



(12)发明专利

(10)授权公告号 CN 106649103 B

(45)授权公告日 2019.04.16

(21)申请号 201611056866.X

(22)申请日 2016.11.25

(65)同一申请的已公布的文献号

申请公布号 CN 106649103 A

(43)申请公布日 2017.05.10

(73)专利权人 深圳大学

地址 518061 广东省深圳市南山区深圳大学南校区计软大楼724

(72)发明人 蔡树彬 明仲 周小婷 卢亚辉

(74)专利代理机构 深圳市君胜知识产权代理事务所(普通合伙) 44268

代理人 王永文 刘文求

(51)Int.Cl.

G06F 11/36(2006.01)

(56)对比文件

CN 103336742 A,2013.10.02,

CN 106021102 A,2016.10.12,

CN 104679644 A,2015.06.03,

US 2014211021 A1,2014.07.31,

明仲,尹剑飞,杨巍等.一种Web系统性能测试框架及其混合建模过程.《计算机研究与发展》.2010,第47卷(第7期),

审查员 方源

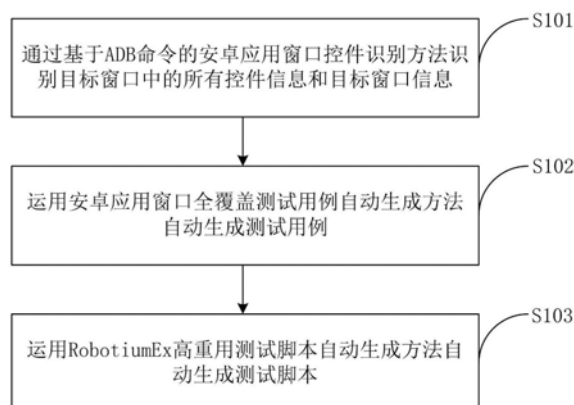
权利要求书3页 说明书11页 附图3页

(54)发明名称

一种安卓应用程序自动化黑盒测试的方法及系统

(57)摘要

本发明公开了一种安卓应用程序自动化黑盒测试的方法及系统,识别目标窗口中的所有控件信息和目标窗口信息时,在获取控件信息时只需解析通过ADB命令获得的控件信息文档即可得到所有控件信息,并且通过ADB命令可直接获取窗口信息,过程简单;自动生成测试用例时,可在自动生成测试用例过程中,对安卓应用窗口全覆盖;自动生成测试脚本时,通过对Robotium自动化测试框架封装,只需在测试用例文档中添加关键字和测试数据,即可实现自动生成测试脚本,当测试数据发生变化时,只需修改测试用例文档中的测试数据即可;本发明在不需要程序源码的情况下,提高了测试的自动化程度、界面的覆盖率及测试脚本的可重用性。



1. 一种安卓应用程序自动化黑盒测试的方法,其特征在于,该安卓应用程序自动化黑盒测试的方法包括以下步骤:步骤一,通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息;步骤二,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例;步骤三,运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本;

步骤一,通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息包括以下步骤:步骤(1),通过ADB向设备分别发送dump和adb shell dumpsys window w|findstr mCurrent命令;步骤(2),通过执行dump命令获得保存当前设备屏幕上的窗口控件信息的xml文件,通过执行adb shell dumpsys window w|findstr mCurrent命令获得当前窗口所属的Activity名和窗口哈希码;步骤(3),把获得的xml文件抓取到本地主机上,对其进行解析,并把获得的当前窗口所属的Activity名和窗口哈希码信息加入控件信息中,生成控件节点列表;步骤(4),把生成的控件节点列表重新写入一个xml文件中。

2. 如权利要求1所述的安卓应用程序自动化黑盒测试的方法,其特征在于,步骤一,通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息时,控件信息包括控件类型、控件名称和控件属性,窗口信息包括窗口所在Activity名称和窗口哈希码。

3. 如权利要求1所述的安卓应用程序自动化黑盒测试的方法,其特征在于,步骤二,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例时,需要构建一个基于安卓应用窗口的遍历模型,遍历模型为一个有向图,图节点表示一个窗口节点,而图的边表示一个模拟动作,在基于窗口的遍历模型中,窗口标识(Window Flag)用于唯一标识一个窗口,在窗口标识中保存了Activity名称和窗口哈希码信息,可通过分别对Activity名称和窗口哈希码进行对比来判断两个窗口是否相同;任务(Task)可表示为窗口标识,动作,控件节点这样一个三元组,其中窗口标识表示一个动作在此窗口下执行,动作表示一个待模拟的操作,控件节点表示待模拟操作所属控件,只有当应用程序运行到此窗口且此窗口上的动作被执行后,一个任务才算被执行;窗口节点(Window Node)中保存了与窗口相关的所有信息,其中包括对应的窗口标识、当前窗口中包含的所有待执行的任务形成的任务列表、父节点、子节点列表、从起始窗口节点到当前窗口节点的路径上执行的所有任务列表以及当前窗口中的转换列表;转换(Transition)用于表示模型中的一个转换关系,它保存了任务和窗口节点信息,其中任务表示当前窗口节点中的一个任务,而节点表示任务执行后所产生的子节点;运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例包括以下步骤:步骤(1),通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息,根据控件类型为每个控件生成一个任务,形成任务列表,根据窗口信息生成窗口标识,并把窗口标识加入窗口节点中;步骤(2),如果窗口节点的任务列表不为空,依次从任务列表中取出一个任务,并根据窗口标识判断当前窗口是否为执行任务所需要的窗口,如果是所需窗口,则直接执行任务,如果不是所需窗口,则先恢复到当前任务所需要的窗口,然后执行任务;步骤(3),判断执行这个任务后窗口是否发生切换;步骤(4),如果发生切换,则判断是否跳转到其他应用的窗口中,如果跳转到其它应用的窗口中,则重启应用程序,循环步骤(1)、步骤(2)、步骤(3),否则把当前任务加入到任务

