



(12)发明专利申请

(10)申请公布号 CN 108536589 A

(43)申请公布日 2018.09.14

(21)申请号 201810253142.7

(22)申请日 2018.03.26

(71)申请人 广州小鹏汽车科技有限公司  
地址 510000 广东省广州市中新广州知识  
城九佛建设路333号245室

(72)发明人 刘坚 曾郁荣 单文龙 黄荣海  
夏珩

(74)专利代理机构 广州嘉权专利商标事务所有  
限公司 44205  
代理人 胡辉 郑泽萍

(51)Int.Cl.  
G06F 11/36(2006.01)  
G06F 11/34(2006.01)

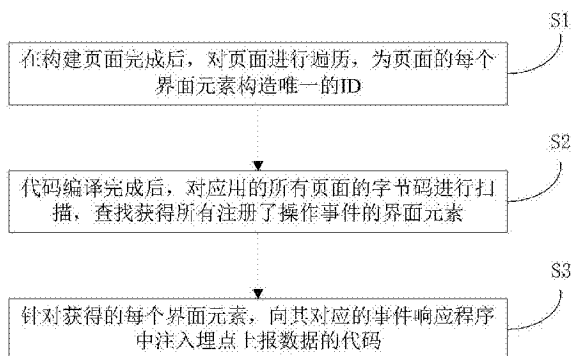
权利要求书1页 说明书6页 附图2页

(54)发明名称

一种应用程序埋点方法及系统

(57)摘要

本发明公开了一种应用程序埋点方法及系统,在构建页面完成后,对页面进行遍历,为页面的每个界面元素构造唯一的ID;代码编译完成后,对应用的所有页面的字节码进行扫描,查找获得所有注册了操作事件的界面元素;针对获得的每个界面元素,向其对应的事件响应程序中注入埋点上报数据的代码。本发明实现对界面元素的统一、自动化全埋点,在界面元素被触发后,在运行对应的事件响应程序同时,执行插入在事件响应程序中的埋点上报数据的代码,直接进行数据上报,可以保证数据上报的全面性,及时、快速、高效,可广泛应用于软件技术行业中。



1. 一种应用程序埋点方法,其特征在于,包括步骤:
  - S1、在构建页面完成后,对页面进行遍历,为页面的每个界面元素构造唯一的ID;
  - S2、代码编译完成后,对应用的所有页面的字节码进行扫描,查找获得所有注册了操作事件的界面元素;
  - S3、针对获得的每个界面元素,向其对应的事件响应程序中注入埋点上报数据的代码。
2. 根据权利要求1所述的应用程序埋点方法,其特征在于,还包括步骤:
  - S4、当检测到页面的界面元素被触发时,运行其对应的事件响应程序,并直接进行数据上报。
3. 根据权利要求1所述的应用程序埋点方法,其特征在于,所述步骤S1,具体包括:
  - S101、在构建页面完成后,对页面的所有界面元素构成的多叉树进行遍历,获得页面的每个界面元素的唯一路径;
  - S102、根据每个界面元素的唯一路径为其构造唯一的ID。
4. 根据权利要求3所述的应用程序埋点方法,其特征在于,所述步骤S102,其具体为:将每个界面元素的唯一路径所构成的字符串或者路径的MD5值,作为该界面元素对应的唯一的ID。
5. 根据权利要求1所述的应用程序埋点方法,其特征在于,所述步骤S1中,采用多叉树的广度优先遍历算法或者深度优先遍历算法对页面进行遍历,且遍历过程中,采用顺序号区分界面元素类型相同的来构建每个界面元素的唯一性ID。
6. 根据权利要求1所述的应用程序埋点方法,其特征在于,页面的界面元素包括页面的元素和控件。
7. 根据权利要求2所述的应用程序埋点方法,其特征在于,所述步骤S4,具体为:当检测到页面的界面元素被触发时,运行其对应的事件响应程序,显示对应的事件响应界面,同时执行插入在事件响应程序中的埋点上报数据的代码,实现埋点数据上报。
8. 根据权利要求1所述的应用程序埋点方法,其特征在于,所述步骤S1中,所述应用程序运行于Android系统,通过动态代理方式进行AOP切面编程的方法实现构造界面元素的唯一ID;  
或所述应用程序运行于iOS系统,通过对viewController进行hook编程,从而利用iOS系统的Runtime,使用method swizzling映射方法实现构造界面元素的唯一ID。
9. 根据权利要求8所述的应用程序埋点方法,其特征在于,所述步骤S3中,所述应用程序运行于Android系统,通过Java的Javassist库或ASM框架修改字节码,实现插桩编程,往操作事件的事件响应程序中注入埋点上报数据的代码;  
或所述应用程序运行于iOS系统,通过Cydia Substrate进行插桩编程注入埋点上报数据的代码。
10. 一种应用程序埋点系统,其特征在于,包括:
  - 至少一个处理器;
  - 至少一个存储器,用于存储至少一个程序;当所述至少一个程序被所述至少一个处理器执行,使得所述至少一个处理器实现如权利要求1-9任一项所述的应用程序埋点方法。

