



(12) 发明专利申请

(10) 申请公布号 CN 112230917 A

(43) 申请公布日 2021. 01. 15

(21) 申请号 202011084491.4

(22) 申请日 2020.10.12

(71) 申请人 上海赛可出行科技服务有限公司
地址 200131 上海市浦东新区自由贸易试
验区杨高北路2001号1幢4部位三层
333室

(72) 发明人 金小俊 赵化 王兴明

(51) Int.Cl.

G06F 8/38 (2018.01)

G06F 11/36 (2006.01)

G06F 3/0488 (2013.01)

G06F 9/451 (2018.01)

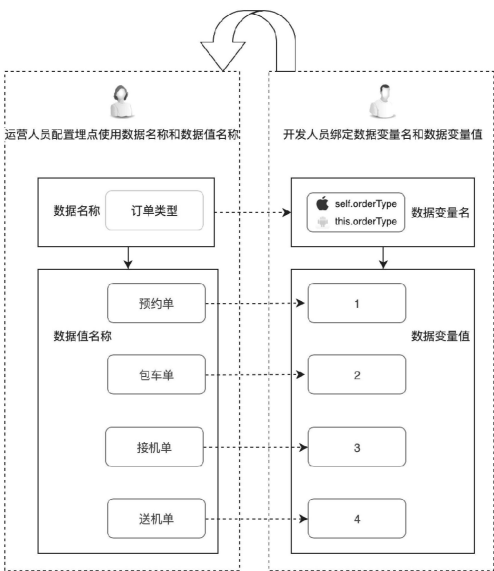
权利要求书2页 说明书8页 附图3页

(54) 发明名称

一种基于数据和状态的移动应用埋点方法

(57) 摘要

本发明公开了一种基于数据和状态的移动应用埋点方法,涉及数据采集分析技术领域,方法的步骤包括:一、开发人员以页面为单位将对应类中的变量映射成数据,将控件交互、页面交互、视图交互和方法执行映射成状态;二、运营人员通过对这些数据和状态进行组合,配置出埋点信息;三、将埋点信息配置表下发到移动客户端;四、客户端在运行时对配置表中的数据和状态进行匹配,实现埋点的记录。本发明实现了埋点数据的精准采集,无需大数据的参与,更为重要的是,本发明实现了Android和iOS平台埋点采集过程的统一,无需对两个平台分别进行埋点配置和埋点下发。



1. 一种基于数据和状态的移动应用埋点方法,其特征在于,所述移动应用埋点方法包含埋点配置MIS系统和客户端埋点SDK,埋点配置和实施流程如下:

一、MIS系统埋点配置

以页面为单位进行埋点配置,页面以类名作为唯一标识;若存在该页面的设计图,则直接将设计图导入MIS系统,若没有相应的设计图,则在APP端进行截屏并将该页面截图上传到MIS系统,若是导入的设计图则需要开发人员在MIS端输入页面标识符;

1. 数据配置

页面导入成功后,由开发人员将数据名称,数据变量名和数据值名称进行绑定后,绑定后生成数据列表,供运营人员配置埋点时使用,比如开发人员在数据/状态配置系统配置数据订单类型,数据名称为订单类型,数据变量名安卓为`this.orderType`,iOS为`self.orderType`,数据值名称为预约单,往返单,包车单,对应的变量值为0,1,2,配置完成后,运营人员在埋点配置系统看到的数据为:可配置数据订单类型,对应的取值为预约单,往返单,包车单;

2. 状态配置

状态配置为指定在何种情况下进行埋点值的匹配和记录,MIS系统默认提供两种类型的可选状态,包括通用状态和方法执行状态,通用状态包括控件交互,页面交互和视图交互状态,方法执行状态为指定的方法或函数的执行状态;

2.1 控件交互

控件交互包括点击事件,长按手势,拖动手势,滑动手势,缩放手势,表示用户对APP进行交互的状态,比如下单页面,有三个tab,对应订单类型中的预约单,往返单,包车单,点击相应的tab后展现不同的下单界面,则运营人员在配置状态时为,设置控件交互下的点击事件状态,当订单类型为预约单记录埋点值`click_reserve_order`,即当有点击事件发生时,检测订单类型是否发生变化,若发生变化且新值为预约单,则记录埋点值`click_reserve_order`;

2.2 页面交互

页面交互状态主要用来记录PV埋点,包括页面进入,页面退出,在MIS系统可直接配置页面进入和页面退出分别对应的埋点值,比如在订单详情页,配置页面进入状态下的埋点值为`order_detail_enter`,页面退出状态下的埋点值为`order_detail_leave`,埋点SDK在监控导航栈变化的时候,当导航栈新增内容时判断当前页面是否是订单详情页,若是,则记录埋点`order_detail_enter`,页面退出的埋点同理;

2.3 视图交互

视图交互状态包括普通视图,弹窗两种,普通视图为当前页面上添加或展现的视图,弹窗则为Window上添加或展现的视图,在设置视图交互状态时还需指定视图的名称,视图的名称以视图的类名来标识,如果是非自定义的视图,不同的弹窗其类名是相同的,这种情况下视图标识符取类名加弹窗上的标题文本,比如点击查看计费详情按钮,会弹出计费详情页面,其状态配置为:在普通视图状态时,当视图名称为计费详情页时,记录埋点值`click_price_detail_view`,需要注意的是,此处的计费详情页也属于数据配置,需要由开发人员事先绑定数据值后再提供给运营人员,计费详情页会绑定标识名称,即类名`PriceDetailView`;