执行路径列表中,循环步骤(1),获取跳转后的窗口节点信息,把新窗口节点信息和当前执行的任务加入到当前转换中,把当前转换添加到当前窗口节点的转换列表中,同时把前一个窗口节点添加为新窗口节点的父窗口节点;步骤(5),如果没有发生切换,循环步骤(2)、步骤(3);步骤(6),判断当前窗口是否为结束窗口,即点击这个窗口上的任何控件都不会切换到新的窗口,如果为结束窗口则根据任务执行路径列表生成一个测试用例,同时删除任务执行路径列表中最后一个任务;步骤(7),判断是否遍历完整个应用程序,如果已经遍历完整个应用程序,则保存整个遍历结果,如果没有遍历完整个应用程序,则返回上一个窗口,递归执行前面六个步骤,从而实现对整个应用进行遍历。

4.如权利要求3所述的安卓应用程序自动化黑盒测试的方法,其特征在于,在步骤(2)中,恢复到当前任务所需窗口的过程为:先从任务中获取任务执行所需要的目标窗口标识,然后获取应用程序当前窗口标识,如果目标窗口标识和当前窗口标识相同则不做处理,如果不同则模拟返回操作返回前一个窗口,再判断返回后的窗口标识是否为目标窗口标识;重复上述过程,直到找到目标窗口为止;如果退回到应用程序的起始窗口还没有找到目标窗口,则从目标窗口节点中获取保存的已经执行的任务列表,从起始窗口开始,依次执行任务列表中的任务,直到运行到目标窗口。

5.如权利要求3所述的安卓应用程序自动化黑盒测试的方法,其特征在于,在步骤(6)中,生成的测试用例保存在Excel文档中,测试用例的内容包括测试用例名称、操作步骤、当前Activity名称、目标Activity名称、被测控件的相关信息、操作关键字、结果检查关键字、输入数据和预期输出数据字段。

6.如权利要求1所述的安卓应用程序自动化黑盒测试的方法,其特征在于,在步骤二中,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例时,可使用<Activity名称、当前窗口截图>替代<Activity名称、窗口哈希码>来唯一标识一个安卓应用窗口。

7.如权利要求1所述的安卓应用程序自动化黑盒测试的方法,其特征在于,在步骤三中,运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本时,首先对Robotium框架中Solo类包含的常用方法进行封装,然后通过Excel文档中输入相应的关键字调用封装好的测试脚本生成方法,自动生成测试脚本;运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本包括以下步骤:步骤(1),对Robotium框架中Solo类包含的常用方法进行封装,得到常用的测试脚本生成方法;步骤(2),给封装好的测试脚本生成方法绑定相应的关键字,形成RobotiumEx自动化测试框架;步骤(3),运用安卓应用窗口全覆盖测试用例自动生成方法生成Excel测试用例文档;步骤(4),在生成的测试用例文档中添加相应的测试数据和关键字;步骤(5),读取添加关键字和测试数据后的测试用例,获取测试用例对应的所有关键字,依次通过关键字调用相应的测试脚本生成方法,为测试用例自动生成测试脚本。

8.一种安卓应用程序自动化黑盒测试的系统,其特征在于,该安卓应用程序自动化黑盒测试的系统包括:基于ADB命令的安卓应用窗口控件识别模块,用于获取当前窗口的控件信息和窗口信息,控件信息包括控件类型、控件名称和控件属性,窗口信息包括窗口所在Activity名称和窗口哈希码;安卓应用窗口全覆盖测试用例自动生成模块,与所述基于ADB命令的安卓应用窗口控件识别模块相连接,用于根据深度优先遍历方式,在自动生

成测试用例的过程中对安卓应用窗口进行全覆盖；RobotiumEx高重用测试脚本自动生成模块，与所述安卓应用窗口全覆盖测试用例自动生成模块相连接，用于对Robotium框架中Solo类包含的常用方法进行封装，并通过在Excel 文档中输入相应的关键字调用封装好的测试脚本生成方法，自动生成测试脚本。

一种安卓应用程序自动化黑盒测试的方法及系统

技术领域

[0001] 本发明属于安卓应用程序测试技术领域,尤其涉及一种安卓应用程序自动化黑盒测试的方法及系统。

背景技术

[0002] 窗口(window)为应用程序显示在设备屏幕上的整个图形界面,唯一标识当前界面。引起窗口变化的因素包括不同Activity间的相互切换、同一个Activity组件中不同布局控件的显示与隐藏等。

[0003] ADB(Android Debug Bridge)即Android调试桥,是Android系统中的一种命令行工具,可以利用ADB实现Android手机与PC的连接。

[0004] APK(Android Package)即Android安装包。

[0005] 随着移动互联网技术的快速发展,Android应用程序的开发数量也越来越多。为保证Android应用程序的质量,应用程序在正式发布到应用市场之前,需要对其进行测试。但是开发者只提供了APK应用包给应用市场,只能进行黑盒测试。同时,自动化黑盒测试也非常符合最近比较流行的移动云测试服务。开发者只需上传被测应用程序的APK文件,就可利用云服务平台的资源进行测试。因此,实现在无源码的情况下对Android应用程序进行自动化测试是非常有意义的。然而,现有的方法和技术仍然不能充分满足自动化和覆盖率方面的要求,主要存在的问题包括:其一,需要手动编写或者手动录制测试脚本,导致测试的自动化程度不高;其二,需先通过静态分析技术构建Activity转换图,不仅过程复杂,而且界面覆盖率不高。

[0006] Tanzirul Azim等人提出了使用目标和深度优先遍历的方式模拟用户操作与Android应用程序进行交互,从而实现对应用程序进行测试,具体的实现过程为:

[0007] (1) 对被测应用程序进行静态分析,构建基于Activity组件的遍历模型;

[0008] (2) 运用目标和深度优先遍历的方式,模拟用户操作与Android应用程序进行交互;

[0009] (3) 在遍历的过程中生成测试结果。

[0010] 该技术方案缺点:由于采用的是基于Activity组件的遍历模型,由于一个Activity可能包含多个不同的窗口,同一个Activity中的窗口可以相互切换,基于Activity的遍历模型其实并没有真正建立显示界面之间的切换关系,使得其对GUI的遍历不够充分,从而影响界面覆盖率。

发明内容

[0011] 本发明的目的在于提供一种安卓应用程序自动化黑盒测试的方法及系统,旨在解决现有技术提供的安卓应用程序的方法,需要手动编写或者手动录制测试脚本,导致测试的自动化程度不高,同时需先通过静态分析技术构建Activity转换图,过程复杂,界面覆盖率不高的问题。