## 一种应用程序埋点方法及系统

### 技术领域

[0001] 本发明涉及软件技术领域,特别是涉及一种应用程序埋点方法及系统。

### 背景技术

[0002] 互联网业界通过采集网站与手机客户端程序数据,追踪问题,从而达到优化、改进产品的目的,其主要采用的是代码埋点上报数据方案,这种方案在检测到点击、切换、选择等操作事件发生时,才构建事件对应的界面元素的ID,然后在事件对应的核心业务中编写对应的埋点数据后上报,这种埋点方案的缺点是非常耗费开发者的时间,而且埋点业务与核心业务严重耦合,导致可扩展性差,也不便于后期维护,甚至会导致埋错位置,以及埋点所对应事件ID出现错乱的情况,另外,在事件发生时,才编写对应的埋点数据,容易出现卡顿等问题,容易出现数据漏报问题,无法保证数据上报的全面性。

### 发明内容

[0003] 为了解决上述的技术问题,本发明的目的是提供一种应用程序埋点方法及系统。

[0004] 本发明解决其技术问题所采用的技术方案是:

[0005] 一种应用程序埋点方法,包括步骤:

[0006] S1、在构建页面完成后,对页面进行遍历,为页面的每个界面元素构造唯一的ID;

[0007] S2、代码编译完成后,对应用的所有页面的字节码进行扫描,查找获得所有注册了操作事件的界面元素;

[0008] S3、针对获得的每个界面元素,向其对应的事件响应程序中注入埋点上报数据的代码。

[0009] 进一步,还包括步骤:

[0010] S4、当检测到页面的界面元素被触发时,运行其对应的事件响应程序,并直接进行数据上报。

[0011] 进一步,所述步骤S1,具体包括:

[0012] S101、在构建页面完成后,对页面的所有界面元素构成的多叉树进行遍历,获得页面的每个界面元素的唯一路径;

[0013] S102、根据每个界面元素的唯一路径为其构造唯一的ID。

[0014] 进一步,所述步骤S102,其具体为:

[0015] 将每个界面元素的唯一路径所构成的字符串或者路径的MD5值,作为该界面元素对应的唯一的ID。

[0016] 进一步,所述步骤S1中,采用多叉树的广度优先遍历算法或者深度优先遍历算法对页面进行遍历,且遍历过程中,采用顺序号区分界面元素类型相同的来构建每个界面元素的唯一性ID。

[0017] 进一步,页面的界面元素包括页面的元素和控件。

[0018] 进一步,所述步骤S4,具体为:

[0019] 当检测到页面的界面元素被触发时,运行其对应的事件响应程序,显示对应的事件响应界面,同时执行插入在事件响应程序中的埋点上报数据的代码,实现埋点数据上报。

[0020] 进一步,所述步骤S1中,

[0021] 所述应用程序运行于Android系统,通过动态代理方式进行AOP切面编程的方法实现构造界面元素的唯一ID;

[0022] 或所述应用程序运行于iOS系统,通过对viewController进行hook编程,从而利用iOS系统的Runtime,使用method swizzling映射方法实现构造界面元素的唯一ID。

[0023] 进一步,所述步骤S3中,

[0024] 所述应用程序运行于Android系统,通过Java的Javassist库或ASM框架修改字节码,实现插桩编程,往操作事件的事件响应程序中注入埋点上报数据的代码;

[0025] 或所述应用程序运行于iOS系统,通过Cydia Substrate进行插桩编程注入埋点上报数据的代码。

[0026] 本发明解决其技术问题所采用的另一技术方案是:

[0027] 一种应用程序埋点系统,包括:

[0028] 至少一个处理器;

[0029] 至少一个存储器,用于存储至少一个程序;