2.4方法执行

方法执行状态用来表示某种操作的进行,以页面中的指定方法开始执行和结束执行为依据,比如,当用户请求行程预估价格时,记录埋点值request_estimate_price,请求预估价格失败时记录埋点值request_estimate_price_failed.在这个情况下,首先由开发人员将请求行程预估价格与类中的方法requestEstimatePrice()进行绑定,之后,运营人员在MIS端配置埋点:在方法执行-请求行程预估价格-开始状态时,记录埋点值request_estimate_price,在方法执行-请求预估价格-结束状态时,当返回值为false时,记录埋点值request_estimate_price_failed,方法执行默认提供开始和结束两个子状态,对应到埋点SDK端即hook的时候,在原方法执行前注入埋点代码和原方法执行后注入埋点代码;

二、客户端埋点SDK

客户端埋点SDK包含的功能和流程如下:

1. 与MIS系统的连接和信息传输

埋点SDK实现在DEBUG模式下摇一摇上传当前页面截图到MIS端,并在上传之前判断是否与MIS已建立连接,若未建立连接则调用接口上传连接信息至MIS系统;

2. 数据变量值的读取和匹配

对MIS端下发的埋点配置表中的变量在状态条件满足时进行匹配,运行时读取值的方法:iOS端采用KVC的方式,Android采用反射的方式;

3. 状态的监控

3.1控件交互状态的监控

通过hook控件的响应事件来进行监控,对于点击事件,hook系统的点击事件方法,当该方法执行时表示点击事件发生,此时读取埋点配置表中该状态下的数据列表,对数据值进行匹配,根据匹配结果记录埋点值,手势事件的监控同理;

3.2页面交互状态的监控

对系统的导航栈进行监控,若导航栈内的页面发生变化,即代表页面发生了跳转,根据导航栈内页面是增加还是减少来判断是页面进入或是页面退出;

3.3视图交互状态的监控

对当前视图或Window视图的所有子视图列表进行监控,若列表发生增减,即代表视图发生了变化,根据发生变化的视图标识符来进行埋点匹配;

3.4方法执行状态的监控

读取MIS系统下发的埋点配置表中指定的状态方法名,对所有指定的方法进行hook操作,hook时根据子状态是开始还是结束来判断hook的操作是在原方法执行前生效还是执行后生效,另外,hook方法时同时获取方法的入参和出参,供埋点的子状态下数据匹配时使用。

一种基于数据和状态的移动应用埋点方法

技术领域

[0001] 本发明涉及数据采集分析技术领域,特别涉及一种基于数据和状态的移动应用埋点方法。

背景技术

[0002] 目前业内的埋点方法主要有代码埋点,可视化埋点和全埋点。

[0003] 代码埋点相对较灵活,支持包含逻辑判断的多场景设置,但缺点是埋点代码和业务代码高耦合,且埋点更新依赖发版,不支持动态更新。除此之外,安卓和iOS平台需要分别编码实现埋点。

[0004] 可视化埋点通过识别页面内的可埋点控件(添加了各类事件的控件,比如点击事件、手势事件等等)后将页面上传MIS端,运营人员在MIS端对各控件添加埋点配置。优点是配置过程可视化,且支持动态更新。其缺点主要有两点:

[0005] 由于页面实现方式的复杂性和不确定性导致可埋点控件的识别时有遗漏或误判;

[0006] iOS和Android的控件标识无法统一,造成两端需要分别进行埋点配置。

[0007] 全埋点的方式是捕捉和监测用户在应用里的所有行为,并全部上报。优点是由于是全量数据采集,所以产品迭代过程中不需要关注埋点逻辑。缺点是,采集数据量大,且无法实现埋点数据的精准采集,需要配合大数据分析,成本较高。

发明内容

[0008] 本发明要解决的技术问题是克服现有技术的缺陷,提供一种基于数据和状态的移动应用埋点方法。

[0009] 为了解决上述技术问题,本发明提供了如下的技术方案:

[0010] 本发明一种基于数据和状态的移动应用埋点方法,所述移动应用埋点方法包含埋点配置MIS系统和客户端埋点SDK,埋点配置和实施流程如下:

[0011] 一、MIS系统埋点配置

[0012] 以页面为单位进行埋点配置,页面以类名作为唯一标识;若存在该页面的设计图,则直接将设计图导入MIS系统,若没有相应的设计图,则在APP端进行截屏并将该页面截图上传到MIS系统,若是导入的设计图则需要开发人员在MIS端输入页面标识符;

[0013] 1.数据配置

[0014] 页面导入成功后,由开发人员将数据名称,数据变量名和数据值名称进行绑定后,绑定后生成数据列表,供运营人员配置埋点时使用,比如开发人员在数据/状态配置系统配置数据订单类型,数据名称为订单类型,数据变量名安卓为this.orderType,iOS为self.orderType,数据值名称为预约单,往返单,包车单,对应的变量值为0,1,2,配置完成后,运营人员在埋点配置系统看到的数据为:可配置数据订单类型,对应的取值为预约单,往返单,包车单;