[0012] 本发明是这样实现的,一种安卓应用程序自动化黑盒测试的方法,该安卓应用程序自动化黑盒测试的方法包括以下步骤:

[0013] 步骤一,通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息;

[0014] 步骤二,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例;

[0015] 步骤三,运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本。

[0016] 进一步,步骤一,通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息时,控件信息包括控件类型、控件名称和控件属性,窗口信息包括窗口所在Activity名称和窗口哈希码;

[0017] 通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息包括以下步骤:

[0018] 步骤(1),通过ADB向设备分别发送dump和adb shell dumpsys window w|findstr mCurrent命令;

[0019] 步骤(2),通过执行dump命令获得保存当前设备屏幕上的窗口控件信息的xml文件,通过执行adb shell dumpsys window w|findstr mCurrent命令获得当前窗口所属的Activity名和窗口哈希码;

[0020] 步骤(3),把获得的xml文件抓取到本地主机上,对其进行解析,并把获得的当前窗口所属的Activity名和窗口哈希码信息加入控件信息中,生成控件节点列表;

[0021] 步骤(4),把生成的控件节点列表重新写入一个xml文件中。

[0022] 进一步,步骤二,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例时,需要构建一个基于安卓应用窗口的遍历模型,遍历模型为一个有向图,图节点表示一个窗口节点,而图的边表示一个模拟动作,在基于窗口的遍历模型中,窗口标识(Window Flag)用于唯一标识一个窗口,在窗口标识中保存了Activity名称和窗口哈希码信息,可通过分别对Activity名称和窗口哈希码进行对比来判断两个窗口是否相同;任务(Task)可表示为(窗口标识,动作,控件节点)这样一个三元组,其中窗口标识表示一个动作在此窗口下执行,动作表示一个待模拟的操作,控件节点表示待模拟操作所属控件,只有当应用程序运行到此窗口且此窗口上的动作被执行后,一个任务才算被执行;窗口节点(Window Node)中保存了与窗口相关的所有信息,其中包括对应的窗口标识、当前窗口中包含的所有待执行的任务形成的任务列表、父节点、子节点列表、从起始窗口节点到当前窗口节点的路径上执行的所有任务列表以及当前窗口中的转换列表;转换(Transition)用于表示模型中的一个转换关系,它保存了任务和窗口节点信息,其中任务表示当前窗口节点中的一个任务,而节点表示任务执行后所产生的子节点;

[0023] 运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例包括以下步骤:

[0024] 步骤(1),通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息,根据控件类型为每个控件生成一个任务,形成任务列表,根据窗口信息生成窗口标识,并把窗口标识加入窗口节点中;

[0025] 步骤(2),如果窗口节点的任务列表不为空,依次从任务列表中取出一个任务,并根据窗口标识判断当前窗口是否为执行任务所需要的窗口,如果是所需窗口,则直接执行

任务,如果不是所需窗口,则先恢复到当前任务所需要的窗口,然后执行任务;

[0026] 步骤(3),判断执行这个任务后窗口是否发生切换;

[0027] 步骤(4),如果发生切换,则判断是否跳转到其他应用的窗口中,如果跳转到其它应用的窗口中,则重启应用程序,循环步骤(1)、步骤(2)、步骤(3),否则把当前任务加入到任务执行路径列表中,循环步骤(1),获取跳转后的窗口节点信息,把新窗口节点信息和当前执行的任务加入到当前转换中,把当前转换添加到当前窗口节点的转换列表中,同时把前一个窗口节点添加为新窗口节点的父窗口节点;

[0028] 步骤(5),如果没有发生切换,循环步骤(2)、步骤(3);

[0029] 步骤(6),判断当前窗口是否为结束窗口,即点击这个窗口上的任何控件都不会切换到新的窗口,如果为结束窗口则根据任务执行路径列表生成一个测试用例,同时删除任务执行路径列表中最后一个任务;

[0030] 步骤(7),判断是否遍历完整个应用程序,如果已经遍历完整个应用程序,则整个遍历结果,如果没有遍历完整个应用程序,则返回上一个窗口,递归执行前面六个步骤,从而实现对整个应用进行遍历。

[0031] 在步骤(2)中,恢复到所需窗口的过程为:

[0032] 先从任务中获取任务执行所需要的目标窗口标识,然后获取应用程序当前窗口标识,如果目标窗口标识和当前窗口标识相同则不做处理,如果不同则模拟返回操作返回前一个窗口,再判断返回后的窗口标识是否为目标窗口标识;重复上述过程,直到找到目标窗口为止;如果退回到应用程序的起始窗口还没有找到目标窗口,则从目标窗口节点中获取保存的已经执行的任务列表,从起始窗口开始,依次执行任务列表中的任务,直到运行到目标窗口。

[0033] 在步骤(6)中,生成的测试用例保存在Excel文档中,测试用例的内容包括测试用例名称、操作步骤、当前Activity名称、目标Activity名称、被测控件的相关信息、操作关键字、结果检查关键字、输入数据和预期输出数据字段。

[0034] 进一步,在步骤二中,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例时,可使用<Activity名称、当前窗口截图>替代<Activity名称、窗口哈希码>来唯一标识一个安卓应用窗口。

[0035] 进一步,在步骤三中,运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本时,首先对Robotium框架中Solo类包含的常用方法进行封装,然后通过Excel文档中输入相应的关键字调用封装好的测试脚本生成方法,自动生成测试脚本;

[0036] 运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本包括以下步骤:

[0037] 步骤(1),对Robotium框架中Solo类包含的常用方法进行封装,得到常用的测试脚本生成方法;

[0038] 步骤(2),给封装好的测试脚本生成方法绑定相应的关键字,形成RobotiumEx自动化测试框架;

[0039] 步骤(3),运用安卓应用窗口全覆盖测试用例自动生成方法生成Excel测试用例文档;

[0040] 步骤(4),在生成的测试用例文档中添加相应的测试数据和关键字;

[0041] 步骤(5),读取添加关键字和测试数据后的测试用例,获取测试用例对应的所有关

键字,依次通过关键字调用相应的测试脚本生成方法,为测试用例自动生成测试脚本。

[0042] 本发明的另一目的在于提供一种安卓应用程序自动化黑盒测试的系统,该安卓应用程序自动化黑盒测试的系统包括:

