，用于获取待测应用，并根据测试包数据对所述待测应用进行重打包，得到目标应用；所述目标应用的进程包含所述测试包数据；

[0044] 脚本生成模块，用于根据针对所述目标应用的操作指令、所述测试包数据，在所述目标应用的进程中查找网页视图，并获取所述网页视图中与所述操作指令相关联的控件信息，并根据所述控件信息以及所述操作指令所指示的操作行为，生成测试脚本；

[0045] 测试模块，用于加载所述测试脚本，并根据所述测试脚本和所述测试包数据在所述目标应用的进程中查找所述网页视图，并在所述网页视图中查找所述控件信息对应的坐标信息，并对所述目标应用中位于所述坐标信息处的控件进行模拟操作，生成测试结果。

[0046] 其中，还包括：

[0047] 初始化模块，用于启动所述目标应用，并初始化所述目标应用中的所述测试包数据，并为所述目标应用的进程和测试应用的进程创建进程通信服务；所述测试应用的进程用于加载所述测试脚本。

[0048] 其中，所述脚本生成模块包括：

[0049] 第一数组生成单元，用于根据所述测试包数据，在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树，并对所述视图结构树进行遍历，生成视图结构数组；

[0050] 页面遍历单元,用于接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过查询代码段对所述网页视图的页面内容进行遍历,得到遍历结果;

[0051] 第一控件获取单元,用于获取所述操作指令所指示的操作位置的坐标信息,并从所述遍历结果中获取所述坐标信息对应的控件信息;

[0052] 第一脚本转化单元,用于根据所述目标应用中的回调函数将所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

[0053] 其中,所述脚本生成模块包括:

[0054] 第二数组生成单元,用于根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0055] 第二控件获取单元,用于接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过loadURL从所述网页视图的页面内容中,获取所述操作指令所指示的操作位置的控件信息;

[0056] 第二脚本转化单元,用于拦截由loadURL获取到的所述控件信息,并通过编码语言注册通道将拦截到的所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

[0057] 其中,所述测试模块包括:

[0058] 指令发送单元,用于在所述测试应用的进程中加载所述测试脚本,并通过所述进程通信服务将所述测试脚本中的测试指令发送到所述目标应用的进程;

[0059] 第三数组生成单元,用于根据所述测试包数据和所述测试指令,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0060] 坐标获取单元,用于若根据所述视图结构数组在所述测试指令所指示的待操作对象中查找到网页视图,则通过查询代码段在所述网页视图中,查找所述测试脚本中的所述控件信息所对应的控件,并获取所述控件的坐标信息;

[0061] 坐标发送单元,用于通过回调函数将所述控件的坐标信息返回至系统测试框架;

[0062] 模拟操作单元,用于基于所述系统测试框架,对所述目标应用中位于所述坐标信息处的所述控件进行模拟操作,生成测试结果。

[0063] 其中,还包括:

[0064] 设置模块,用于获取检测指令,并根据所述检测指令设置自定义网页回调接口服务;

[0065] 检测模块,用于若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测和记录所述目标应用的界面中的网页页面检测信息;

[0066] 显示模块,用于将所述网页页面检测信息添加至所述测试结果,并对添加后的测试结果进行显示。

[0067] 其中,所述设置模块包括:

[0068] 指令获取单元,用于获取检测指令;

[0069] 函数重写单元,用于若所述网页视图包含网页回调接口服务,则根据所述检测指

令获取所述网页回调接口服务,并对所述网页回调接口服务中的功能函数进行重写,得到自定义网页回调接口服务;

[0070] 创建单元,用于若所述网页视图未包含网页回调接口服务,则创建所述自定义网页回调接口服务。

[0071] 其中,所述检测模块包括:

[0072] 触发单元,用于若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测所述目标应用中的网页页面;

[0073] 检测信息记录单元,用于若检测到所述网页页面处于开始加载状态,则通过所述自定义网页回调接口服务中的第一功能函数记录网页加载起始时间信息;

[0074] 所述检测信息记录单元,还用于若检测到所述网页页面处于结束加载状态,则通过所述自定义网页回调接口服务中的第二功能函数记录网页加载结束时间信息;

[0075] 所述检测信息记录单元,还用于若检测到所述网页页面处于页面跳转状态,则通过所述自定义网页回调接口服务中的第三功能函数记录页面跳转信息;

[0076] 所述检测信息记录单元,还用于若检测到所述网页页面处于页面加载异常状态,则通过所述自定义网页回调接口服务中的第四功能函数记录页面加载异常信息;

[0077] 所述检测信息记录单元,还用于若检测到所述网页页面处于资源请求状态,则通过所述自定义网页回调接口服务中的第五功能函数统计资源加载数量;

[0078] 确定单元,用于将所述网页加载起始时间信息、网页加载结束时间信息、页面跳转信息、页面加载异常信息、资源加载数量,确定为所述网页页面检测信息。

[0079] 本发明实施例一方面提供了一种终端设备,包括:处理器和存储器;

[0080] 所述处理器和存储器相连,其中,所述存储器用于存储程序代码,所述处理器用于调用所述程序代码,以执行如本发明实施例中一方面中的方法。

[0081] 本发明实施例一方面提供了一种计算机存储介质,所述计算机存储介质存储有计算机程序,所述计算机程序包括程序指令,所述程序指令当被处理器执行时,执行如本发明实施例中一方面中的方法。

[0082] 本发明实施例通过获取待测应用,并根据测试包数据对待测应用进行重打包,得到目标应用;目标应用的进程包含测试包数据;根据针对目标应用的操作指令、测试包数据,在目标应用的进程中获取网页视图,并获取网页视图中与操作指令相关联的控件信息,并根据控件信息以及操作指令所指示的操作行为,生成测试脚本;加载测试脚本,并根据测试脚本和测试包数据在目标应用的进程中获取网页视图,并在网页视图中查找控件信息对应的坐标信息,并对目标应用中位于坐标信息处的控件进行模拟操作,生成测试结果。可见通过重打包的机制,可以使得目标应用的进程包含测试包数据,进而可以基于测试包数据实现对原生的网页视图或自定义的网页视图(如X5 webview)中的控件信息的抓取和事件模拟,从而可以实现测试脚本的录制和自动化测试,整套方案不但可以适配基于X5 webview的微信小程序,对于所有X5内核的应用都可以适用,即本发明实施例可以对基于原生的网页视图或基于自定义的网页视图的应用进行自动化测试。

附图说明

[0083] 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简单地介绍,显而易见地,下面描述中的附图仅仅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的前提下,还可以根据这些附图获得其他的附图。

- [0084] 图1是本发明实施例提供的一种应用测试方法的流程示意图;
- [0085] 图2是本发明实施例提供的一种测试应用的界面示意图;
- [0086] 图3是本发明实施例提供的一种测试应用和目标应用之间的进程交互示意图;
- [0087] 图4是本发明实施例提供的一种测试脚本录制方法的流程示意图;
- [0088] 图5是本发明实施例提供的一种自动化测试方法的流程示意图;
- [0089] 图6是本发明实施例提供的一种页面检测方法的流程示意图;
- [0090] 图7是本发明实施例提供的一种应用测试装置的结构示意图;
- [0091] 图8是本发明实施例提供的一种终端设备的结构示意图。

具体实施方式

[0092] 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,所描述的实施例仅仅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。

[0093] 请参见图1,是本发明实施例提供的一种应用测试方法的流程示意图,所述方法可以包括:

[0094] S101,获取待测应用,并根据测试包数据对所述待测应用进行重打包,得到目标应用;所述目标应用的进程包含所述测试包数据;

[0095] 具体的,用户终端可以通过测试应用获取待测应用,例如,测试应用中可以展示应用列表,该应用列表可以包括多个应用名称,用户可以点击该应用列表中的某个应用名称,进而用户终端根据点击指令选取该应用名称对应的应用,作为待测应用。测试应用可以是指安装在用户终端中的用于对应用程序进行测试的应用。该测试应用可以包括测试服务,该测试服务可以提供测试包数据以及重打包功能,即测试应用选取待测应用后,可以通过测试包数据对所述待测应用进行重打包,即测试服务可以将测试包数据打包至待测应用中,再将包含测试包数据的待测应用称之为目标应用,测试包数据可以为dex文件。其中,测试包数据可以包括测试模块代码,该测试模块代码具有获取当前页面中的控件信息的功能,因此,目标应用通过运行该测试模块代码,可以使得目标应用具备获取当前页面中的控件信息的功能。

[0096] 对待测应用进行重打包后,可以启动目标应用,在启动过程中,还可以一并加载目标应用中的测试包数据,并对测试包数据进行初始化,该初始化的过程可以包括:初始化katapplication、初始化Art Hook.katapplication可以称为自定义的应用初始化类,在Android系统中,每个应用启动都会先执行应用初始化类,katapplication在这里的作用是在应用初始化时候将建立通信服务和注册接口拦截服务,即在该初始化的过程中可以基于katapplication为目标应用的进程和测试应用的进程创建进程通信服务,该进程通信服务可以为socketserver,socketserver可以提供socket通信,即通过socketserver可以实现两个不同应用进程之间的数据通信。该进程通信服务位于目标应用的进程中。其中,Art

Hook可称之为方法拦截,作用是在系统调用到某个方法时候,会被拦截到,从而在执行这个方法前后执行指定功能。因此,本发明实施例通过初始化Art Hook,可以在被测应用启动后基于Art Hook进行拦截,以开启socket通信服务。初始化katapplication、初始化Art Hook的过程均在目标应用的进程中实现。

[0097] S102,根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本;