[0015] 2.状态配置

[0016] 状态配置为指定在何种情况下进行埋点值的匹配和记录,MIS系统默认提供两种类型的可选状态,包括通用状态和方法执行状态,通用状态包括控件交互,页面交互和视图交互状态,方法执行状态为指定的方法或函数的执行状态;

[0017] 2.1控件交互

[0018] 控件交互包括点击事件,长按手势,拖动手势,滑动手势,缩放手势,表示用户对APP进行交互的状态,比如下单页面,有三个tab,对应订单类型中的预约单,往返单,包车单,点击相应的tab后展现不同的下单界面,则运营人员在配置状态时为,设置控件交互下的点击事件状态,当订单类型为预约单记录埋点值click_reserve_order,即当有点击事件发生时,检测订单类型是否发生变化,若发生变化且新值为预约单,则记录埋点值click_reserve_order;

[0019] 2.2页面交互

[0020] 页面交互状态主要用来记录PV埋点,包括页面进入,页面退出,在MIS系统可直接配置页面进入和页面退出分别对应的埋点值,比如在订单详情页,配置页面进入状态下的埋点值为order_detail_enter,页面退出状态下的埋点值为order_detail_leave,埋点SDK在监控导航栈变化的时候,当导航栈新增内容时判断当前页面是否是订单详情页,若是,则记录埋点order_detail_enter,页面退出的埋点同理;

[0021] 2.3视图交互

[0022] 视图交互状态包括普通视图,弹窗两种,普通视图为当前页面上添加或展现的视图,弹窗则为Window上添加或展现的视图,在设置视图交互状态时还需指定视图的名称,视图的名称以视图的类名来标识,如果是非自定义的视图,不同的弹窗其类名是相同的,这种情况下视图标识符取类名加弹窗上的标题文本,比如点击查看计费详情按钮,会弹出计费详情页面,其状态配置为:在普通视图状态时,当视图名称为计费详情页时,记录埋点值click_price_detail_view,需要注意的是,此处的计费详情页也属于数据配置,需要由开发人员事先绑定数据值后再提供给运营人员,计费详情页会绑定标识名称,即类名PriceDetailView;

[0023] 2.4方法执行

[0024] 方法执行状态用来表示某种操作的进行,以页面中的指定方法开始执行和结束执行为依据,比如,当用户请求行程预估价格时,记录埋点值request_estimate_price,请求预估价格失败时记录埋点值request_estimate_price_failed.在这个情况下,首先由开发人员将请求行程预估价格与类中的方法requestEstimatePrice()进行绑定,之后,运营人员在MIS端配置埋点:在方法执行-请求行程预估价格-开始状态时,记录埋点值request_estimate_price,在方法执行-请求预估价格-结束状态时,当返回值为false时,记录埋点值request_estimate_price_failed,方法执行默认提供开始和结束两个子状态,对应到埋点SDK端即hook的时候,在原方法执行前注入埋点代码和原方法执行后注入埋点代码;

[0025] 二、客户端埋点SDK

[0026] 客户端埋点SDK包含的功能和流程如下:

[0027] 1. 与MIS系统的连接和信息传输

[0028] 埋点SDK实现在DEBUG模式下摇一摇上传当前页面截图到MIS端,并在上传之前判断是否与MIS已建立连接,若未建立连接则调用接口上传连接信息至MIS系统;

[0029] 2.数据变量值的读取和匹配

[0030] 对MIS端下发的埋点配置表中的变量在状态条件满足时进行匹配,运行时读取值的方法:iOS端采用KVC的方式,Android采用反射的方式;

[0031] 3.状态的监控

[0032] 3.1控件交互状态的监控

[0033] 通过hook控件的响应事件来进行监控,对于点击事件,hook系统的点击事件方法,当该方法执行时表示点击事件发生,此时读取埋点配置表中该状态下的数据列表,对数据值进行匹配,根据匹配结果记录埋点值,手势事件的监控同理;

[0034] 3.2页面交互状态的监控

[0035] 对系统的导航栈进行监控,若导航栈内的页面发生变化,即代表页面发生了跳转,根据导航栈内页面是增加还是减少来判断是页面进入或是页面退出;

[0036] 3.3视图交互状态的监控

[0037] 对当前视图或Window视图的所有子视图列表进行监控,若列表发生增减,即代表视图发生了变化,根据发生变化的视图标识符来进行埋点匹配;

[0038] 3.4方法执行状态的监控

[0039] 读取MIS系统下发的埋点配置表中指定的状态方法名,对所有指定的方法进行hook操作,hook时根据子状态是开始还是结束来判断hook的操作是在原方法执行前生效还是执行后生效,另外,hook方法时同时获取方法的入参和出参,供埋点的子状态下数据匹配时使用。

[0040] 与现有技术相比,本发明的有益效果如下:

[0041] 1.通过数据和状态的组合可以描述任意埋点场景,故本发明对于场景化埋点高度支持;

[0042] 2.和可视化埋点类似,本发明亦提供了一个MIS系统实现数据和状态的可视化配置,但无需识别页面内的控件,也无需对控件做唯一标识;

[0043] 3.多平台自然统一(埋点对应的数据名称是相同的,只是在各平台代码中绑定的变量名不同);