[0043] 基于ADB命令的安卓应用窗口控件识别模块,用于获取当前窗口的控件信息和窗口信息,控件信息包括控件类型、控件名称和控件属性,窗口信息包括窗口所在Activity名称和窗口哈希码;

[0044] 安卓应用窗口全覆盖测试用例自动生成模块,与所述基于ADB命令的安卓应用窗口控件识别模块相连接,用于根据深度优先遍历方式,在自动生成测试用例的过程中对安卓应用窗口进行全覆盖;

[0045] RobotiumEx高重用测试脚本自动生成模块,与所述安卓应用窗口全覆盖测试用例自动生成模块相连接,用于对Robotium框架中Solo类包含的常用方法进行封装,并通过在Excel文档中输入相应的关键字调用封装好的测试脚本生成方法,自动生成测试脚本。

[0046] 本发明提供的安卓应用程序自动化黑盒测试的方法及系统,首先通过基于ADB命令的安卓应用窗口控件识别方法,识别目标窗口中的所有控件信息和目标窗口信息;然后运用安卓应用窗口全覆盖测试用例自动生成方法,自动生成测试用例;最后运用RobotiumEx高重用测试脚本自动生成方法,自动生成测试脚本;在通过基于ADB命令的安卓应用窗口控件识别方法,识别目标窗口中的所有控件信息和目标窗口信息时,在获取控件信息时只需解析通过ADB命令获得的控件信息文档即可得到所有控件信息,并且通过ADB命令可以直接获取窗口信息,过程简单并且节约时间;在运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例时,可在自动生成测试用例的过程中,对安卓应用窗口进行全覆盖;在运用RobotiumEx高重用测试脚本自动生成方法,自动生成测试脚本时,通过对Robotium自动化测试框架进行封装,只需在测试用例文档中添加关键字和测试数据,即可实现自动生成测试脚本,并且当测试数据发生变化时,只需修改测试用例文档中的测试数据即可,而不需修改测试脚本,从而提高了测试脚本的可重用性;本发明达到了在不需要程序源码的情况下,提高测试的自动化程度、界面覆盖率和提高测试脚本的可重用性的目的。

附图说明

[0047] 图1是本发明实施例提供的安卓应用程序自动化黑盒测试的方法的实现流程图;

[0048] 图2是本发明实施例提供的基于ADB命令的安卓应用窗口控件识别的实现流程图;

[0049] 图3是本发明实施例提供的安卓应用窗口全覆盖测试用例自动生成的实现流程图;

[0050] 图4是本发明实施例提供的SCLICK关键字封装的实现流程图;

[0051] 图5是本发明实施例提供的安卓应用程序自动化黑盒测试的系统的结构框图。

[0052] 图中:51、基于ADB命令的安卓应用窗口控件识别模块;52、安卓应用窗口全覆盖测试用例自动生成模块;53、RobotiumEx高重用测试脚本自动生成模块。

具体实施方式

[0053] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于

限定本发明。

[0054] 如图1所示,本发明是这样实现的,一种安卓应用程序自动化黑盒测试的方法,该安卓应用程序自动化黑盒测试的方法包括以下步骤:

[0055] 步骤S101,通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息;

[0056] 步骤S102,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例;

[0057] 步骤S103,运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本。

[0058] 在本发明实施例中,步骤S101,通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息时,控件信息包括控件类型、控件名称和控件属性,窗口信息包括窗口所在Activity名称和窗口哈希码;

[0059] 如图2所示,通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息包括以下步骤:

[0060] 步骤(1),通过ADB向设备分别发送dump和adb shell dumpsys window w|findstr mCurrent命令;

[0061] 步骤(2),通过执行dump命令获得保存当前设备屏幕上的窗口控件信息的xml文件,通过执行adb shell dumpsys window w|findstr mCurrent命令获得当前窗口所属的Activity名和窗口哈希码;

[0062] 步骤(3),把获得的xml文件抓取到本地主机上,对其进行解析,并把获得的当前窗口所属的Activity名和窗口哈希码信息加入控件信息中,生成控件节点列表;

[0063] 步骤(4),把生成的控件节点列表重新写入一个xml文件中。

[0064] 在本发明实施例中,步骤S102,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例时,需要构建一个基于安卓应用窗口的遍历模型,遍历模型为一个有向图,图节点表示一个窗口节点,而图的边表示一个模拟动作,在基于窗口的遍历模型中,窗口标识(Window Flag)用于唯一标识一个窗口,在窗口标识中保存了Activity名称和窗口哈希码信息,可通过分别对Activity名称和窗口哈希码进行对比来判断两个窗口是否相同;任务(Task)可表示为(窗口标识,动作,控件节点)这样一个三元组,其中窗口标识表示一个动作在此窗口下执行,动作表示一个待模拟的操作,控件节点表示待模拟操作所属控件,只有当应用程序运行到此窗口且此窗口上的动作被执行后,一个任务才算被执行;窗口节点(Window Node)中保存了与窗口相关的所有信息,其中包括对应的窗口标识、当前窗口中包含的所有待执行的任务形成的任务列表、父节点、子节点列表、从起始窗口节点到当前窗口节点的路径上执行的所有任务列表以及当前窗口中的转换列表;转换(Transition)用于表示模型中的一个转换关系,它保存了任务和窗口节点信息,其中任务表示当前窗口节点中的一个任务,而节点表示任务执行后所产生的子节点;

[0065] 如图3所示,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例包括以下步骤:

[0066] 步骤(1),通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息,根据控件类型为每个控件生成一个任务,形成任务列表,根据窗口信息生成窗口标识,并把窗口标识加入窗口节点中;

[0067] 步骤(2),如果窗口节点的任务列表不为空,依次从任务列表中取出一个任务,并

根据窗口标识判断当前窗口是否为执行任务所需要的窗口,如果是所需窗口,则直接执行任务,如果不是所需窗口,则先恢复到当前任务所需要的窗口,然后执行任务;

[0068] 步骤(3),判断执行这个任务后窗口是否发生切换;