[0030] 当所述至少一个程序被所述至少一个处理器执行,使得所述至少一个处理器实现所述的应用程序埋点方法。

[0031] 本发明方法、系统的有益效果是:本发明通过遍历页面,获得页面中注册了操作事件的界面元素,然后通过向注册了操作事件的界面元素对应的事件响应程序中注入埋点上报数据的代码,实现对界面元素的统一、自动化全埋点,从而在界面元素被触发后,在运行对应的事件响应程序的同时,执行插入在事件响应程序中的埋点上报数据的代码,直接进行数据上报,无需实时构建界面元素ID,无需手动编写埋点程序,避免了出现卡顿或数据漏报的问题,可以保证数据上报的全面性,及时、快速、高效。

## 附图说明

[0032] 图1是本发明的一种应用程序埋点方法的流程图;

[0033] 图2是本发明具体实施例中页面的多叉树以及节点路径的示意图;

[0034] 图3是本发明具体实施例中页面的多叉树以及节点MD5值的示意图;

[0035] 图4是本发明的一种应用程序埋点系统的结构框图。

## 具体实施方式

[0036] 方法实施例

[0037] 参照图1,本实施例提供了一种应用程序埋点方法,包括步骤:

[0038] S1、在构建页面完成后,对页面进行遍历,为页面的每个界面元素构造唯一的ID;

[0039] S2、代码编译完成后,对应用的所有页面的字节码进行扫描,查找获得所有注册了操作事件的界面元素;

[0040] S3、针对获得的每个界面元素,向其对应的事件响应程序中注入埋点上报数据的代码。

[0041] 进一步作为优选的实施方式,还包括步骤:

[0042] S4、当检测到页面的界面元素被触发时,运行其对应的事件响应程序,并直接进行数据上报。

[0043] 所述操作事件包括点击、切换、选择等事件,指页面所能实现的所有操作或响应。

[0044] 本发明所称页面,包括网页页面、Android手机页面、iOS手机页面等,这些页面具有相同特性:网页布局文件html、Android布局文件xml文件、iOS布局文件xml文件(包括storyboard和xib文件)都是DOM Tree树状结构。因此,可以用界面元素在DOM Tree中的路径来标识界面元素,构造唯一的ID。

[0045] 本方案通过向注册了操作事件例如点击、切换、选择等事件的界面元素对应的事件响应程序中注入埋点上报数据的代码,实现对界面元素的统一、自动化全埋点,从而在界面元素被触发例如被点击、切换或选择后,在运行对应的事件响应程序同时,执行插入在事件响应程序中的埋点上报数据的代码,直接进行数据上报,无需实时构建界面元素ID,无需手动编写埋点程序,避免了出现卡顿或数据漏报的问题,可以保证数据上报的全面性,及时、快速、高效。

[0046] 进一步作为优选的实施方式,所述步骤S1,具体包括:

[0047] S101、在构建页面完成后,对页面的所有界面元素构成的多叉树进行遍历,获得页面的每个界面元素的唯一路径;

[0048] S102、根据每个界面元素的唯一路径为其构造唯一的ID。

[0049] 进一步作为优选的实施方式,所述步骤S102,其具体为:

[0050] 将每个界面元素的唯一路径所构成的字符串或者路径的MD5值,作为该界面元素对应的唯一的ID。

[0051] 简单对每个界面元素设置ID,可能会出现重复或者出错问题,以Android平台为例,在布局文件XML中设置的ID会跟随系统版本的不同而产生不同的数字ID,会导致数据上报ID不统一问题。因此本发明通过对页面的多叉树进行遍历,根据路径来构造界面元素的唯一ID,因为每个界面元素在多叉树中的路径是唯一的,本方案可以保证所构造ID的唯一性。另外,本方案的路径因为包含了界面元素所处的页面的名字,例如网页的html文件名,Android客户端的Activity、Fragment等,也确保了每个界面元素ID的唯一性。

[0052] 进一步作为优选的实施方式,所述步骤S1中,采用多叉树的广度优先遍历算法或者深度优先遍历算法对页面进行遍历,且遍历过程中,采用顺序号区分界面元素类型相同的来构建每个界面元素的唯一性ID。

[0053] 进一步作为优选的实施方式,页面的界面元素包括页面的元素和控件。

[0054] 进一步作为优选的实施方式,所述步骤S4,具体为:

[0055] 当检测到页面的界面元素被触发时,运行其对应的事件响应程序,显示对应的事件响应界面,同时执行插入在事件响应程序中的埋点上报数据的代码,实现埋点数据上报。这里,所称的界面元素被触发,包括被点击、选择、切换等。

[0056] 进一步作为优选的实施方式,所述步骤S1中,Android系统通过动态代理方式进行AOP切面编程的方法实现构造界面元素的唯一ID;iOS系统通过对viewController进行hook编程,从而利用iOS系统的Runtime,使用method swizzling映射方法实现构造界面元素的唯一ID。

[0057] 如果是在界面元素被点击、选择、切换等操作后,在事件响应时才对页面的多叉树进行遍历,生成界面元素的唯一ID,然后生成埋点数据,每次操作界面元素时再执行遍历耗时过程,会导致页面卡顿,体验性很差等问题。为了不影响网页和客户端程序的性能,本方案采用在页面创建的时候构造该页面中的所有界面元素的唯一ID,从而步骤S3根据该唯一ID直接向界面元素对应的事件响应程序中注入埋点上报数据的代码,及时而高效。以Android客户端为例,通过对Android Frameworks进行AOP切面编程的方式实现对Android系统的应用框架进行改造,从而在页面创建的同时遍历页面中所有View构成的多叉树,完成每个界面元素ID的构造。因为软件开发大都采用MVC、MVP、MVVM架构,所以网站网页、iOS客户端同样也适用。

[0058] 进一步作为优选的实施方式,所述步骤S3中,Android系统通过Java的Javassist库或ASM框架修改字节码,实现插桩编程,往操作事件的事件响应程序中注入埋点上报数据的代码;iOS系统通过Cydia Substrate进行插桩编程注入埋点上报数据的代码。

[0059] 以Android客户端为例,通过Java的Javassist库或ASM框架,在编译出字节码文件后,对页面的所有字节码进行扫描,找出所有继承自Activity、Fragment,且实现了点击等操作回调接口的类,这些类即为注册了操作事件(例如点击、切换、选择等事件)的类,通过AOP切面编程方式,在所有继承自Activity、Fragment的类需要追踪的事件响应程序(一般选择页面出现的声明周期方法、页面消息的声明周期方法)中实现插桩编程,往操作事件的事件响应程序中注入埋点上报数据的代码,来跟踪页面跳转与切换行为。同理,往所有实现了点击接口的类的点击事件响应程序中注入埋点数据上报代码,来实现所有控件点击事件的无代码全埋点上报数据。通过以上方法修改字节码之后,将修改后的字节码成功打包到APK中,就实现了无代码埋点上报数据,能详细追踪到客户端的使用情况、页面切换情况、页面停留时长情况等。因此,只要点击了页面的任意界面元素,都会有对应的埋点数据上报,保证了上报数据的全面性。

[0060] 方法详细实施例

[0061] 本实施例是前述方法实施例的一详细例子,具体包括以下步骤:

[0062] 一、构造元素、控件的ID

[0063] 在网站、Android客户端、iOS客户端中,所有页面的布局,其实都是元素或控件构成的一个多叉树(DOM Tree)。每个元素或者控件在多叉树中的路径是唯一的。以Android客户端为例(iOS原理相同),本方案采用从根节点到元素或者控件构成的路径字符串作为ID,并且控件类型相同的兄弟元素或者控件的之间采用顺序编号来区分,例如图2的多叉树结构中,根节点类型为RootView,RootView节点的子节点类型为LinearLayout和RelativeLayout,其中LinearLayout的四个子节点的类型依次为:Button、Button、TextVie和Button。图2中显示了多叉树的所有节点的路径,其中,最下层3个Button节点的路径通过顺序编号[0]、[1]、[2]来区分。

[0064] 此外,图2中各节点的也可以使用路径构建的MD5值来作为ID,如图3所示,每个节点的ID均由路径的MD5值表示。

[0065] 当一个页面有两个Button,都注册了点击Click事件,在编译出字节码后,通过修改字节码方式在点击事件响应程序中添加了数据上报代码。在页面加载创建的时候,为两个Button分别构建唯一的ID,通过setTag的方式保存在Button实例中。

[0066] 优选的,本实施例中,构建元素、控件等界面元素的唯一ID时,除了控件在二叉树中的路径外,将控件所处的页面(Activity、Fragment)名称,控件的内容也作为控件ID的一部分,例如图2中页面为ExampleActivity中的布局时,则左起第一个Button的ID为:

[0067] ExampleActivity/RootView/LinearLayout/Button[0]+Button[0].getText();

[0068] 将页面名称、控件内容也作为ID的一部分,确保了每个控件ID的唯一性。

[0069] 二、在页面加载创建时进行编程

[0070] 1)、Android可以通过动态代理等方式实现AOP切面编程。具体步骤如下:

[0071] 首先,布局加载创建器PhoneLayoutInflater;其次,执行布局加载方法,即inflate方法,包括int resource,ViewGroup root,boolean attachToRoot等参数;然后,通过AOP切面编程方式,在其中注入构造界面元素的唯一ID的代码;最后,完成页面构建,实现界面元素ID的构造。

[0072] 2)、iOS可以对viewController进行hook编程,利用iOS的Runtime,使用method swizzling拦截页面本身的代码,用自定义代码替换页面本身的代码,构造界面元素的唯一ID。

[0073] 三、修改代码,插入埋点上报数据的代码

[0074] 1)、Android实现

[0075] 通过Java的Javassist库或ASM框架修改字节码,注入埋点数据上报代码。Android编译出class文件之后的打包的过程,具体如下:编译出class文件之后,遍历class文件,通过Javassist库或ASM框架注入埋点数据上报代码,生成dex文件,然后打包成apk安装包。

[0076] 本方案只需要在编译出class文件之后,生成dex文件之前,对所有class文件进行遍历,比如需要上报点击事件,只需要查找到实现了View.OnClickListener接口的class文件,在其中的onClick(View view)方法中注入埋点上报数据的代码。

[0077] 当点击前述步骤中提到的同一页面的任意一个Button时候,demo都会显示当前点击Button的点击时间、所在页面、Button的ID、Button的文字描述。例如点击其中一个Button,该Button点击不但会弹出一个对话框(弹出对话框可以当作核心业务),而且会显示当前点击Button的详细时间、页所处页面、Button的ID等(这些是埋点上报数据业务),所以通过修改字节码注入埋点上报数据的代码并不会影响原有的核心业务,二者是相互独立的。

[0078] 同样的,继续点击对话框中的控件也会显示相应的详细时间、所处页面、对话框按键内容等,实现无代码全埋点的效果。对于复杂的页面,页面的任意元素,控件被触发时,也会显示相应的时间、页面、ID等。

[0079] 另外,如果现在要跟踪页面的切换,比如比Activity生命周期还有复杂的Fragment的切换的事件跟踪监听,可以在编译出字节码之后查找所有继承继承自Fragment的类,比在其onResume()、onPause()、setUserVisibleHint(boolean isVisibleToUser)、onHiddenChanged(boolean hidden)方法中注入跟踪页面切换的代码。当切换到BaseFragment页面,能成功追踪到BaseFragment变为可视化的时刻。当退出当前BaseFragment页面,能成功追踪到BaseFragment退出的时刻,以及用户在BaseFragment的驻留时长。

[0080] 2)、iOS实现

[0081] 同理,iOS可以使用Cydia Substrate等进行插桩编程插入埋点上报数据的代码。

[0082] 四、统一化、自动化埋点上报数据,实现解耦

[0083] 在前述步骤完成后,当检测到页面的界面元素被点击、选择、切换等操作时,运行其对应的事件响应程序,同时执行插入在事件响应程序中的埋点上报数据的代码,实现埋点数据上报。。

[0084] 因此,采用本方法后,各种事件的数据上报的代码逻辑,可以单独编写,实现了埋点数据上报业务与核心业务的相互独立,降低了耦合度,便于后期的维护与扩展。

[0085] 系统实施例

[0086] 参照图4,本实施例提供了一种应用程序埋点系统,包括:

[0087] 至少一个处理器100;

[0088] 至少一个存储器200,用于存储至少一个程序;

[0089] 当所述至少一个程序被所述至少一个处理器100执行,使得所述至少一个处理器100实现所述的应用程序埋点方法。

[0090] 本实施例的应用程序埋点系统,可执行本发明方法实施例所提供的应用程序埋点方法,可执行方法实施例的任意组合实施步骤,具备该方法相应的功能和有益效果。

[0091] 以上是对本发明的较佳实施进行了具体说明,但本发明创造并不限于所述实施例,熟悉本领域的技术人员在不违背本发明精神的前提下还可做出种种的等同变形或替换,这些等同的变型或替换均包含在本申请权利要求所限定的范围内。