[0044] 4.埋点实现逻辑简单,可行性高,埋点设置为对数据和状态进行配置和管理,埋点记录则为指定状态下数据值的匹配过程;

[0045] 5.支持埋点的动态修改和下发;

[0046] 6.埋点逻辑与业务代码逻辑无侵入无耦合。

附图说明

[0047] 附图用来提供对本发明的进一步理解,并且构成说明书的一部分,与本发明的实施例一起用于解释本发明,并不构成对本发明的限制。在附图中:

[0048] 图1为实施例数据绑定和配置示意图;

[0049] 图2为实施例通用状态绑定和配置示意图;

[0050] 图3为实施例方法执行状态绑定和配置示意图;

[0051] 图4为实施例埋点语义化配置示意图;

[0052] 图5为实施例整体流程和步骤示意图。

具体实施方式

[0053] 以下结合附图对本发明的优选实施例进行说明,应当理解,此处所描述的优选实施例仅用于说明和解释本发明,并不用于限定本发明。

[0054] 实施例1

[0055] 如图1-5所示,本发明提供一种基于数据和状态的移动应用埋点方法,移动应用埋点方法包含埋点配置MIS系统和客户端埋点SDK,埋点配置和实施流程如下:

[0056] 一、MIS系统埋点配置

[0057] 以页面(iOS为UIViewController,Android为Activity)为单位进行埋点配置,页面以类名作为唯一标识;若存在该页面的设计图,则直接将设计图导入MIS系统,若没有相应的设计图,则在APP端进行截屏并将该页面截图上传到MIS系统(埋点SDK实现摇一摇上传页面截图的功能,同时附着上传该页面的类名标识符);若是导入的设计图则需要开发人员在MIS端输入页面标识符;

[0058] 1.数据配置

[0059] 页面导入成功后,由开发人员将数据名称,数据变量名和数据值名称进行绑定后,绑定后生成数据列表,供运营人员配置埋点时使用。比如开发人员在数据/状态配置系统配置数据订单类型,数据名称为订单类型,数据变量名安卓为this.orderType,iOS为self.orderType,数据值名称为预约单,往返单,包车单,对应的变量值为0,1,2,配置完成后,运营人员在埋点配置系统(数据/状态配置系统和埋点配置系统为MIS配置系统的不同页面)看到的数据为:可配置数据订单类型,对应的取值为预约单,往返单,包车单;

[0060] 2.状态配置

[0061] 状态配置为指定在何种情况下(触发条件)进行埋点值的匹配和记录。MIS系统默认提供两种类型的可选状态,包括通用状态和方法执行状态。通用状态包括控件交互,页面交互和视图交互状态;方法执行状态为指定的方法或函数的执行状态;

[0062] 2.1控件交互

[0063] 控件交互包括点击事件,长按手势,拖动手势,滑动手势,缩放手势,表示用户对APP进行交互的状态,比如下单页面,有三个tab,对应订单类型中的预约单,往返单,包车单,点击相应的tab后展现不同的下单界面,则运营人员在配置状态时为,设置控件交互下的点击事件状态,当订单类型为预约单记录埋点值click_reserve_order,即当有点击事件发生时,检测订单类型是否发生变化,若发生变化且新值为预约单,则记录埋点值click_reserve_order;

[0064] 2.2页面交互

[0065] 页面交互状态主要用来记录PV埋点,包括页面进入,页面退出,在MIS系统可直接配置页面进入和页面退出分别对应的埋点值,比如在订单详情页,配置页面进入状态下的埋点值为order_detail_enter,页面退出状态下的埋点值为order_detail_leave,埋点SDK在监控导航栈变化的时候,当导航栈新增内容时判断当前页面是否是订单详情页,若是,则记录埋点order_detail_enter,页面退出的埋点同理;

[0066] 2.3视图交互

[0067] 视图交互状态包括普通视图,弹窗两种,普通视图为当前页面上添加或展现的视图,弹窗则为Window上添加或展现的视图,在设置视图交互状态时还需指定视图的名称,视

图 的名称以视图的类名来标识,如果是非自定义的视图,不同的弹窗其类名是相同的,这种情况下视图标识符取类名加弹窗上的标题文本(若类名相同标题亦相同则认为是同一个视图),比如点击查看计费详情按钮,会弹出计费详情页面,其状态配置为:在普通视图状态时,当视图名称为计费详情页时,记录埋点值click_price_detail_view,需要注意的是,此处的计费详情页也属于数据配置,需要由开发人员事先绑定数据值后再提供给运营人员,计费详情页会绑定标识名称,即类名PriceDetailView;

[0068] 2.4方法执行

[0069] 方法执行状态用来表示某种操作的进行,以页面中的指定方法开始执行和结束执行为依据,比如,当用户请求行程预估价格时,记录埋点值request_estimate_price,请求预估价格失败时记录埋点值request_estimate_price_failed.在这个情况下,首先由开发人员将请求行程预估价格与类中的方法requestEstimatePrice()进行绑定,之后,运营人员在MIS端配置埋点:在方法执行-请求行程预估价格-开始状态时,记录埋点值request_estimate_price,在方法执行-请求预估价格-结束状态时,当返回值为false时,记录埋点值request_estimate_price_failed,方法执行默认提供开始和结束两个子状态,对应到埋点SDK端即hook的时候,在原方法执行前注入埋点代码和原方法执行后注入埋点代码;