[0098] 具体的,网页视图可以包括原生webview或X5 webview。在执行自动化测试之前,还需生成目标应用的测试脚本,该测试脚本用于后续对目标应用进行自动化测试。在用户终端中的测试应用的界面中可以显示用于录制测试脚本的录制按键,当用户点击该录制按键,用户终端可以根据点击指令,触发测试应用发送脚本录制命令,测试应用具体可以通过socket通信将脚本录制命令发送到目标应用,目标应用中的socketserver监听到该脚本录制命令后,可以基于测试包数据所提供的功能以及该脚本录制命令,开始记录用户对目标应用的所有操作行为(包括如对目标应用进行点击、滑动等操作),并可以记录每个操作行为中被操作的控件的信息,控件的信息可以包括控件的xpath(可扩展标记语言路径语言)信息。记录每个操作行为中被操作的控件的信息的具体过程可以为:

[0099] 基于目标应用中的测试包数据可以反射读取到当前activity的views树(由于测试包数据与目标应用处于同一进程中,所以基于测试包数据可以直接获取到views树,同样也可以获取到views树中的原生webview或X5 webview),再通过深度优先的遍历算法遍历整个views树,再存储在list数组中,然后根据list数组以及录制过程中针对目标应用的操作指令(如点击、滑动所产生的操作指令),【查找操作指令所指示的坐标信息是否有原生webview或X5 webview,如果有,则注入基于JavaScript(一种直译式脚本语言,是一种动态类型、弱类型、基于原型的语言,内置支持类型)的查询代码段,通过该查询代码段可以遍历移动端网页页面(如微信小程序中的网页页面)的document,通过操作指令所指示的坐标信息在document中查找符合条件的控件(即位于操作指令所指示的坐标信息的控件),并获取符合条件的该控件的控件信息,最后通过evaluateJavascript提供的回调函数将控件信息、所记录的所有操作行为返回至测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本】,即用户对目标应用的每一个操作所产生的操作行为信息和被操作的控件信息均可以被转化为测试脚本,直至用户结束对目标应用的操作,则完成整个脚本录制过程。其中,evaluateJavascript是Android系统提供的一个方法名,也可以称之为Javascript代码执行函数。

[0100] S103,加载所述测试脚本,并根据所述测试脚本和所述测试包数据在所述目标应用的进程中查找所述网页视图,并在所述网页视图中查找所述控件信息对应的坐标信息,并对所述目标应用中位于所述坐标信息处的控件进行模拟操作,生成测试结果。

[0101] 具体的,录制完测试脚本后,用户终端可以通过测试应用对测试脚本进行加载,测试应用可以在加载的过程中生成一条条测试指令,并逐一将每条测试指令通过socket通信发送到目标应用的进程通信服务中,进而可以基于测试包数据在目标应用中逐一执行每条测试指令所指示的内容。其中,发送测试指令的顺序与录制测试脚本过程中的每一个操作行为的顺序一致,且一个测试指令所指示的内容即为录制测试脚本过程中针对目标应用的

一条操作行为,且测试指令还可以包含录制过程中与操作指令相关联的控件信息。目标应用接收到测试指令后,即可基于测试包数据反射读取到测试指令所对应的activity的views树(由于测试包数据与目标应用处于同一进程中,所以基于测试包数据可以直接获取到views树,同样也可以获取到views树中的原生webview或X5 webview),再通过深度优先的遍历算法遍历整个views树,再存储在list数组中,然后根据测试指令和list数组,查找测试指令所包含的操作对象是否有原生webview或X5 webview,如果有,则注入基于JavaScript的查询代码段,由于测试指令中包含控件信息(如控件的xpath信息),所以无需再基于该查询代码段重新遍历移动端网页页面(即目标应用的网页页面)的document,只需通过该查询代码段查找测试指令中的控件信息所对应的控件,并获取该控件在网页页面中的坐标信息,再通过evaluateJavascript提供的回调函数将控件的坐标信息返回到instrumentation框架。在instrumentation框架中,基于测试指令中的操作行为,对获得的坐标信息处的控件进行模拟操作,如测试指令包含的操作行为是“对控件A进行点击操作”,则基于instrumentation框架可以对坐标信息处的控件A进行模拟点击,进而得到该模拟操作对应的输出结果,当完成测试脚本中的每一条测试指令,即完成了对目标应用的自动化测试,进而将每一条测试指令对应的输出结果确定为测试结果,测试人员即可将测试结果与预期结果进行对比,以分析出目标应用的性能。

[0102] 由于测试包数据处于目标应用的进程中,所以测试包数据所提供的测试功能可直接抓取目标应用中原生webview或X5 webview的页面内容,因此,就算目标应用使用的是自定义的X5 webview(如目标应用为微信中的小程序),也可以通过测试包数据所提供的测试功能直接抓取到目标应用的网页页面内容,因此,可以顺利地为目标应用进行测试脚本录制,也可以顺利地基于测试脚本对目标应用进行自动化测试。由于测试包数据处于目标应用的进程中,所以测试包数据所提供的测试功能也可以直接调用目标应用中原生webview或X5webview的evaluateJavascript方法,evaluateJavascript方法是系统API方法,所以evaluateJavascript方法提供的回调方法稳定性比较高。因此,本发明实施例通过重打包的机制,可以使得目标应用的进程包含测试包数据,进而可以基于测试包数据实现控件信息的抓取和事件模拟,整套方案不但可以适配微信小程序,对于所有X5内核的应用都可以适用,解决了现有技术中测试框架无法适配X5webview的问题。

[0103] 进一步的,请一并参见图2,图2是本发明实施例提供的一种测试应用的界面示意图。如图2所示,用户终端100的显示屏中显示测试应用的界面,测试应用的界面可以包括:应用列表200c、“脚本录制”按钮200a、“测试”按钮200b,其中,应用列表200c可以展示多个应用的名称,如图2所示的应用列表200c中显示A应用、B应用、C应用的名称。当用户从应用列表200c中点击选择A应用时,则可以将A应用确定为待测应用,当用户继续点击“脚本录制”按钮200a时,则可以对A应用进行重打包操作,并将重打包后的A应用确定为目标应用,并启动和显示A应用,并通过A应用的进程中的测试包数据实现对应用A进行测试脚本录制(录制的具体过程可以参见上述图1对应实施例中的S102),录制完测试脚本后,用户可以继续点击“测试”按钮200b,进而测试应用可以开始加载测试脚本,并再次启动和显示A应用,测试应用将加载测试脚本所生成的测试指令发送到A应用,并通过A应用的进程中的测试包数据以及测试指令实现对应用A进行自动化测试(自动化测试的具体过程可以参见上述图1对应实施例中的S103)。

[0104] 进一步的,请一并参见图3,图3是本发明实施例提供的一种测试应用和目标应用之间的进程交互示意图。如图3所示,测试应用以腾讯开发的xtest为例,目标应用以上述图2对应实施例中的A应用为例,启动xtest后,若用户通过上述图2所示的界面选择A应用进行测试脚本录制,则xtest对应的测试服务(xtestserver)可以将测试包数据(即class.dex)重打包到A应用的进程中,并启动A应用,根据A应用进程中的测试包数据执行应用初始化类(即进行katapplication初始化),并对方法拦截进行初始化(即进行art hook初始化),在这里开启attach方法拦截功能,当应用初始化完成后,每一个进程都会执行一次attach方法,在执行到该方法时,为xtest的进程和A应用的进程创建进程通信服务(即socketserver),等待xtest发送请求指令,即xtest的进程和A应用的进程可以通过socket通信,并通过class.dex实现对A应用的测试脚本录制。录制完测试脚本后,若用户通过上述图2所示的界面选择A应用进行自动化测试,则xtest可以加载测试脚本,并将所生成的测试指令通过socket通信发送到A应用进程中的socketserver,进而可以基于class.dex在A应用中处理测试指令(如提取测试指令所指示的操作行为以及待操作的控件的坐标信息),并将测试指令处理后所得到的坐标信息回调给系统测试框架(即instrumentation框架),并由instrumentation框架基于测试指令所指示的操作行为对坐标信息处的控件进行模拟操作。

[0105] 本发明实施例通过获取待测应用,并根据测试包数据对待测应用进行重打包,得到目标应用;目标应用的进程包含测试包数据;根据针对目标应用的操作指令、测试包数据,在目标应用的进程中获取网页视图,并获取网页视图中与操作指令相关联的控件信息,并根据控件信息以及操作指令所指示的操作行为,生成测试脚本;加载测试脚本,并根据测试脚本和测试包数据在目标应用的进程中获取网页视图,并在网页视图中查找控件信息对应的坐标信息,并对目标应用中位于坐标信息处的控件进行模拟操作,生成测试结果。可见通过重打包的机制,可以使得目标应用的进程包含测试包数据,进而可以基于测试包数据实现对原生的网页视图或自定义的网页视图(如X5 webview)中的控件信息的抓取和事件模拟,从而可以实现测试脚本的录制和自动化测试,整套方案不但可以适配基于X5 webview的微信小程序,对于所有X5内核的应用都可以适用,即本发明实施例可以对基于原生的网页视图或基于自定义的网页视图的应用进行自动化测试。

[0106] 请参见图4,是本发明实施例提供的一种测试脚本录制方法的流程示意图,所述方法中的S201-S206步骤是对上述图2对应实施例中的S102步骤的具体描述,所述方法可以包括:

[0107] S201,根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0108] 具体的,目标应用的进程中包含测试包数据,测试包数据是通过重打包的方式设置在目标应用的进程中,测试包数据的具体内容可以参见上述图2对应实施例中的S101。在执行自动化测试之前,还需生成目标应用的测试脚本,该测试脚本用于后续对目标应用进行自动化测试。在用户终端中的测试应用的界面中可以显示用于录制测试脚本的录制按键,当用户点击该录制按键,用户终端可以根据点击指令,触发测试应用发送脚本录制命令,目标应用中的测试包数据监听到该脚本录制命令后,该测试包数据可以根据该脚本录制命令开始记录用户对目标应用的所有操作行为。在测试脚本开始录制时,用户终端可以