[0069] 步骤(4),如果发生切换,则判断是否跳转到其他应用的窗口中,如果跳转到其它应用的窗口中,则重启应用程序,循环步骤(1)、步骤(2)、步骤(3),否则把当前任务加入到任务执行路径列表中,循环步骤(1),获取跳转后的窗口节点信息,把新窗口节点信息和当前执行的任务加入到当前转换中,把当前转换添加到当前窗口节点的转换列表中,同时把前一个窗口节点添加为新窗口节点的父窗口节点;

[0070] 步骤(5),如果没有发生切换,循环步骤(2)、步骤(3);

[0071] 步骤(6),判断当前窗口是否为结束窗口,即点击这个窗口上的任何控件都不会切换到新的窗口,如果为结束窗口则根据任务执行路径列表生成一个测试用例,同时删除任务执行路径列表中最后一个任务;

[0072] 步骤(7),判断是否遍历完整个应用程序,如果已经遍历完整个应用程序,则整个遍历结果,如果没有遍历完整个应用程序,则返回上一个窗口,递归执行前面六个步骤,从而实现对整个应用进行遍历。

[0073] 在步骤(2)中,恢复到所需窗口的过程为:

[0074] 先从任务中获取任务执行所需要的目标窗口标识,然后获取应用程序当前窗口标识,如果目标窗口标识和当前窗口标识相同则不做处理,如果不同则模拟返回操作返回前一个窗口,再判断返回后的窗口标识是否为目标窗口标识;重复上述过程,直到找到目标窗口为止;如果退回到应用程序的起始窗口还没有找到目标窗口,则从目标窗口节点中获取保存的已经执行的任务列表,从起始窗口开始,依次执行任务列表中的任务,直到运行到目标窗口。

[0075] 在步骤(6)中,生成的测试用例保存在Excel文档中,测试用例的内容包括测试用例名称、操作步骤、当前Activity名称、目标Activity名称、被测控件的相关信息、操作关键字、结果检查关键字、输入数据和预期输出数据字段。

[0076] 在本发明实施例中,在步骤S102中,运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例时,可使用<Activity名称、当前窗口截图>替代<Activity名称、窗口哈希码>来唯一标识一个安卓应用窗口。

[0077] 在本发明实施例中,在步骤S103中,运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本时,首先对Robotium框架中Solo类包含的常用方法进行封装,然后通过Excel文档中输入相应的关键字调用封装好的测试脚本生成方法,自动生成测试脚本;

[0078] 运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本包括以下步骤:

[0079] 步骤(1),对Robotium框架中Solo类包含的常用方法进行封装,得到常用的测试脚本生成方法;

[0080] 步骤(2),给封装好的测试脚本生成方法绑定相应的关键字,形成RobotiumEx自动化测试框架;

[0081] 步骤(3),运用安卓应用窗口全覆盖测试用例自动生成方法生成Excel测试用例文档;

[0082] 步骤(4),在生成的测试用例文档中添加相应的测试数据和关键字;

[0083] 步骤(5),读取添加关键字和测试数据后的测试用例,获取测试用例对应的所有关键字,依次通过关键字调用相应的测试脚本生成方法,为测试用例自动生成测试脚本。

[0084] 如图5所示,本发明的另一目的在于提供一种安卓应用程序自动化黑盒测试的系统,该安卓应用程序自动化黑盒测试的系统包括:

[0085] 基于ADB命令的安卓应用窗口控件识别模块51,用于获取当前窗口的控件信息和窗口信息,控件信息包括控件类型、控件名称和控件属性,窗口信息包括窗口所在Activity名称和窗口哈希码;

[0086] 安卓应用窗口全覆盖测试用例自动生成模块52,与基于ADB命令的安卓应用窗口控件识别模块51相连接,用于根据深度优先遍历方式,在自动生成测试用例的过程中对安卓应用窗口进行全覆盖;

[0087] RobotiumEx高重用测试脚本自动生成模块53,与安卓应用窗口全覆盖测试用例自动生成模块52相连接,用于对Robotium框架中Solo类包含的常用方法进行封装,并通过在Excel文档中输入相应的关键字调用封装好的测试脚本生成方法,自动生成测试脚本。

[0088] 下面结合附图及具体实施例对本发明的应用原理作进一步描述。

[0089] 本发明实施例提供的安卓应用程序自动化黑盒测试的方法包含:基于ADB命令的安卓应用窗口控件识别方法、安卓应用窗口全覆盖测试用例自动生成方法和RobotiumEx高重用测试脚本自动生成方法三大部分,首先通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息;然后运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例;最后运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本。从而达到在不需要程序源码的情况下提高测试的自动化程度、界面覆盖率和提高测试脚本的可重用性的目的。

[0090] 本发明提出了一种安卓应用窗口全覆盖测试用例自动生成和RobotiumEx高重用测试脚本自动生成的自动化黑盒测试方法,该方法包含基于ADB命令的安卓应用窗口控件识别、安卓应用窗口全覆盖测试用例自动生成和RobotiumEx高重用测试脚本自动生成三部分。

[0091] 1) 基于ADB命令的安卓应用窗口控件识别

[0092] 该部分用于获取当前窗口的控件信息和窗口信息,控件信息包括控件类型、控件名称和控件属性等,窗口信息包括窗口所在Activity名称和窗口哈希码。基于ADB命令的安卓应用窗口控件识别过程如图2所示。

[0093] 主要步骤如下:

[0094] (1) 通过ADB向设备分别发送dump和adb shell dumpsys window w|findstr mCurrent命令;

[0095] (2) 通过执行dump命令获得保存当前设备屏幕上的窗口控件信息的xml文件,通过执行adb shell dumpsys window w|findstr mCurrent命令获得当前窗口所属的Activity名和窗口哈希码;

[0096] (3) 把获得的xml文件抓取到本地主机上,对其进行解析,并把获得的当前窗口所属的Activity名和窗口哈希码信息加入控件信息中,生成控件节点列表;

[0097] (4) 把生成的控件节点列表重新写入一个xml文件中。

[0098] 2) 安卓应用窗口全覆盖测试用例自动生成

[0099] 根据深度优先遍历思想,本发明提出了一种安卓应用窗口全覆盖测试用例自动生成方法,这种方法的特征为可以在自动生成测试用例的过程中对安卓应用窗口进行全覆盖。其中在运用这种方法自动生成测试用例的过程中需要构建一个基于安卓应用窗口的遍历模型,这个遍历模型为一个有向图,图节点表示一个窗口节点,而图的边表示一个模拟动作。