[0070] 二、客户端埋点SDK

[0071] 客户端埋点SDK包含的功能和流程如下:

[0072] 1.与MIS系统的连接和信息传输

[0073] 埋点SDK实现在DEBUG模式下摇一摇上传当前页面截图到MIS端,并在上传之前判断是否与MIS已建立连接,若未建立连接则调用接口上传连接信息(包括bundleId,平台,当前登录APP账号,设备名称)至MIS系统

[0074] 2.数据变量值的读取和匹配

[0075] 对MIS端下发的埋点配置表中的变量在状态条件满足时进行匹配(即读取运行时的数据值和表中的值进行匹配),运行时读取值的方法:iOS端采用KVC的方式,Android采用反射的方式;

[0076] 3.状态的监控

[0077] 3.1控件交互状态的监控

[0078] 通过hook控件的响应事件来进行监控,对于点击事件,hook系统的点击事件方法(iOS hook UIControl的sendAction:to:forEvent:方法,Android hook onClick方法),当该方法执行时表示点击事件发生,此时读取埋点配置表中该状态下的数据列表,对数据值进行匹配(或数据值组合),根据匹配结果记录埋点值,手势事件的监控同理;

[0079] 3.2页面交互状态的监控

[0080] 对系统的导航栈进行监控,若导航栈内的页面发生变化,即代表页面发生了跳转(页面交互),根据导航栈内页面是增加还是减少来判断是页面进入或是页面退出;

[0081] 3.3视图交互状态的监控

[0082] 对当前视图或Window视图的所有子视图列表进行监控,若列表发生增减,即代表视图发生了变化,根据发生变化的视图标识符来进行埋点匹配;

[0083] 3.4方法执行状态的监控

[0084] 读取MIS系统下发的埋点配置表中指定的状态方法名,对所有指定的方法进行

hook操作, hook时根据子状态是开始还是结束来判断hook的操作是在原方法执行前生效还是执行后生效, 另外, hook方法时同时获取方法的入参和出参(返回值), 供埋点的子状态下数据匹配时使用。

[0085] 具体的, 本发明中的名词解释:

[0086] 一、数据

[0087] 数据包括数据名称, 数据变量名, 数据值名称和数据变量值。

[0088] 数据名称和数据值名称是指在MIS中提供的供运营人员配置埋点的数据描述, 比如数据名称订单状态, 数据值名称行程中。

[0089] 数据变量名则为数据名称对应的代码中与之绑定的代码变量名, 比如iOS中订单状态对应的变量名为self.orderType, Android中订单状态对应的变量名为this.orderType

[0090] 数据变量值指程序实际运行中该变量对应的值, 比如订单状态为已预订的时候订单状态的数据值为1, 订单状态为行程中的时候订单状态对应的数据值为2。

[0091] 数据示例如下表所示:

[0092]	编号	数据名称	数据变量名	数据值名称	数据变量值
	1	订单类型	orderType	预约单	1
				包车单	2
	2	订单状态	orderStatus	待上车	1
				行程中	2
				已完成	3

[0093] 数据名称, 数据变量名和数据值名称在MIS端进行绑定。

[0094] 如此, 运营人员的配置流程即为, 选择订单状态, 当订单状态为已预订时记录埋点值order_reserved, 当订单状态为行程中时记录埋点值order_ongoing。

[0095] APP端接收到埋点配置表后, 在运行时对相应的数据变量进行匹配, 当变量值为1时记录埋点值order_reserved, 当变量值为2时记录埋点值order_ongoing;

[0096] 二、状态

[0097] APP端在对数据变量值进行匹配时需要一个触发时机, 即什么时候去检查数据的变化。传统的可视化埋点是在用户点击了埋点控件后, 在事件响应方法中记录埋点值, 也就是触发的时机是控件事件发生时。由此, 本方案引入状态的概念, 在状态发生变化的时候检

查数据的变化,状态包含以下几类:

[0098] 1.发生用户操作事件时(hook点击事件,手势事件等);

[0099] 2.UI发生变化时,比如导航栈中的控制器发生了增减;

[0100] 3.当前页面或Window上的视图层发生了变化;

[0101] 4.页面内指定的某个函数(方法)执行前或执行后;

[0102] 这四类状态的详细解释和示例应用场景如下:

[0103] 1.发生用户操作事件时(hook点击事件,手势事件等)

[0104] hook各类控件的各类事件响应方法,比如按钮的点击事件或者视图的手势事件等。这些事件的响应方法被触发即代表用户进行了某种类型的操作,当该状态发生时对变量值进行检测和埋点处理。具体的示例,假设APP中有4个tab,用户选中不同的tab需要记录不同的埋点值,当点击事件发生时,即去检测tab对应的变量值是否发生了变化(需要保存一份初始值列表入栈用来做对比),若发生了变化则记录对应的埋点。在这个情况下,运营人员在MIS端的埋点配置流程为:在控件交互-点击事件状态下,当TAB标识为设置页时,记录埋点值click_setting_tab;

[0105] 2.页面发生变化时(导航栈中的控制器内容发生了增减)