根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并通过深度优先的遍历算法对所述视图结构树进行遍历,并将遍历结果存储在视图结构数组(即list数组)中。其中,由于测试包数据处于目标应用的进程中,所以通过测试包数据对应的测试模块可以直接在目标应用的进程中读取所述目标应用的界面的views树(即视图结构树)。由于是直接读取的views树,所以就算views树包含网页视图(如原生webview或X5 webview),也是可以读取到完整的views树,即可以成功的读取到views树中的网页视图。

[0109] S202,接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过查询代码段对所述网页视图的页面内容进行遍历,得到遍历结果;

[0110] 具体的,在录制测试脚本的过程中,每当接收到针对目标应用的操作指令(如由点击操作、滑动操作所产生的操作指令)时,都可以根据list数组检测该操作指令所指示的操作位置是否包含网页视图,若包含网页视图,则注入基于JavaScript的查询代码段,通过该查询代码段可以遍历网页视图的页面内容中的document,即遍历结果包含该document。

[0111] S203,获取所述操作指令所指示的操作位置的坐标信息,并从所述遍历结果中获取所述坐标信息对应的控件信息;

[0112] 具体的,用户终端进一步获取操作指令所指示的操作位置的坐标信息,其中,遍历结果可以包括网页页面中所有控件对应的坐标信息和控件信息(如xpath信息),因此,用户终端可以根据获取到的坐标信息,在遍历结果中获取该坐标信息对应的控件信息。

[0113] S204,根据所述目标应用中的回调函数将所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本;

[0114] 具体的,由于测试包数据处于目标应用的进程中,所以测试包数据所提供的测试功能也可以直接调用目标应用中原生webview或X5 webview的evaluateJavascript方法,evaluateJavascript方法是系统API方法,所以evaluateJavascript方法提供的回调方法稳定性比较高。进而用户终端可以通过evaluateJavascript提供的回调函数将控件信息返回测试应用,即用户对目标应用的每一个操作所产生的操作行为信息和被操作的控件信息均可以返回到测试应用中,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本,直至用户结束对目标应用的操作,则完成整个脚本录制过程。

[0115] 可选的,上述S202中,在接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,还可以通过loadURL从所述网页视图的页面内容中,获取所述操作指令所指示的操作位置的控件信息,其中,loadURL可以是一种脚本执行方法,具体用于加载脚本代码段并执行;拦截由loadURL获取到的所述控件信息,并通过编码语言注册通道将拦截到的所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。其中,虽然通过反射调用X5 webview的loadUrl方法,也能够抓取到网页页面的内容,但是loadUrl方法是无返回值的,所以要想返回控件信息到测试端,就需要利用hook技术拦截loadUrl方法,然后利用addJavaScriptInterface注册Java与Javascript通信的桥梁,该桥

梁即为编码语言注册通道,并通过编码语言注册通道可以实现控件信息的返回,进而以达到自动化测试的目的。其中,addJavaScriptInterface可以称之为Javascript接口注册方法,且用于向系统注册Javascript调用接口,注册后,就可以利用loadurl方法在java端调用Javascript方法。

[0116] 本发明实施例通过重打包的机制,可以使得目标应用的进程包含测试包数据,进而可以基于测试包数据实现对原生的网页视图或自定义的网页视图(如X5 webview)中的控件信息的抓取,从而可以实现测试脚本的录制,整套方案不但可以适配基于X5 webview的微信小程序的测试脚本录制,对于所有X5内核的应用都可以适用。

[0117] 请参见图5,是本发明实施例提供的一种自动化测试方法的流程示意图,所述方法中的S301-S305步骤是对上述图2对应实施例中S103步骤的具体描述,所述方法可以包括:

[0118] S301,在所述测试应用的进程中加载所述测试脚本,并通过所述进程通信服务将所述测试脚本中的测试指令发送到所述目标应用的进程;

[0119] 具体的,录制完测试脚本后,用户终端可以通过测试应用对测试脚本进行加载,测试应用可以在加载的过程中生成一条条测试指令,并逐一将每条测试指令通过socket通信发送到目标应用的进程通信服务中,进而可以在目标应用中基于测试包数据逐一执行每条测试指令所指示的内容。其中,发送测试指令的顺序与录制测试脚本过程中的每一个操作行为的顺序一致,且一个测试指令所指示的内容即为录制测试脚本过程中针对目标应用的一条操作行为,且测试指令还可以包含录制过程中与操作指令相关联的控件信息。

[0120] S302,根据所述测试包数据和所述测试指令,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0121] 具体的,目标应用接收到测试指令后,即可基于测试包数据反射读取到测试指令所对应的activity的views树(由于测试包数据与目标应用处于同一进程中,所以基于测试包数据可以直接获取到views树,同样也可以获取到views树中的原生webview或X5 webview),该views树可以称之为目标应用的界面对应的视图结构树,再通过深度优先的遍历算法遍历整个views树,再将遍历结果存储在list数组(即视图结构数组)中。

[0122] S303,若根据所述视图结构数组在所述测试指令所指示的待操作对象中查找到网页视图,则通过查询代码段在所述网页视图中,查找所述测试脚本中的所述控件信息所对应的控件,并获取所述控件的坐标信息;

[0123] 具体的,用户终端可以根据测试指令以及list数组查找测试指令所包含的操作对象是否有网页视图(如原生webview或X5 webview),如果有,则注入基于JavaScript的查询代码段,由于测试指令中包含控件信息(如控件的xpath信息),所以无需再基于该查询代码段重新遍历移动端网页页面的document,只需通过该查询代码段查找测试指令中的控件信息所对应的控件,并获取该控件在网页页面中的坐标信息。

[0124] S304,通过回调函数将所述控件的坐标信息返回至系统测试框架;

[0125] 具体的,由于测试包数据处于目标应用的进程中,所以测试包数据对应的测试模块也可以直接调用目标应用中原生webview或X5 webview的evaluateJavascript方法,因此,该测试模块可以通过evaluateJavascript提供的回调函数将控件的坐标信息返回到系统测试框架,该系统测试框架可以是指instrumentation框架。

[0126] S305,基于所述系统测试框架,对所述目标应用中位于所述坐标信息处的所述控

件进行模拟操作,生成测试结果;

[0127] 具体的,在基于instrumentation框架、测试指令中的操作行为,对获得的坐标信息处的控件进行模拟操作,如测试指令包含的操作行为是“对控件A进行点击操作”,则基于instrumentation框架可以对坐标信息处的控件A进行模拟点击,进而得到该模拟操作对应的输出结果,当完成测试脚本中的每一条测试指令,即完成了对目标应用的自动化测试,进而将每一条测试指令对应的输出结果确定为测试结果,该测试结果可以显示在用户终端的界面中,测试人员即可将测试结果与预期结果进行对比,以分析出目标应用的性能。或者,用户终端也可以将测试指令对应的输出结果与预期结果进行对比,将对比结果确定为测试结果。

[0128] 本发明实施例通过重打包的机制,可以使得目标应用的进程包含测试包数据,进而可以基于测试包数据实现对原生的网页视图或自定义的网页视图(如X5 webview)中的控件信息的抓取和事件模拟,从而可以实现基于测试脚本的自动化测试,整套方案不但可以适配基于X5 webview的微信小程序,对于所有X5内核的应用都可以适用,即本发明实施例可以对基于原生的网页视图或基于自定义的网页视图的应用进行自动化测试。

[0129] 请参见图6,是本发明实施例提供的一种页面检测方法的流程示意图,所述方法可以在上述图2对应实施例中的S102步骤之后执行,即该方法可以在生成测试脚本之后,且在根据测试脚本进行自动化测试的过程中执行。所述方法可以包括:

[0130] S401,获取检测指令,并根据所述检测指令设置自定义网页回调接口服务;

[0131] 具体的,在根据测试脚本对目标应用进行自动化测试过程中的任一时刻,均可以通过测试脚本发起检测指令,即该检测指令可以预先封装在测试脚本中,测试应用在加载测试脚本的过程中,可以将测试脚本中的检测指令通过socket通信发送到目标应用中测试包数据对应的测试模块中,该测试模块可以在接收到检测指令的同时还可以接收测试脚本中的测试指令,即该测试模块可以同时根据检测指令进行检测操作,并根据测试指令进行自动化测试操作。该测试模块接收到检测指令后,可以根据检测指令获取网页视图(如原生webview或X5webview),并检测该网页视图是否包含网页回调接口服务,该网页回调接口服务可以是指webviewclient。若网页视图包含webviewclient,则该测试模块可以对webviewclient中的功能函数进行重写,得到自定义网页回调接口服务,即自定义网页回调接口服务包含重写后的功能函数,这些重写后的功能函数可以包括第一功能函数(onPageStarted)、第二功能函数(onPageFinished)、第三功能函数(shouldOverrideUrlLoading)、第四功能函数(onReceivedError)、第五功能函数(shouldInterceptRequest)。onPageStarted是用于检测页面加载的开始时间,并在开始加载网页时会回调;onPageFinished是用于检测页面加载完成的时间,并在结束加载网页时会回调;shouldOverrideUrlLoading是用于通过拦截url跳转来检测页面跳转信息;onReceivedError是用于检测页面加载异常信息,并在加载页面出错的时候会回调;shouldInterceptRequest是用于统计资源加载数,并在每一次请求资源时都会通过该函数来回调。若网页视图未包含webviewclient,则该测试模块可以直接为网页视图创建对应的自定义网页回调接口服务,所创建的自定义网页回调接口服务同样包含onPageStarted、onPageFinished、shouldOverrideUrlLoading、onReceivedError、shouldInterceptRequest。

[0132] S402,若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测和记录所述目标应用的界面中的网页页面检测信息;