[0100] 为了更好地描述基于窗口的遍历模型,本发明对以下概念进行了定义:

[0101] 定义:窗口标识(Window Flag)用于唯一标识一个窗口,在窗口标识中保存了Activity名称和窗口哈希码信息,可以通过分别对Activity名称和窗口哈希码进行对比来判断两个窗口是否相同。

[0102] 定义:任务(Task)可表示为(窗口标识,动作,控件节点)这样一个三元组,其中窗口标识表示一个动作在此窗口下执行,动作表示一个待模拟的操作,控件节点表示待模拟操作所属控件,只有当应用程序运行到此窗口且此窗口上的动作被执行后,一个任务才算被执行。

[0103] 定义:窗口节点(Window Node)中保存了与窗口相关的所有信息,其中包括对应的窗口标识、当前窗口中包含的所有待执行的任务形成的任务列表、父节点、子节点列表、从起始窗口节点到当前窗口节点的路径上执行的所有任务列表以及当前窗口中的转换列表。

[0104] 定义:转换(Transition)用于表示模型中的一个转换关系,它保存了任务和窗口节点信息,其中任务表示当前窗口节点中的一个任务,而节点表示任务执行后所产生的子节点。

[0105] 安卓应用窗口全覆盖测试用例自动生成过程如图3所示。

[0106] 主要步骤如下:

[0107] (1)通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息,根据控件类型为每个控件生成一个任务,形成任务列表,根据窗口信息生成窗口标识,并把它们加入窗口节点中;

[0108] (2)如果窗口节点的任务列表不为空,依次从任务列表中取出一个任务,并根据窗口标识判断当前窗口是否为执行任务所需要的窗口,如果是所需窗口,则直接执行任务,如果不是所需窗口,则先恢复到当前任务所需要的窗口,然后执行任务;

[0109] 恢复到所需窗口的过程为:先从任务中获取任务执行所需要的目标窗口标识,然后获取应用程序当前窗口标识,如果目标窗口标识和当前窗口标识相同则不做处理,如果不同则模拟返回操作返回前一个窗口,再判断返回后的窗口标识是否为目标窗口标识。重复上述过程,直到找到目标窗口为止。如果退回到应用程序的起始窗口还没有找到目标窗口,则从目标窗口节点中获取保存的已经执行的任务列表,从起始窗口开始,依次执行任务列表中的任务,直到运行到目标窗口。

[0110] (3)判断执行这个任务后窗口是否发生切换;

[0111] (4)如果发生切换,则判断是否跳转到其他应用的窗口中,如果跳转到其它应用的窗口中,则重启应用程序,循环步骤(1)、(2)、(3)。否则把当前任务加入到任务执行路径列表中,循环步骤(1),获取跳转后的窗口节点信息,把新窗口节点信息和当前执行的任务加入到当前转换中,把当前转换添加到当前窗口节点的转换列表中,同时把前一个窗口节点添加为新窗口节点的父窗口节点;

[0112] (5) 如果没有发生切换,循环步骤(2)(3);

[0113] (6) 判断当前窗口是否为结束窗口,即点击这个窗口上的任何控件都不会切换到新的窗口,如果为结束窗口则根据任务执行路径列表生成一个测试用例,同时删除任务执行路径列表中最后一个任务;

[0114] (7) 判断是否遍历完整个应用程序,如果已经遍历完整个应用程序,则整个遍历结果。如果没有遍历完整个应用程序,则返回上一个窗口,递归执行前面六个步骤,从而实现对整个应用进行遍历。

[0115] 生成的测试用例保存在Excel文档中,测试用例的内容包括测试用例名称、操作步骤、当前Activity名称、目标Activity名称、被测控件的相关信息、操作关键字、结果检查关键字、输入数据和预期输出数据等字段。

[0116] (3) RobotiumEx高重用测试脚本自动生成

[0117] RobotiumEx高重用测试脚本自动生成方法首先对Robotium框架中Solo类包含的常用方法进行封装;然后通过Excel文档中输入相应的关键字调用封装好的测试脚本生成方法,自动生成测试脚本。

[0118] 主要步骤如下:

[0119] (1) 对Robotium框架中Solo类包含的常用方法进行封装,得到常用的测试脚本生成方法;

[0120] (2) 给封装好的测试脚本生成方法绑定相应的关键字,形成RobotiumEx自动化测试框架;

[0121] (3) 运用安卓应用窗口全覆盖测试用例自动生成方法生成Excel测试用例文档;

[0122] (4) 在生成的测试用例文档中添加相应的测试数据和关键字;

[0123] (5) 读取添加关键字和测试数据后的测试用例,获取测试用例对应的所有关键字,依次通过关键字调用相应的测试脚本生成方法,为测试用例自动生成测试脚本。

[0124] RobotiumEx自动化测试框架封装了操作关键字和结果检查关键字两种关键字,表1为所封装的部分关键字表。

[0125] 表1部分关键字表

[0126]

关键字	关键字类型	控件信息	输入值	验证数据	注释
SCLICK	操作关键字	View 值	空	无此项	短按点击
LCLICK	操作关键字	View 值	空	无此项	长按点击
INPUT	操作关键字	View 值	需输入的字符	无此项	输入
DRAG	操作关键字	View 值	空	无这项	拖拽
JUMP	结果检查关键字	空	无此项	目标跳转 Activity	验证 Activity 跳 转
INPUTCHECK	结果检查关键字	View 值	无此项	需要验证的字符	验证输入的字符 与预期是否相同
TOAST	结果检查关键字	空	无此项	Toast 提示内容	验证 Toast 的内 容是否正确显示

[0127] 下面以SCLICK关键字为例对关键字的封装过程进行详细介绍,SCLICK关键字的封装过程如图4所示。

[0128] 本发明技术方案带来的有益效果

[0129] (1) 由于一个Activity可能包含多个不同的窗口,同一个Activity中的窗口可以相互切换,基于Activity的遍历模型其实并没有真正建立显示界面之间的切换关系,因此安卓应用窗口全覆盖测试用例自动生成方法可以提高测试的界面覆盖率。