[0106] 监控系统导航栈,当导航栈的内容发生变化时,对其内的数据进行匹配检查。具体的示例,假设需要在用户进入订单详情页时记录埋点值,则APP端在导航栈发生变化即页面发生跳转的时候检测当前页面是否是订单详情页,若是,则记录埋点值。在这个情况下,运营人员在MIS端的埋点配置流程为:在订单详情页面进行配置,当页面交互-页面进入状态时,记录埋点值view_order_detail_page。

[0107] 3.当前页面或Window上的视图层发生了变化

[0108] 监控当前页面或Window图层子视图的变化,比如用户删除订单时会弹出弹窗,让用户确认是否删除,在这个情况下记录埋点值user_delete_order,同时需要判断用户是个人用户还是企业用户。在这个情况下,运营人员在MIS端的埋点配置流程为:在视图交互-弹窗状态时,当弹窗名称为删除订单时,且用户类型为个人用户时,记录埋点值individual_user_delete_order;

[0109] 4.页面内指定的某个函数(方法)执行前或执行后

[0110] 在这个情况下,状态为某个函数的执行。hook指定的函数,当函数执行时即为状态发生,此时检查数据的变化,进行匹配并记录埋点。比如,当用户请求行程预估价格时,记录埋点值request_estimate_price,请求预估价格失败时记录埋点值request_estimate_price_failed在这个情况下,运营人员在MIS端的埋点配置流程为:在方法执行-请求行程预估价格-开始状态时,记录埋点值request_estimate_price,在方法执行-请求行程预估价格-结束状态时,当返回值为false时,记录埋点值request_estimate_price_failed。

[0111] 如此,埋点的配置过程也是一个语义化的描述过程。

[0112] 总结而言,本发明提出的全新的创新式埋点思想,集合了代码埋点的场景化埋点的灵活度支持,又避免了代码埋点耦合;集合了可视化埋点的可视化配置功能,又避免了可视化埋点繁琐的控件标识以及控件事件的单一埋点场景的局限。而相对于全埋点来说,本发明实现了埋点数据的精准采集,无需大数据的参与。更为重要的是,本发明实现了Android和iOS平台埋点采集过程的统一,无需对两个平台分别进行埋点配置和埋点下发。

[0113] 最后应说明的是:以上所述仅为本发明的优选实施例而已,并不用于限制本发明,尽管参照前述实施例对本发明进行了详细的说明,对于本领域的技术人员来说,其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