[0133] 具体的,若检测到所述目标应用的界面包含所述网页视图,即说明目标应用跳转到移动端网页页面(如微信应用进入小程序时所展现的网页页面),此时,可以触发该自定义网页回调接口服务检测目标应用中的网页页面;若检测到所述网页页面处于开始加载状态,则通过所述自定义网页回调接口服务中的第一功能函数记录网页加载起始时间信息;若检测到所述网页页面处于结束加载状态,则通过所述自定义网页回调接口服务中的第二功能函数记录网页加载结束时间信息;若检测到所述网页页面处于页面跳转状态,则通过所述自定义网页回调接口服务中的第三功能函数记录页面跳转信息;若检测到所述网页页面处于页面加载异常状态,则通过所述自定义网页回调接口服务中的第四功能函数记录页面加载异常信息;若检测到所述网页页面处于资源请求状态,则通过所述自定义网页回调接口服务中的第五功能函数统计资源加载数量。将所述网页加载起始时间信息、网页加载结束时间信息、页面跳转信息、页面加载异常信息、资源加载数量,确定为所述网页页面检测信息。其中,通过网页加载起始时间信息和网页加载结束时间信息,可以计算出页面加载时间。

[0134] S403,将所述网页页面检测信息添加至所述测试结果,并对添加后的测试结果进行显示。

[0135] 具体的,将所述网页页面检测信息添加至所述测试结果,使得网页页面检测信息和通过自动化测试所得到的测试结果可以一并以报告的形式在用户终端的显示屏中进行显示。

[0136] 本发明实施例通过重打包的机制,可以使得目标应用的进程包含测试包数据,进而可以基于测试包数据实现对原生的网页视图或自定义的网页视图(如X5 webview)中的控件信息的抓取和事件模拟,从而可以实现测试脚本的录制和自动化测试,整套方案不但可以适配基于X5 webview的微信小程序,对于所有X5内核的应用都可以适用,即本发明实施例可以对基于原生的网页视图或基于自定义的网页视图的应用进行自动化测试;同时还能对于网页页面进行有效的检测,能够获取到页面加载时间、页面跳转信息、页面加载异常信息、资源加载数量。

[0137] 请参见图7,是本发明实施例提供的一种应用测试装置的结构示意图。该应用测试装置1可以包括:重打包模块10、初始化模块20、脚本生成模块30、测试模块40、设置模块50、检测模块60、显示模块70;

[0138] 重打包模块10,用于获取待测应用,并根据测试包数据对所述待测应用进行重打包,得到目标应用;所述目标应用的进程包含所述测试包数据;

[0139] 初始化模块20,用于启动所述目标应用,并初始化所述目标应用中的所述测试包数据,并为所述目标应用的进程和测试应用的进程创建进程通信服务;所述测试应用的进程用于加载所述测试脚本;

[0140] 脚本生成模块30,用于根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本;

[0141] 测试模块40,用于加载所述测试脚本,并根据所述测试脚本和所述测试包数据在

所述目标应用的进程中查找所述网页视图,并在所述网页视图中查找所述控件信息对应的坐标信息,并对所述目标应用中位于所述坐标信息处的控件进行模拟操作,生成测试结果。

[0142] 其中,重打包模块10、初始化模块20、脚本生成模块30、测试模块40的具体功能实现方式可以参见上述图1对应实施例中的S101-S103,这里不再进行赘述。

[0143] 设置模块50,用于获取检测指令,并根据所述检测指令设置自定义网页回调接口服务;

[0144] 检测模块60,用于若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测和记录所述目标应用的界面中的网页页面检测信息;

[0145] 显示模块70,用于将所述网页页面检测信息添加至所述测试结果,并对添加后的测试结果进行显示。

[0146] 其中,设置模块50、检测模块60、显示模块70的具体功能实现方式可以参见上述图6对应实施例中的S401-S403,这里不再进行赘述。

[0147] 进一步的,再请参见图7,在图7中该脚本生成模块30可以包括:第一数组生成单元301、页面遍历单元302、第一控件获取单元303、第一脚本转化单元304、第二数组生成单元305、第二控件获取单元306、第二脚本转化单元307;

[0148] 第一数组生成单元301,用于根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0149] 页面遍历单元302,用于接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过查询代码段对所述网页视图的页面内容进行遍历,得到遍历结果;

[0150] 第一控件获取单元303,用于获取所述操作指令所指示的操作位置的坐标信息,并从所述遍历结果中获取所述坐标信息对应的控件信息;

[0151] 第一脚本转化单元304,用于根据所述目标应用中的回调函数将所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

[0152] 第二数组生成单元305,用于根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0153] 第二控件获取单元306,用于接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过loadURL从所述网页视图的页面内容中,获取所述操作指令所指示的操作位置的控件信息;

[0154] 第二脚本转化单元307,用于拦截由loadURL获取到的所述控件信息,并通过编码语言注册通道将拦截到的所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

[0155] 其中,第一数组生成单元301、页面遍历单元302、第一控件获取单元303、第一脚本转化单元304、第二数组生成单元305、第二控件获取单元306、第二脚本转化单元307的具体功能实现方式可以参见上述图4对应实施例中的S201-S204,这里不再进行赘述。其中,当第一数组生成单元301、页面遍历单元302、第一控件获取单元303、第一脚本转化单元304用于

脚本录制时,第二数组生成单元305、第二控件获取单元306、第二脚本转化单元307将停止操作。当第二数组生成单元305、第二控件获取单元306、第二脚本转化单元307用于脚本录制时,第一数组生成单元301、页面遍历单元302、第一控件获取单元303、第一脚本转化单元304将停止操作。其中,第一数组生成单元301和第二数组生成单元305可以合并为同一个单元。

[0156] 进一步的,再请参见图7,在图7中该测试模块40可以包括:指令发送单元401、第三数组生成单元402、坐标获取单元403、坐标发送单元404、模拟操作单元405;

[0157] 指令发送单元401,用于在所述测试应用的进程中加载所述测试脚本,并通过所述进程通信服务将所述测试脚本中的测试指令发送到所述目标应用的进程;

[0158] 第三数组生成单元402,用于根据所述测试包数据和所述测试指令,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0159] 坐标获取单元403,用于若根据所述视图结构数组在所述测试指令所指示的待操作对象中查找到网页视图,则通过查询代码段在所述网页视图中,查找所述测试脚本中的所述控件信息所对应的控件,并获取所述控件的坐标信息;

[0160] 坐标发送单元404,用于通过回调函数将所述控件的坐标信息返回至系统测试框架;

[0161] 模拟操作单元405,用于基于所述系统测试框架,对所述目标应用中位于所述坐标信息处的所述控件进行模拟操作,生成测试结果。

[0162] 其中,指令发送单元401、第三数组生成单元402、坐标获取单元403、坐标发送单元404、模拟操作单元405的具体实现方式可以参见上述图5对应实施例中的S301-S305,这里不再进行赘述。

[0163] 进一步的,再请参见图7,在图7中的该设置模块50可以包括:指令获取单元502、函数重写单元501、创建单元503;

[0164] 指令获取单元502,用于获取检测指令;

[0165] 函数重写单元501,用于若所述网页视图包含网页回调接口服务,则根据所述检测指令获取所述网页回调接口服务,并对所述网页回调接口服务中的功能函数进行重写,得到自定义网页回调接口服务;

[0166] 创建单元503,用于若所述网页视图未包含网页回调接口服务,则创建所述自定义网页回调接口服务。

[0167] 其中,指令获取单元502、函数重写单元501、创建单元503的具体实现方式可以参见上述图6对应实施例中的S401,这里不再进行赘述。

[0168] 进一步的,再请参见图7,在图7中的该检测模块60可以包括:触发单元601、检测信息记录单元602、确定单元603;

[0169] 触发单元601,用于若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测所述目标应用中的网页页面;

[0170] 检测信息记录单元602,用于若检测到所述网页页面处于开始加载状态,则通过所述自定义网页回调接口服务中的第一功能函数记录网页加载起始时间信息;

[0171] 所述检测信息记录单元602,还用于若检测到所述网页页面处于结束加载状态,则

通过所述自定义网页回调接口服务中的第二功能函数记录网页加载结束时间信息；

[0172] 所述检测信息记录单元602,还用于若检测到所述网页页面处于页面跳转状态,则通过所述自定义网页回调接口服务中的第三功能函数记录页面跳转信息；

[0173] 所述检测信息记录单元602,还用于若检测到所述网页页面处于页面加载异常状态,则通过所述自定义网页回调接口服务中的第四功能函数记录页面加载异常信息；

[0174] 所述检测信息记录单元602,还用于若检测到所述网页页面处于资源请求状态,则通过所述自定义网页回调接口服务中的第五功能函数统计资源加载数量；

[0175] 确定单元603,用于将所述网页加载起始时间信息、网页加载结束时间信息、页面跳转信息、页面加载异常信息、资源加载数量,确定为所述网页页面检测信息。

[0176] 其中,触发单元601、检测信息记录单元602、确定单元603的具体实现方式可以参见上述图6对应实施例中的S402,这里不再进行赘述。

[0177] 本发明实施例通过重打包的机制,可以使得目标应用的进程包含测试包数据,进而可以基于测试包数据实现对原生的网页视图或自定义的网页视图(如X5 webview)中的控件信息的抓取和事件模拟,从而可以实现测试脚本的录制和自动化测试,整套方案不但可以适配基于X5 webview的微信小程序,对于所有X5内核的应用都可以适用,即本发明实施例可以对基于原生的网页视图或基于自定义的网页视图的应用进行自动化测试;同时还能对于网页页面进行有效的检测,能够获取到页面加载时间、页面跳转信息、页面加载异常信息、资源加载数量。

[0178] 请参见图8,是本发明实施例提供的一种终端设备的结构示意图。如图8所示,所述终端设备1000可以包括:处理器1001,网络接口1004和存储器1005,此外,所述终端设备1000还可以包括:用户接口1003,和至少一个通信总线1002。其中,通信总线1002用于实现这些组件之间的连接通信。其中,用户接口1003可以包括显示屏(Display)、键盘(Keyboard),可选用户接口1003还可以包括标准的有线接口、无线接口。网络接口1004可选的可以包括标准的有线接口、无线接口(如WI-FI接口)。存储器1004可以是高速RAM存储器,也可以是非不稳定的存储器(non-volatile memory),例如至少一个磁盘存储器。存储器1004可选的还可以是至少一个位于远离前述处理器1001的存储装置。如图8所示,作为一种计算机存储介质的存储器1004中可以包括操作系统、网络通信模块、用户接口模块以及设备控制应用程序。