[0130] (2) 由于在使用Robotium编写测试脚本的时候,测试数据是包含在脚本中的,当测试数据发生变化时候,就需要修改测试脚本,代码重用性不高。而RobotiumEx高重用测试脚本自动生成方法在测试数据发生变化时候,只需修改测试用例文档中的测试数据即可,从而提高了测试脚本的可重用性。

[0131] 在运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例这一步骤中,可以使用<Activity名称,当前窗口截图>替代<Activity名称,窗口哈希码>来唯一标识一个安卓应用窗口。

[0132] 本发明根据深度优先遍历思想,提出了一种安卓应用窗口全覆盖测试用例自动生成方法,这种方法可以在自动生成测试用例的过程中对安卓应用窗口进行全覆盖;本发明对整个技术方案的改进:首先通过基于ADB命令的安卓应用窗口控件识别方法识别目标窗口中的所有控件信息和目标窗口信息,这种方法在获取控件信息时只需解析通过ADB命令获得的控件信息文档即可得到所有控件信息,并且通过ADB命令可以直接获取窗口信息,过程简单并且节约时间;然后运用安卓应用窗口全覆盖测试用例自动生成方法自动生成测试用例,这种方法可以在自动生成测试用例的过程中对安卓应用窗口进行全覆盖;最后运用RobotiumEx高重用测试脚本自动生成方法自动生成测试脚本,这种方法通过对Robotium自动化测试框架进行封装,只需在测试用例文档中添加关键字和测试数据,即可实现自动生成测试脚本,并且当测试数据发生变化时,只需修改测试用例文档中的测试数据即可,而不需修改测试脚本,从而提高了测试脚本的可重用性。