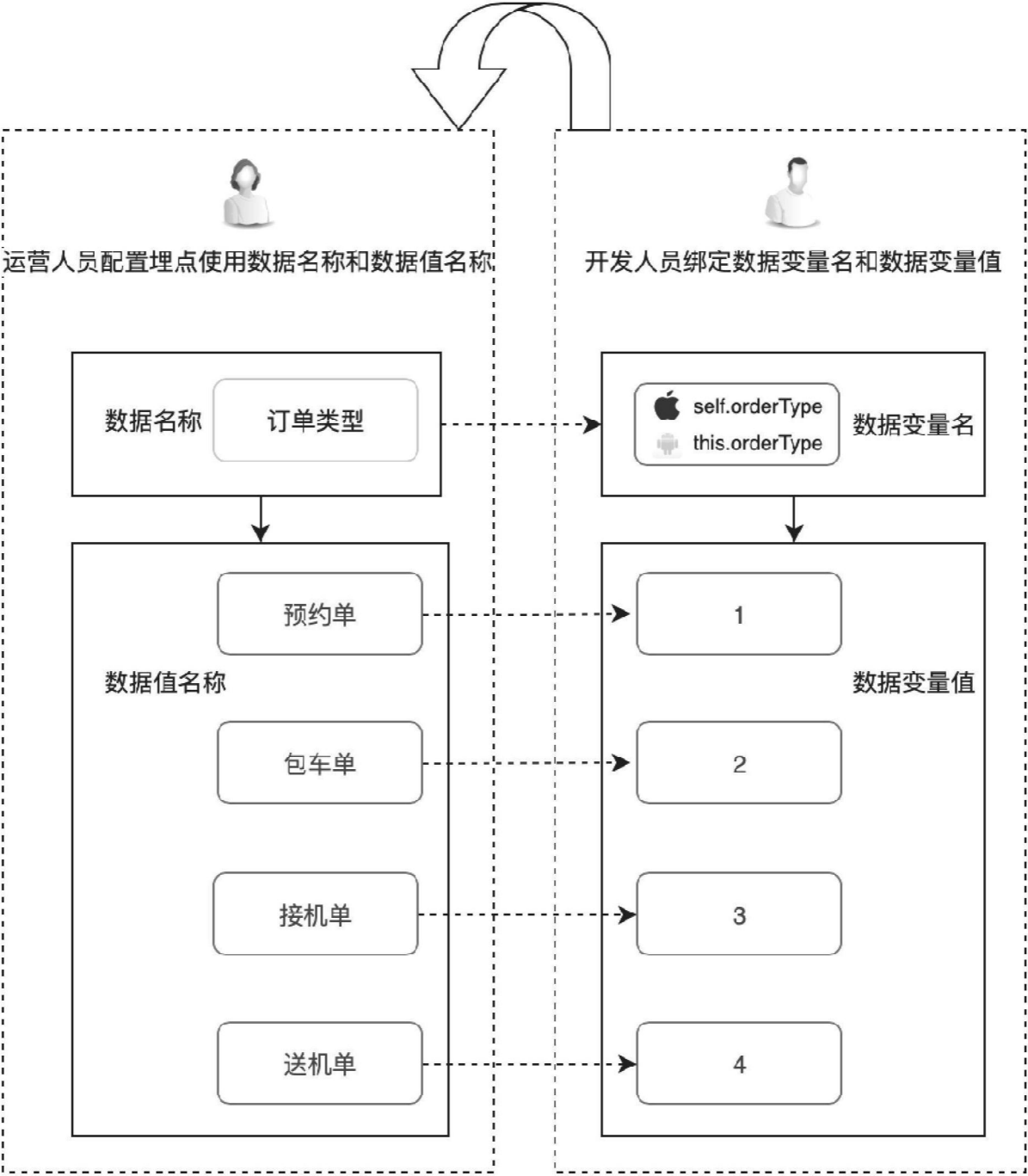


图1

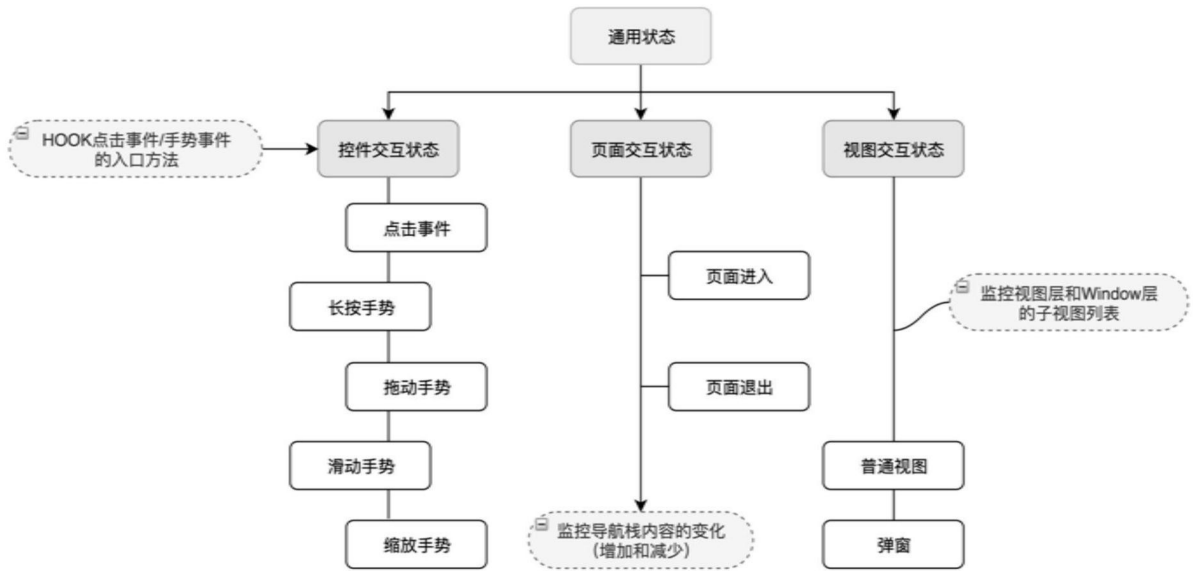


图2

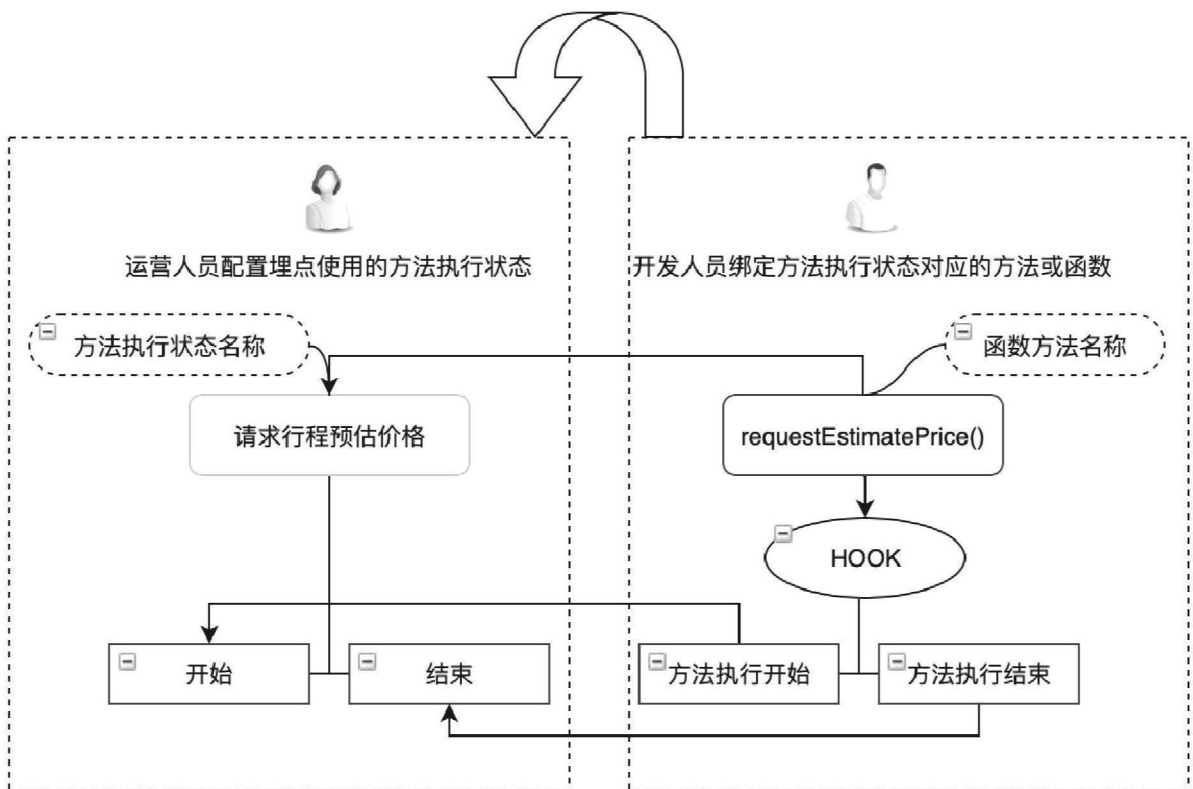