[0179] 在图8所示的终端设备1000中,网络接口1004可提供网络通讯功能;而用户接口1003主要用于为用户提供输入的接口;而处理器1001可以用于调用存储器1004中存储的设备控制应用程序,以实现:

[0180] 获取待测应用,并根据测试包数据对所述待测应用进行重打包,得到目标应用;所述目标应用的进程包含所述测试包数据;

[0181] 根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本;

[0182] 加载所述测试脚本,并根据所述测试脚本和所述测试包数据在所述目标应用的进程中查找所述网页视图,并在所述网页视图中查找所述控件信息对应的坐标信息,并对所述目标应用中位于所述坐标信息处的控件进行模拟操作,生成测试结果。

[0183] 具体的,终端设备1000可以通过测试应用获取待测应用,例如,测试应用中可以展示应用列表,该应用列表可以包括多个应用名称,用户可以点击该应用列表中的某个应用名称,进而终端设备1000根据点击指令选取该应用名称对应的应用,作为待测应用。测试应用可以是指安装在终端设备1000中的用于对应用程序进行测试的应用。该测试应用可以包括测试服务,该测试服务可以提供测试包数据以及重打包功能,即测试应用选取待测应用后,可以通过测试包数据对所述待测应用进行重打包,即测试服务可以将测试包数据打包至待测应用中,再将包含测试包数据的待测应用称之为目标应用,测试包数据可以为dex文件。其中,测试包数据可以包括测试模块代码,该测试模块代码具有获取当前页面中的控件信息的功能,因此,目标应用通过运行该测试模块代码,可以使得目标应用具备获取当前页面中的控件信息的功能。

[0184] 对待测应用进行重打包后,可以启动目标应用,在启动过程中,还可以一并加载目标应用中的测试包数据,并对测试包数据进行初始化,该初始化的过程可以包括:初始化katapplication、初始化Art Hook。katapplication可以称为自定义的应用初始化类,在Android系统中,每个应用启动都会先执行应用初始化类,katapplication在这里的作用是在应用初始化时候将建立通信服务和注册接口拦截服务,即在该初始化的过程中可以基于katapplication为目标应用的进程和测试应用的进程创建进程通信服务,该进程通信服务可以为socketserver,socketserver可以提供socket通信,即通过socketserver可以实现两个不同应用进程之间的数据通信。该进程通信服务位于目标应用的进程中。其中,Art Hook可称之为方法拦截,作用是在系统调用到某个方法时候,会被拦截到,从而在执行这个方法前后执行指定功能。因此,本发明实施例通过初始化Art Hook,可以在被测应用启动后基于Art Hook进行拦截,以开启socket通信服务。初始化katapplication、初始化Art Hook的过程均在目标应用的进程中实现。

[0185] 其中,网页视图可以包括原生webview或X5 webview。在执行自动化测试之前,还需生成目标应用的测试脚本,该测试脚本用于后续对目标应用进行自动化测试。在终端设备1000中的测试应用的界面中可以显示用于录制测试脚本的录制按键,当用户点击该录制按键,终端设备1000可以根据点击指令,触发测试应用发送脚本录制命令,测试应用具体可以通过socket通信将脚本录制命令发送到目标应用,目标应用中的socketserver监听到该脚本录制命令后,可以基于测试包数据所提供的功能以及该脚本录制命令,开始记录用户对目标应用的所有操作行为(包括如对目标应用进行点击、滑动等操作),并可以记录每个操作行为中被操作的控件的信息,控件的信息可以包括控件的xpath(可扩展标记语言路径语言)信息。记录每个操作行为中被操作的控件的信息的具体过程可以为:基于目标应用中的测试包数据可以反射读取到当前activity的views树(由于测试包数据与目标应用处于同一进程中,所以基于测试包数据可以直接获取到views树,同样也可以获取到views树中的原生webview或X5 webview),再通过深度优先的遍历算法遍历整个views树,再存储在list数组中,然后根据list数组以及录制过程中针对目标应用的操作指令(如点击、滑动所产生的操作指令),查找操作指令所指示的坐标信息是否有原生webview或X5 webview,如果有,则注入基于JavaScript(一种直译式脚本语言,是一种动态类型、弱类型、基于原型的语言,内置支持类型)的查询代码段,通过该查询代码段可以遍历移动端网页页面(如微信小程序中的网页页面)的document,通过操作指令所指示的坐标信息在document中查找符

合条件的控件(即位于操作指令所指示的坐标信息的控件),并获取符合条件的该控件的控件信息,最后通过evaluateJavascript提供的回调函数将控件信息、所记录的所有操作行为返回至测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本,即用户对目标应用的每一个操作所产生的操作行为信息和被操作的控件信息均可以被转化为测试脚本,直至用户结束对目标应用的操作,则完成整个脚本录制过程。其中,evaluateJavascript是Android系统提供的一个方法名,也可以称之为Javascript代码执行函数。

[0186] 其中,录制完测试脚本后,终端设备1000可以通过测试应用对测试脚本进行加载,测试应用可以在加载的过程中生成一条条测试指令,并逐一将每条测试指令通过socket通信发送到目标应用的进程通信服务中,进而可以基于测试包数据在目标应用中逐一执行每条测试指令所指示的内容。其中,发送测试指令的顺序与录制测试脚本过程中的每一个操作行为的顺序一致,且一个测试指令所指示的内容即为录制测试脚本过程中针对目标应用的一条操作行为,且测试指令还可以包含录制过程中与操作指令相关联的控件信息。目标应用接收到测试指令后,即可基于测试包数据反射读取到测试指令所对应的activity的views树(由于测试包数据与目标应用处于同一进程中,所以基于测试包数据可以直接获取到views树,同样也可以获取到views树中的原生webview或X5 webview),再通过深度优先的遍历算法遍历整个views树,再存储在list数组中,然后根据测试指令和list数组,查找测试指令所包含的操作对象是否有原生webview或X5 webview,如果有,则注入基于JavaScript的查询代码段,由于测试指令中包含控件信息(如控件的xpath信息),所以无需再基于该查询代码段重新遍历移动端网页页面(即目标应用的网页页面)的document,只需通过该查询代码段查找测试指令中的控件信息所对应的控件,并获取该控件在网页页面中的坐标信息,再通过evaluateJavascript提供的回调函数将控件的坐标信息返回到instrumentation框架。在instrumentation框架中,基于测试指令中的操作行为,对获得的坐标信息处的控件进行模拟操作,如测试指令包含的操作行为是“对控件A进行点击操作”,则基于instrumentation框架可以对坐标信息处的控件A进行模拟点击,进而得到该模拟操作对应的输出结果,当完成测试脚本中的每一条测试指令,即完成了对目标应用的自动化测试,进而将每一条测试指令对应的输出结果确定为测试结果,测试人员即可将测试结果与预期结果进行对比,以分析出目标应用的性能。

[0187] 由于测试包数据处于目标应用的进程中,所以测试包数据所提供的测试功能可直接抓取目标应用中原生webview或X5 webview的页面内容,因此,就算目标应用使用的是自定义的X5 webview(如目标应用为微信中的小程序),也可以通过测试包数据所提供的测试功能直接抓取到目标应用的网页页面内容,因此,可以顺利地为目标应用进行测试脚本录制,也可以顺利地基于测试脚本对目标应用进行自动化测试。由于测试包数据处于目标应用的进程中,所以测试包数据所提供的测试功能也可以直接调用目标应用中原生webview或X5webview的evaluateJavascript方法,evaluateJavascript方法是系统API方法,所以evaluateJavascript方法提供的回调方法稳定性比较高。因此,本发明实施例通过重打包的机制,可以使得目标应用的进程包含测试包数据,进而可以基于测试包数据实现控件信息的抓取和事件模拟,整套方案不但可以适配微信小程序,对于所有X5内核的应用都可以适用,解决了现有技术中测试框架无法适配X5webview的问题。

[0188] 在一个实施例中,处理器1001还可以执行以下步骤:

[0189] 启动所述目标应用,并初始化所述目标应用中的所述测试包数据,并为所述目标应用的进程和测试应用的进程创建进程通信服务;所述测试应用的进程用于加载所述测试脚本。

[0190] 在一个实施例中,处理器1001在执行根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本时,具体执行以下步骤:

[0191] 根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0192] 接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过查询代码段对所述网页视图的页面内容进行遍历,得到遍历结果;

[0193] 获取所述操作指令所指示的操作位置的坐标信息,并从所述遍历结果中获取所述坐标信息对应的控件信息;

[0194] 根据所述目标应用中的回调函数将所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

[0195] 在一个实施例中,处理器1001在执行根据针对所述目标应用的操作指令、所述测试包数据,在所述目标应用的进程中查找网页视图,并获取所述网页视图中与所述操作指令相关联的控件信息,并根据所述控件信息以及所述操作指令所指示的操作行为,生成测试脚本时,具体执行以下步骤:

[0196] 根据所述测试包数据,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0197] 接收针对所述目标应用的操作指令,并根据所述视图结构数组在所述操作指令所指示的操作位置查找到网页视图时,通过loadURL从所述网页视图的页面内容中,获取所述操作指令所指示的操作位置的控件信息;

[0198] 拦截由loadURL获取到的所述控件信息,并通过编码语言注册通道将拦截到的所述控件信息返回至所述测试应用,并在所述测试应用中将所述控件信息以及所述操作指令所指示的操作行为转化为所述测试脚本。