[0133] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明的保护范围之内。

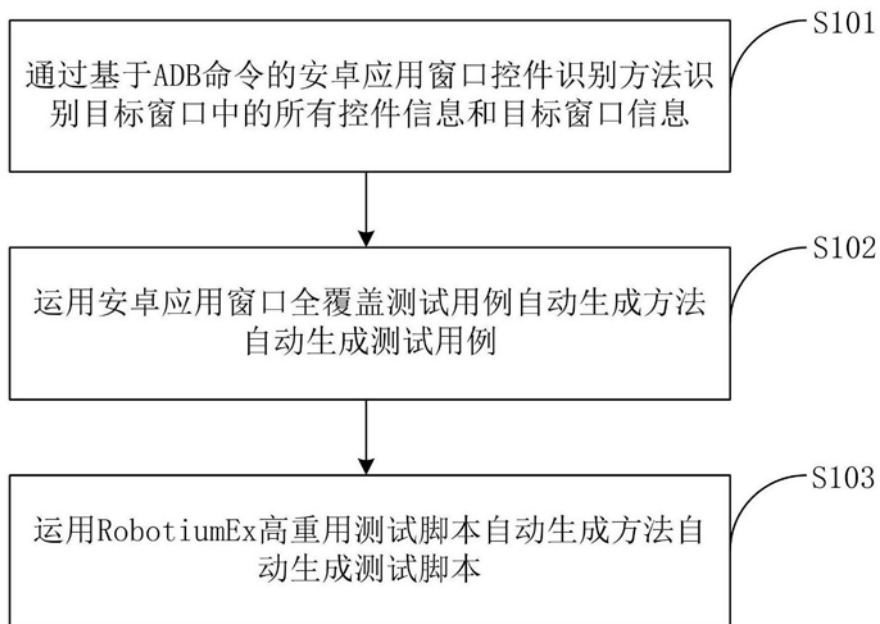


图1



图2

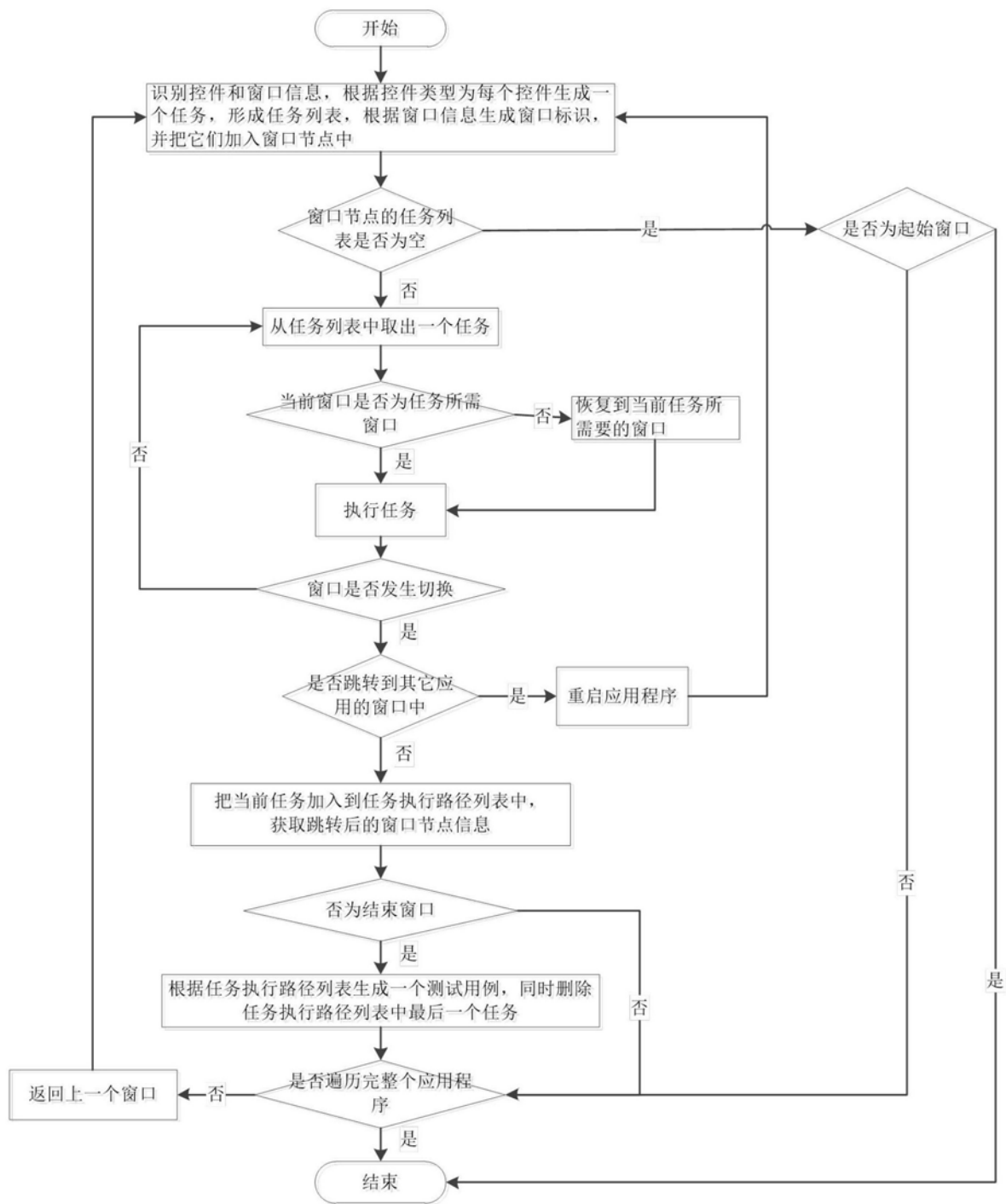


图3

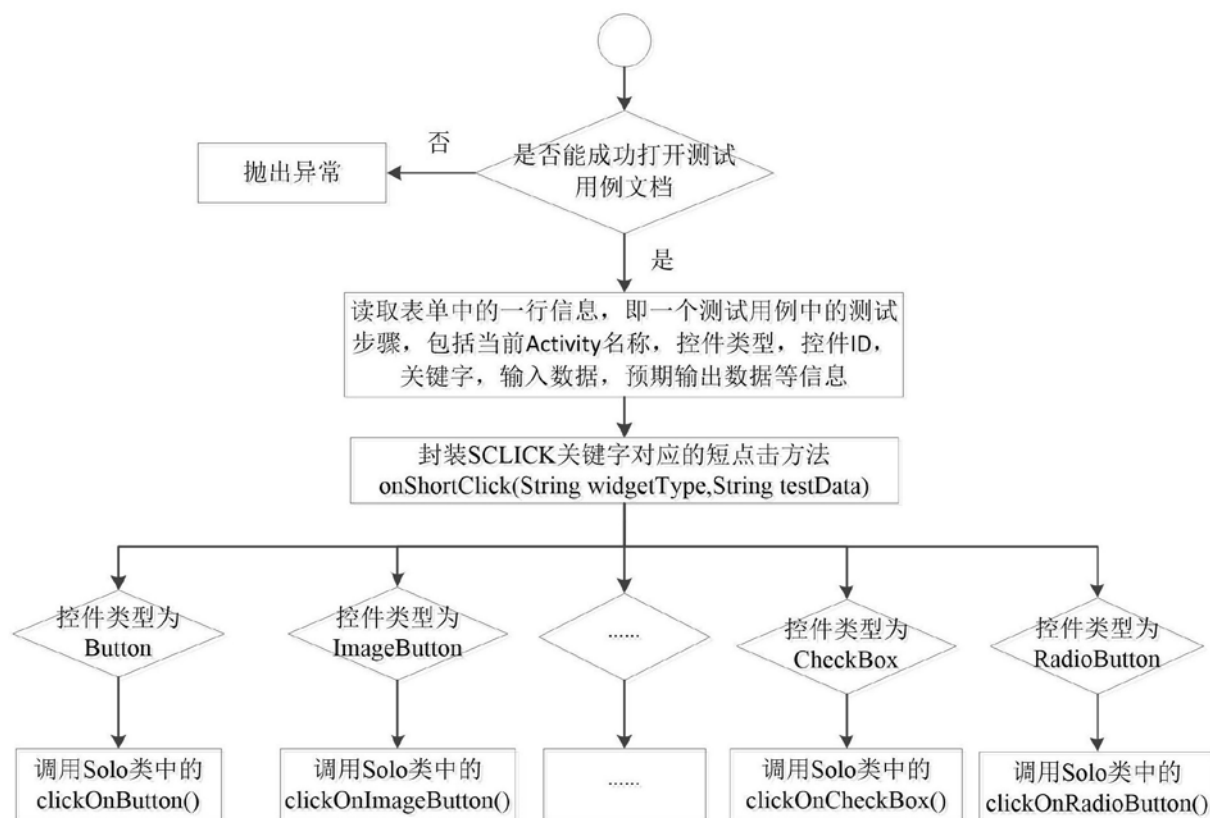


图4

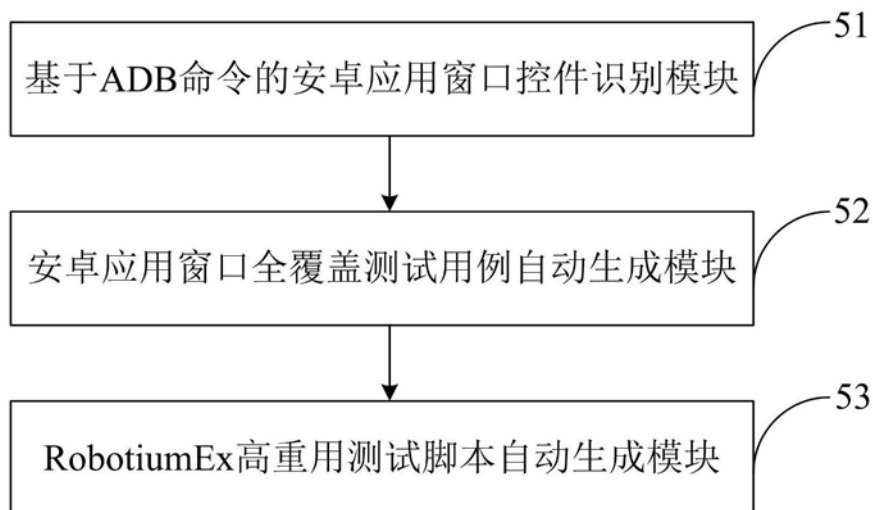


图5