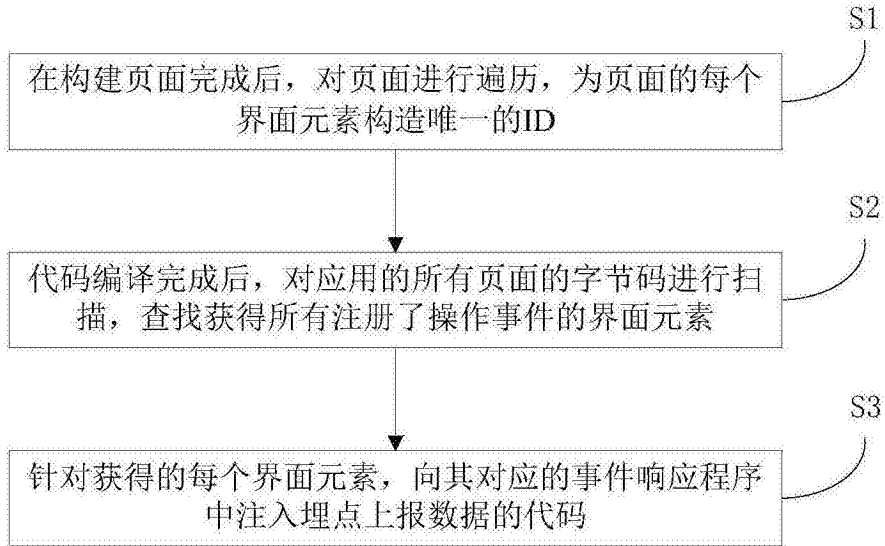


图1

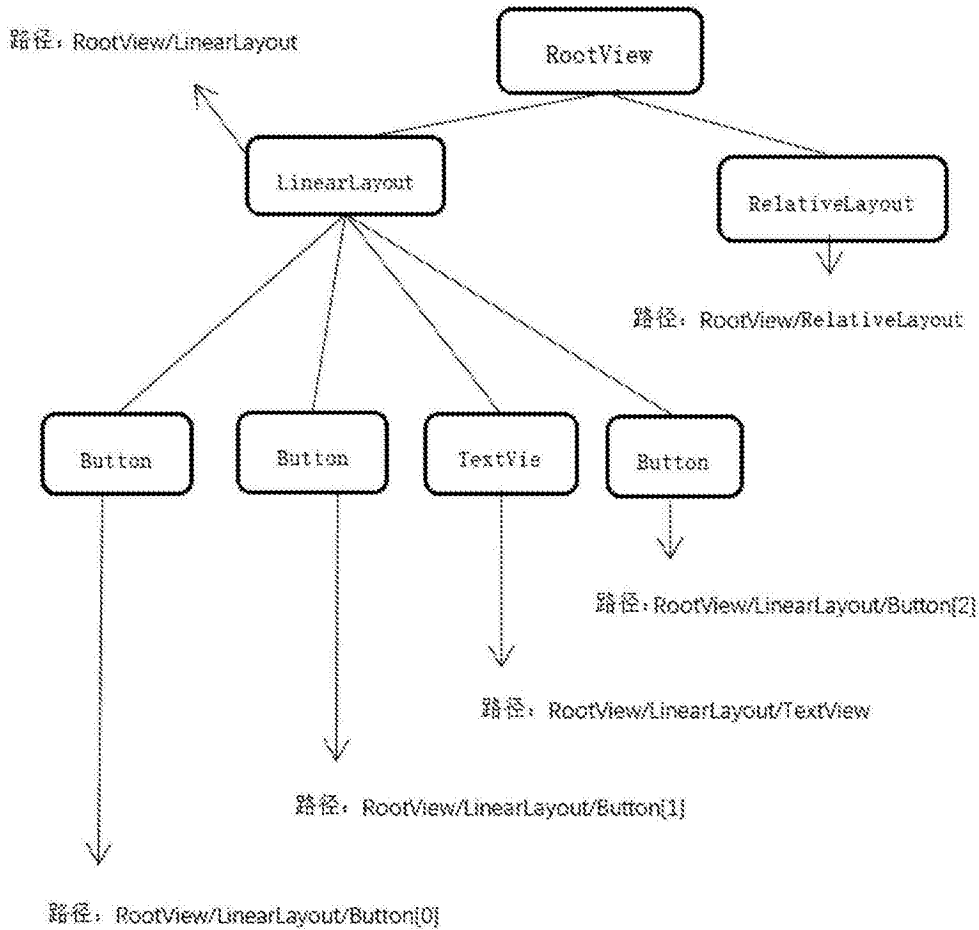


图2