[0199] 在一个实施例中,处理器1001在执行加载所述测试脚本,并根据所述测试脚本和所述测试包数据在所述目标应用的进程中查找所述网页视图,并在所述网页视图中查找所述控件信息对应的坐标信息,并对所述目标应用中位于所述坐标信息处的控件进行模拟操作,生成测试结果时,具体执行以下步骤:

[0200] 在所述测试应用的进程中加载所述测试脚本,并通过所述进程通信服务将所述测试脚本中的测试指令发送到所述目标应用的进程;

[0201] 根据所述测试包数据和所述测试指令,在所述目标应用的进程中读取所述目标应用的界面对应的视图结构树,并对所述视图结构树进行遍历,生成视图结构数组;

[0202] 若根据所述视图结构数组在所述测试指令所指示的待操作对象中查找到网页视图,则通过查询代码段在所述网页视图中,查找所述测试脚本中的所述控件信息所对应的

控件,并获取所述控件的坐标信息;

[0203] 通过回调函数将所述控件的坐标信息返回至系统测试框架;

[0204] 基于所述系统测试框架,对所述目标应用中位于所述坐标信息处的所述控件进行模拟操作,生成测试结果。

[0205] 在一个实施例中,处理器1001还可以执行以下步骤:

[0206] 获取检测指令,并根据所述检测指令设置自定义网页回调接口服务;

[0207] 若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测和记录所述目标应用的界面中的网页页面检测信息;

[0208] 将所述网页页面检测信息添加至所述测试结果,并对添加后的测试结果进行显示。

[0209] 在一个实施例中,处理器1001在执行获取检测指令,并根据所述检测指令设置自定义网页回调接口服务时,可以执行以下步骤:

[0210] 获取检测指令;

[0211] 若所述网页视图包含网页回调接口服务,则根据所述检测指令获取所述网页回调接口服务,并对所述网页回调接口服务中的功能函数进行重写,得到自定义网页回调接口服务;

[0212] 若所述网页视图未包含网页回调接口服务,则创建所述自定义网页回调接口服务。

[0213] 在一个实施例中,处理器1001在执行若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测和记录所述目标应用的界面中的网页页面检测信息时,可以执行以下步骤:

[0214] 若检测到所述目标应用的界面包含所述网页视图,则触发所述自定义网页回调接口服务检测所述目标应用中的网页页面;

[0215] 若检测到所述网页页面处于开始加载状态,则通过所述自定义网页回调接口服务中的第一功能函数记录网页加载起始时间信息;

[0216] 若检测到所述网页页面处于结束加载状态,则通过所述自定义网页回调接口服务中的第二功能函数记录网页加载结束时间信息;

[0217] 若检测到所述网页页面处于页面跳转状态,则通过所述自定义网页回调接口服务中的第三功能函数记录页面跳转信息;

[0218] 若检测到所述网页页面处于页面加载异常状态,则通过所述自定义网页回调接口服务中的第四功能函数记录页面加载异常信息;

[0219] 若检测到所述网页页面处于资源请求状态,则通过所述自定义网页回调接口服务中的第五功能函数统计资源加载数量;

[0220] 将所述网页加载起始时间信息、网页加载结束时间信息、页面跳转信息、页面加载异常信息、资源加载数量,确定为所述网页页面检测信息。

[0221] 其中,自定义网页回调接口服务可以包括第一功能函数(onPageStarted)、第二功能函数(onPageFinished)、第三功能函数(shouldOverrideUrlLoading)、第四功能函数(onReceivedError)、第五功能函数(shouldInterceptRequest)。onPageStarted是用于检测页面加载的开始时间,并在开始加载网页时会回调;onPageFinished是用于检测页面加

载完成的时间,并在结束加载网页时会回调;shouldOverrideUrlLoading是用于通过拦截url跳转来检测页面跳转信息;onReceivedError是用于检测页面加载异常信息,并在加载页面出错的时候会回调;shouldInterceptRequest是用于统计资源加载数,并在每一次请求资源时都会通过该函数来回调。

[0222] 本发明实施例通过重打包的机制,可以使得目标应用的进程包含测试包数据,进而可以基于测试包数据实现对原生的网页视图或自定义的网页视图(如X5 webview)中的控件信息的抓取和事件模拟,从而可以实现测试脚本的录制和自动化测试,整套方案不但可以适配基于X5 webview的微信小程序,对于所有X5内核的应用都可以适用,即本发明实施例可以对基于原生的网页视图或基于自定义的网页视图的应用进行自动化测试;同时还能对于网页页面进行有效的检测,能够获取到页面加载时间、页面跳转信息、页面加载异常信息、资源加载数量。

[0223] 此外,这里需要指出的是:本发明实施例还提供了一种计算机存储介质,且所述计算机存储介质中存储有前文提及的应用测试装置1所执行的计算机程序,且所述计算机程序包括程序指令,当所述处理器执行所述程序指令时,能够执行前文图1到图6所对应实施例中对所述应用测试方法的描述,因此,这里将不再进行赘述。另外,对采用相同方法的有益效果描述,也不再进行赘述。对于本发明所涉及的计算机存储介质实施例中未披露的技术细节,请参照本发明方法实施例的描述。

[0224] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)或随机存储记忆体(Random Access Memory,RAM)等。