图3

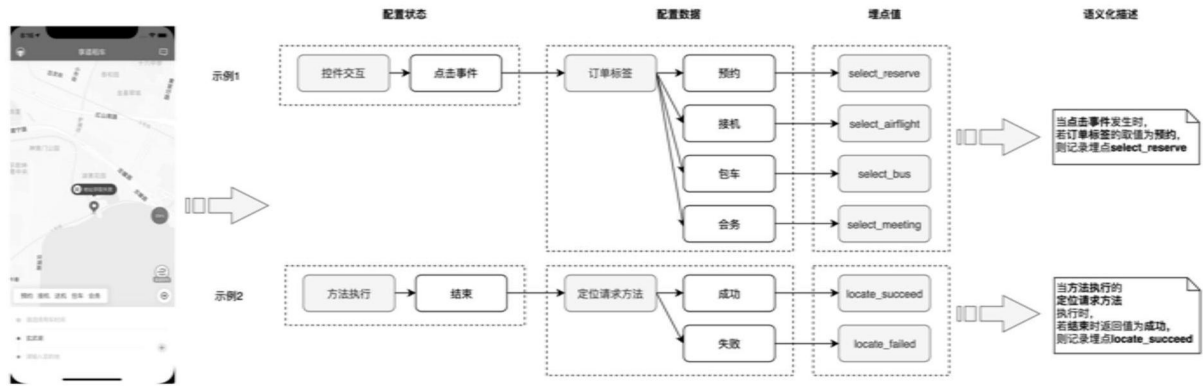


图4

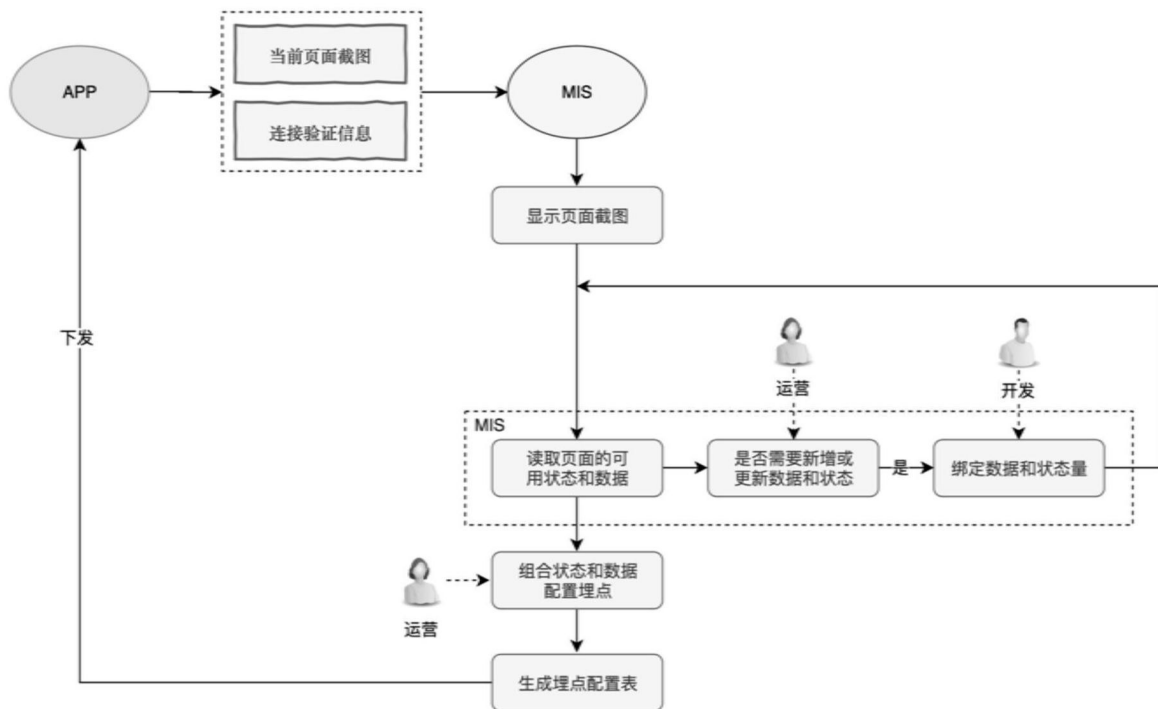


图5