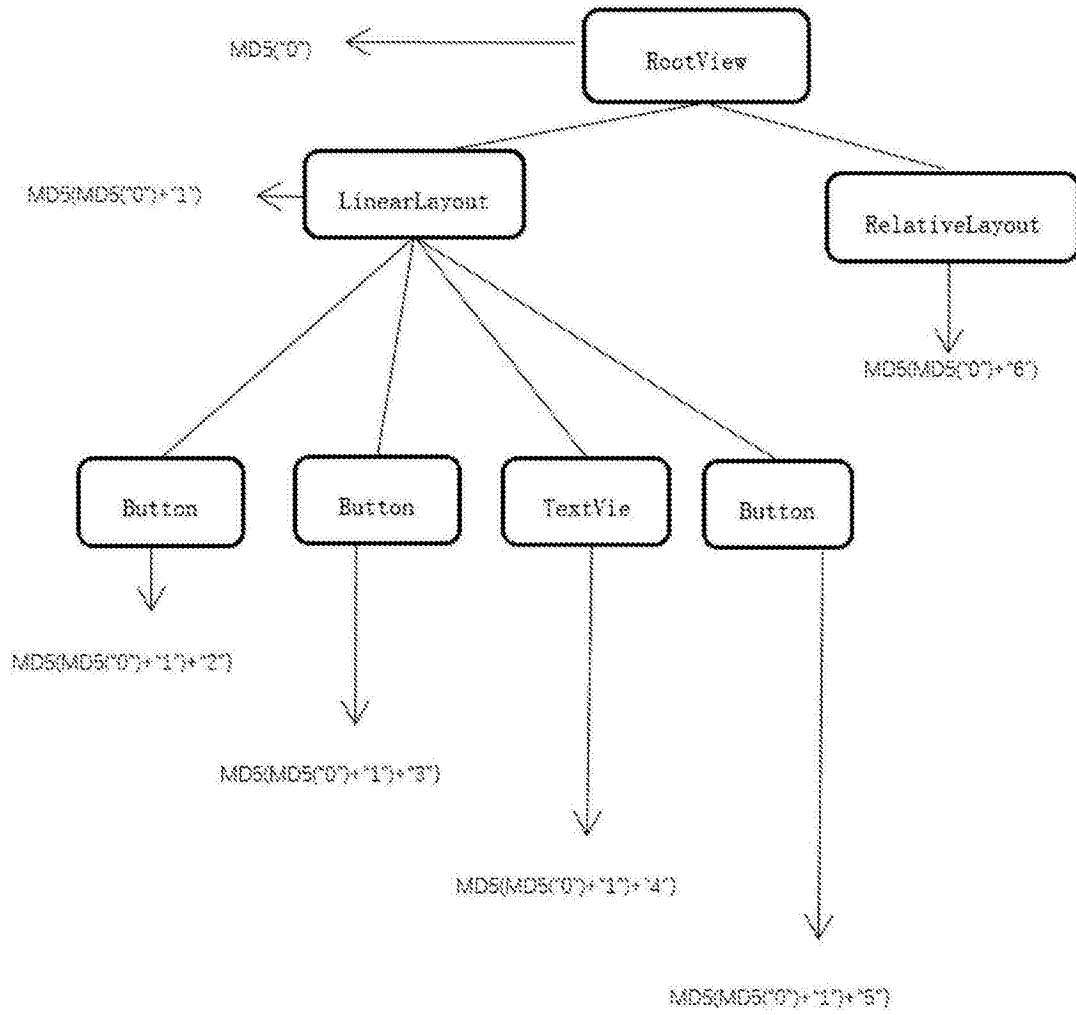


图3

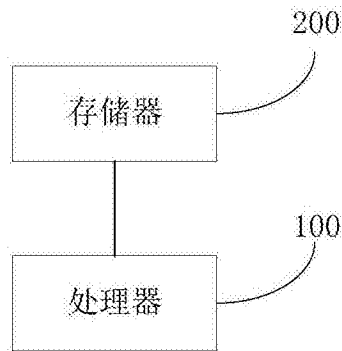


图4