[0225] 以上所揭露的仅为本发明较佳实施例而已,当然不能以此来限定本发明之权利范围,因此依本发明权利要求所作的等同变化,仍属本发明所涵盖的范围。

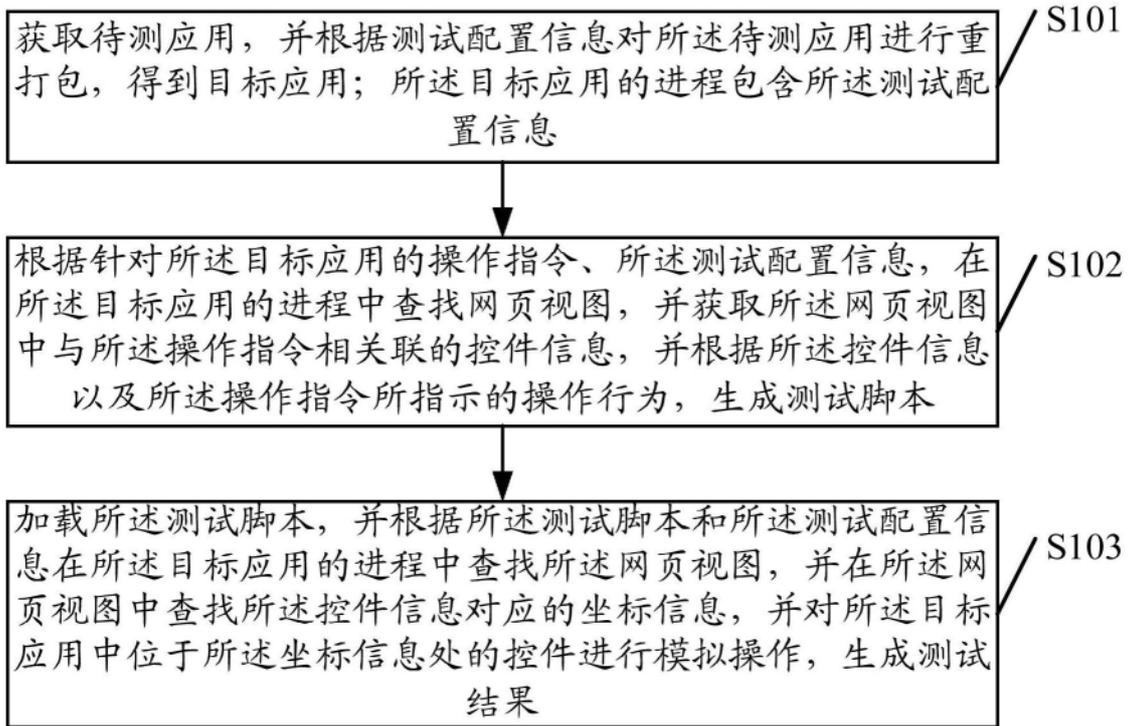


图1

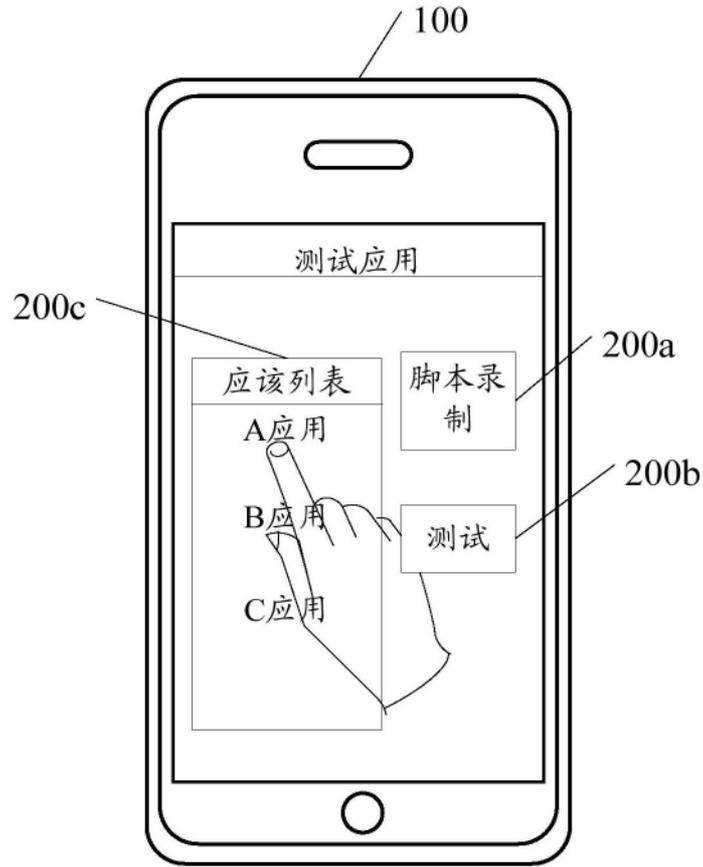


图2

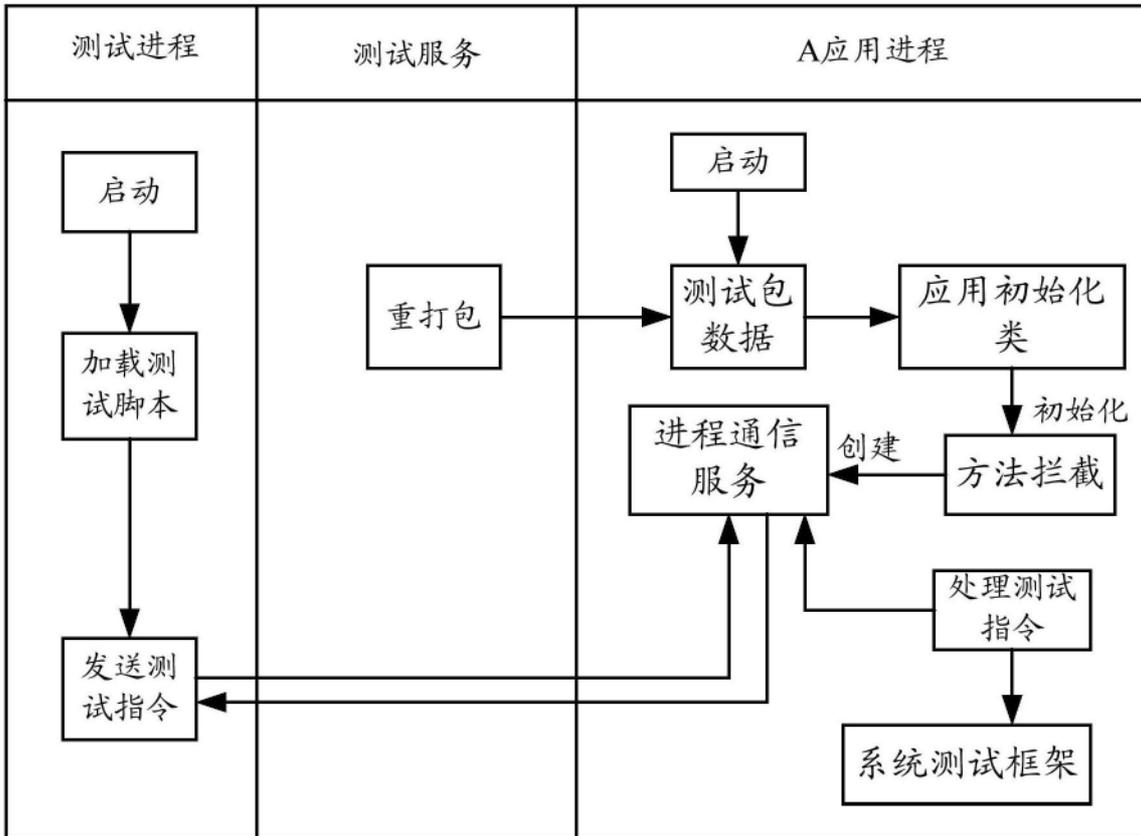


图3

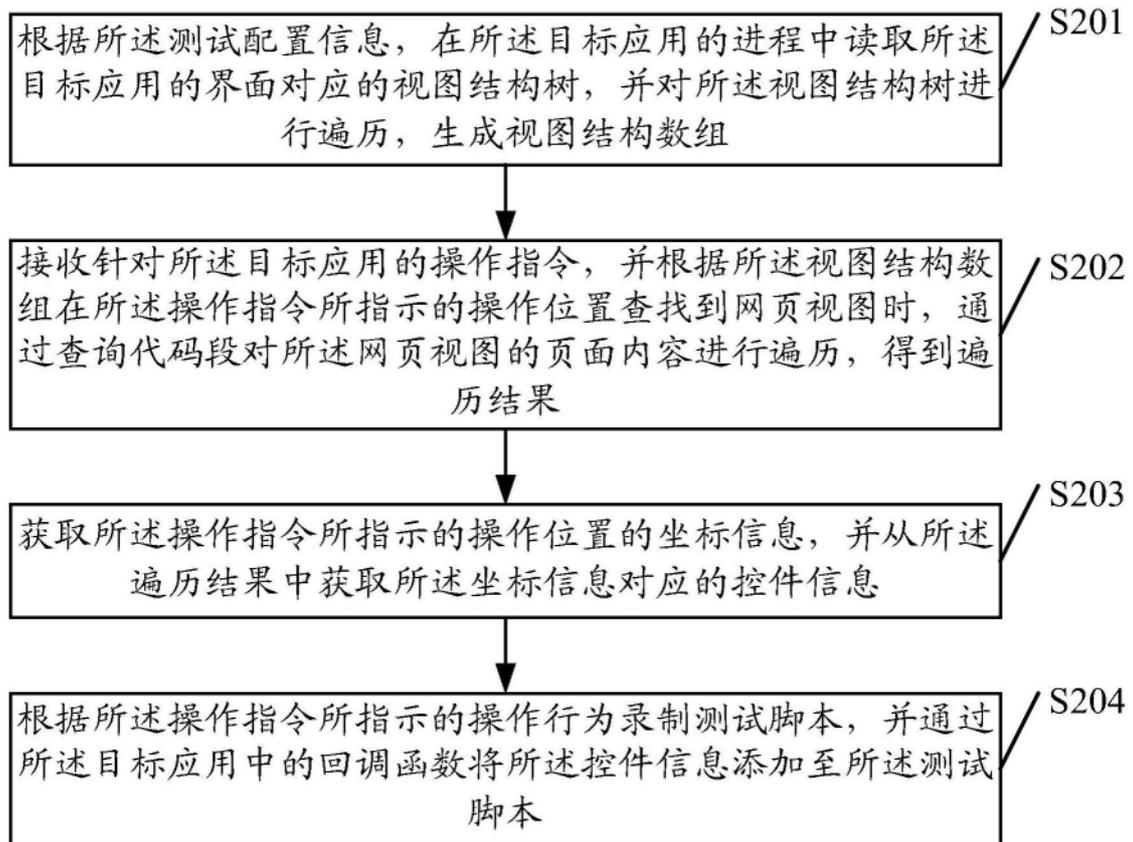


图4

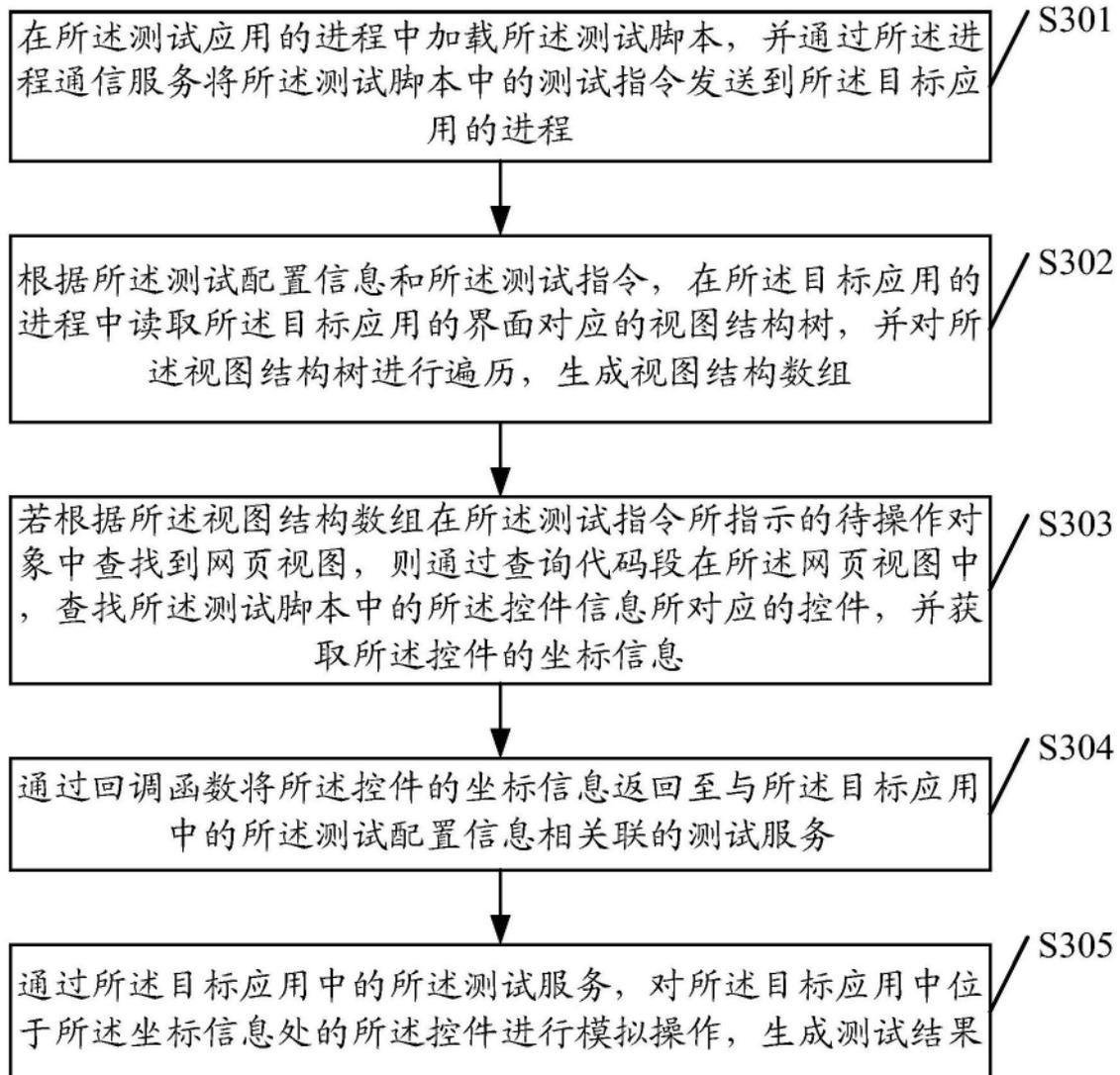


图5

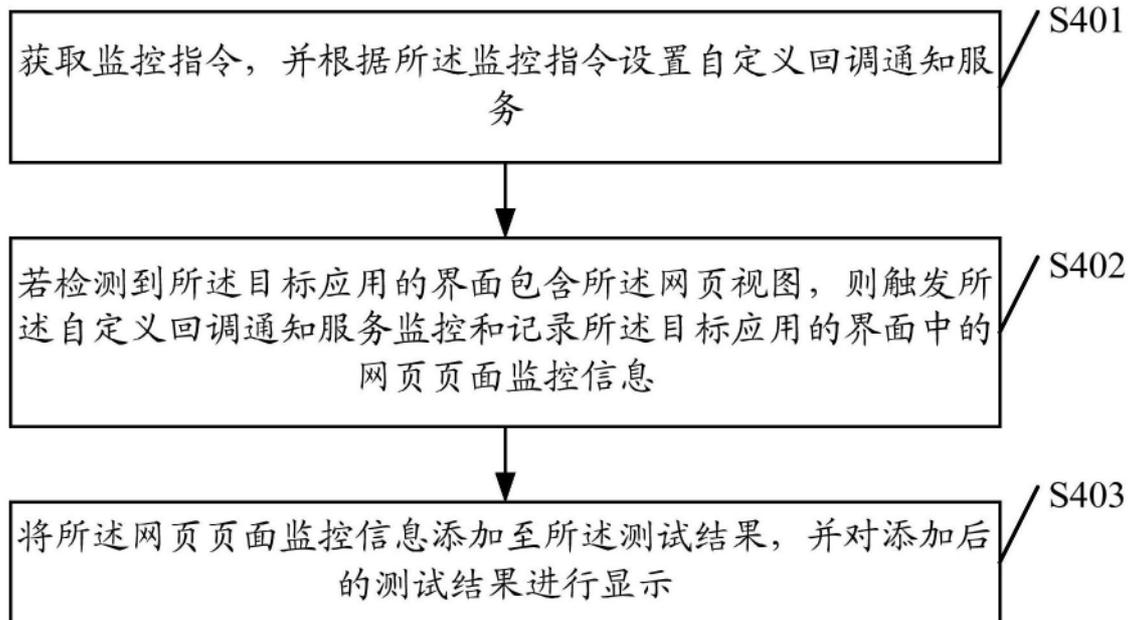


图6

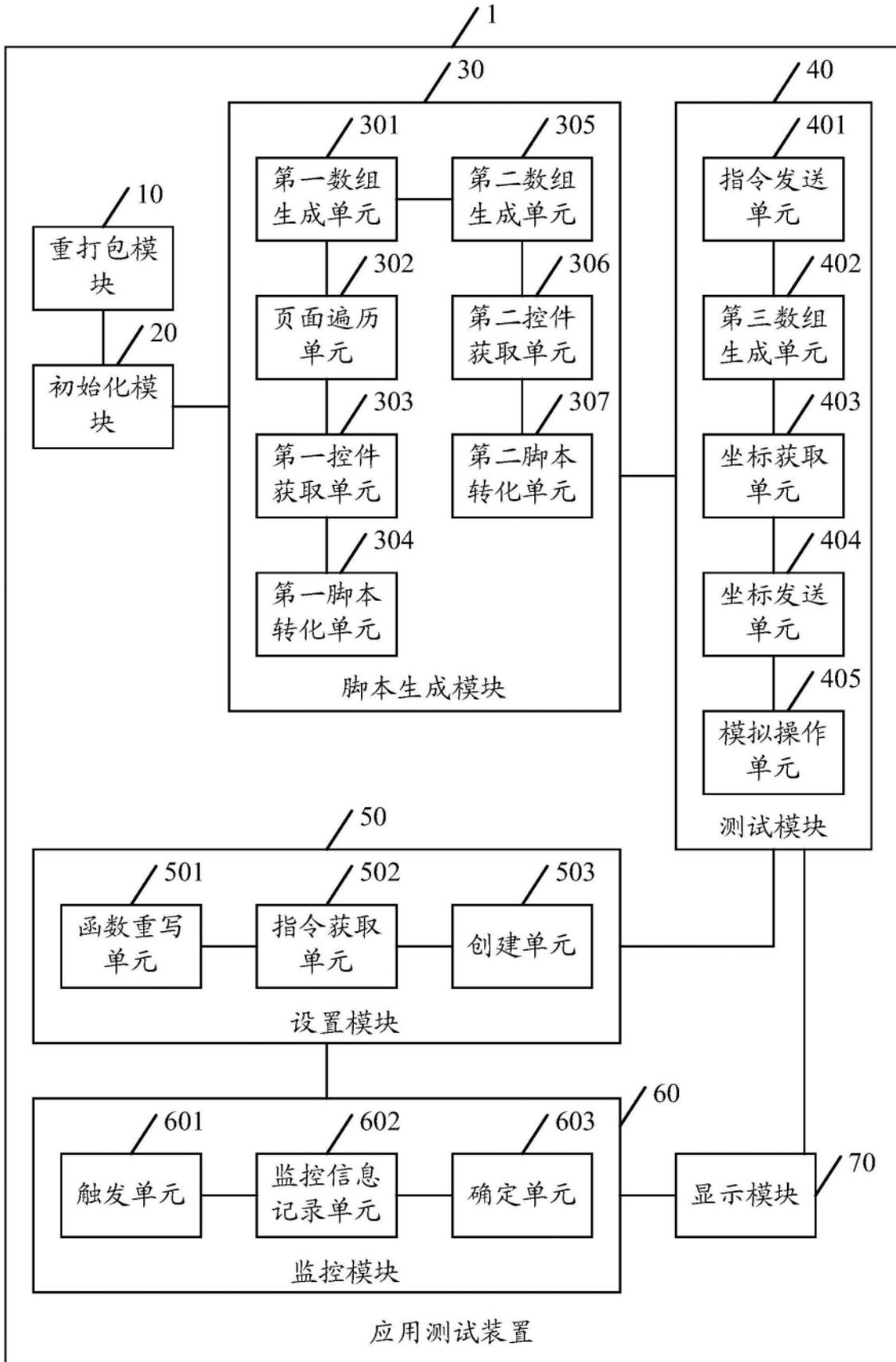


图7

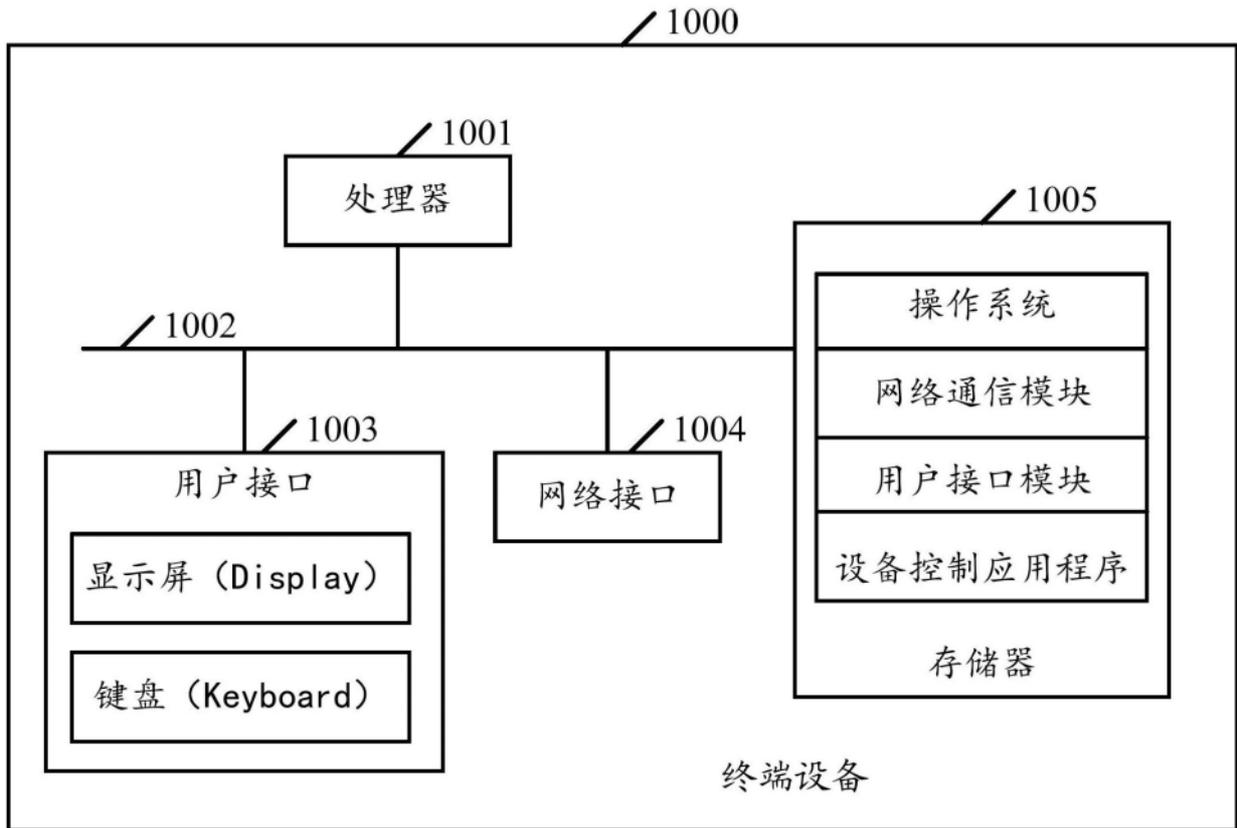


图8