



(12)发明专利申请

(10)申请公布号 CN 108023960 A

(43)申请公布日 2018.05.11

(21)申请号 201711299059.5

(22)申请日 2017.12.08

(71)申请人 百度在线网络技术(北京)有限公司
地址 100085 北京市海淀区上地十街10号
百度大厦三层

(72)发明人 殷勇

(74)专利代理机构 北京市铸成律师事务所

11313

代理人 张臻贤 武晨燕

(51)Int.Cl.

H04L 29/08(2006.01)

G06F 17/30(2006.01)

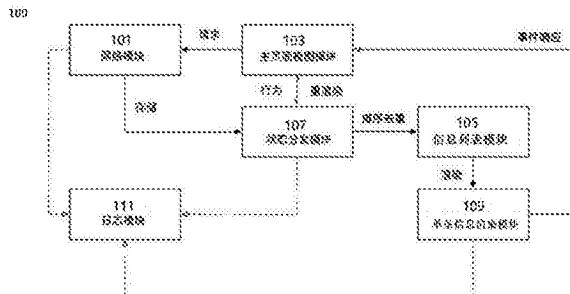
权利要求书2页 说明书7页 附图3页

(54)发明名称

构建Feed流应用的装置和Feed流信息推送的方法

(57)摘要

本发明涉及一种构建Feed流应用的装置，所述Feed流应用包括多个视图组件，所述装置包括：信息列表模块，所述信息列表模块配置成可存储多条信息数据；网络模块，所述网络模块配置成可拉取信息数据并存储到所述信息列表模块中；主页面视图模块，所述主页面视图模块配置成从所述信息列表模块中查询当前可用的信息数据并显示，并监听所述Feed流应用的用户事件；状态分发模块，所述状态分发模块配置成监听所述多个视图组件的状态变量，并且当所述状态变量发生变动时，触发视图的更新。



1. 一种构建Feed流应用的装置,所述Feed流应用包括多个视图组件,所述装置包括:
信息列表模块,所述信息列表模块配置成可存储多条信息数据;
网络模块,所述网络模块配置成可拉取信息数据并存储到所述信息列表模块中;
主页面视图模块,所述主页面视图模块配置成从所述信息列表模块中查询当前可用的信息数据并显示,并监听所述Feed流应用的用户事件;
状态分发模块,所述状态分发模块配置成监听所述多个视图组件的状态变量,并且当所述状态变量发生变动时,触发所述Feed流应用的视图的更新。
2. 根据权利要求1所述的装置,其特征在于,还包括:信息渲染模块,所述信息渲染模块配置成将单条信息数据转换为相应的视图,从而由所述主页面视图模块显示。
3. 根据权利要求1或2所述的装置,其特征在于,还包括:日志统计模块,所述日志统计模块配置成记录所述Feed流应用运行中所涉及的业务指标。
4. 根据权利要求1-3中任一项所述的装置,其特征在于,所述信息列表模块配置成将所述信息数据分为置顶信息列表和正常排序信息列表,所述信息数据优选地为新闻数据。
5. 根据权利要求1-4中任一项所述的装置,其特征在于,所述信息列表模块配置成可对其中存储的信息数据进行去重和/或排序操作。
6. 根据权利要求1-5中任一项所述的装置,其特征在于,所述主页面视图模块采用React-Native的ListView组件渲染被显示的信息数据。
7. 根据权利要求1-6中任一项所述的装置,其特征在于,当监听所述Feed流应用的用户事件后,所述主页面视图模块执行与所述用户事件相对应的业务逻辑。
8. 根据权利要求1-7中任一项所述的装置,其特征在于,所述信息列表模块配置成:当其中存储的信息数据的任意属性字段变化之后,触发持久化操作,将一定数量的信息数据存储在本地设备上。
9. 根据权利要求1-7中任一项所述的装置,其特征在于,所述信息列表模块、网络模块、主页面视图模块和状态分发模块均依托React-Native构建。
10. 根据权利要求1-9中任一项所述的装置,其特征在于,还包括React-Native模块,所述React-Native模块提供应用程序接口API。
11. 根据权利要求1-10中任一项所述的装置,其特征在于,所述状态分发模块配置成当所述状态变量发生变动时,触发与所述发生变动的状态变量相关联的视图组件的更新。
12. 一种Feed流信息推送的方法,包括:
拉取信息数据;
在页面上显示所述信息数据,所述页面包括多个视图组件;
监听用户事件;当监听到用户事件后,执行与所述用户事件相对应的业务逻辑;
监听所述多个视图组件的状态变量,当所述状态变量发生变动时,触发所述页面视图的更新。
13. 根据权利要求12所述的方法,其特征在于,所述在页面上显示所述信息数据的步骤包括:采用React-Native的ListView组件渲染所述信息数据。
14. 一种计算机可读存储介质,包括存储于其上的计算机可执行指令,所述可执行指令在被处理器执行时实施如权利要求12-13中任一项所述的Feed流信息推送的方法。
15. 一种信息推送系统,包括:

信息服务器,配置成提供信息数据;
如权利要求1-11所述的构建Feed流应用的装置,所述装置的网络模块可与所述信息服务器通讯并从所述信息服务器拉取信息数据。

构建Feed流应用的装置和Feed流信息推送的方法

技术领域

[0001] 本发明大致涉及移动应用领域,尤其涉及一种构建Feed流应用的装置和Feed流信息推送的方法。

背景技术

[0002] Feed流由于其表现新闻类信息的方式丰富而又不失灵活,因此受到互联网内容提供商的青睐,大量提供新闻资讯、垂类聚合的移动客户端都设计了提供海量无限信息的Feed流应用。鉴于互联网产品形态对于迭代速度和用户体验的强烈需求,Feed流应用必须从技术上解决在传统的客户端开发模式下的效率低下、版本碎片等体验问题。

[0003] 一个典型的案例是在手机百度客户端中,除了百度搜索外的最重要产品功能即是Feed流垂类聚合新闻。在不断的产品迭代需求背景下,客户端固有的发版节奏已然过于迟缓,且无法对已发布的旧版本客户端中的Feed流进行同步升级。

[0004] 目前业界实现Feed信息流的现有技术方案主要是使用对应移动操作系统上的原生API进行硬编码到客户端安装包中。或者使用内嵌浏览器内核,并搭载传统HTML页面的方式实现。

[0005] 现有技术的主要问题在于:(1)工作量冗余,由于不同操作系统的API不同,必须同时投入双倍的开发和维护成本,包括人力配置;((2)开发成本高,使用操作系统原生API需要更为专业的编码技能;(3)灵活性差,对于放出去的客户端,没有让用户立刻升级的有效办法,很难对软件故障加以补救;(4)HTML页面的实现,用户体验稍有欠缺。

[0006] 以上内容仅是发明人所知晓的技术情况,并不当然代表构成本发明的现有技术。

发明内容

[0007] 为解决现有技术的问题中的一个或多个,本发明提供一种构建Feed流应用的装置,所述Feed流应用包括多个视图组件,所述装置包括:信息列表模块,所述信息列表模块配置成可存储多条信息数据;网络模块,所述网络模块配置成可拉取信息数据并存储到所述信息列表模块中;主页面视图模块,所述主页面视图模块配置成从所述信息列表模块中查询当前可用的信息数据并显示,并监听所述Feed流应用的用户事件;状态分发模块,所述状态分发模块配置成监听所述多个视图组件的状态变量,并且当所述状态变量发生变动时,触发视图的更新。

[0008] 根据本发明的一个方面,所述构建Feed流应用的装置还包括:信息渲染模块,所述信息渲染模块配置成将单条信息数据转换为相应的视图,从而由所述主页面视图模块显示。

[0009] 根据本发明的一个方面,所述构建Feed流应用的装置还包括:日志统计模块,所述日志统计模块配置成记录所述Feed流应用运行中所涉及的业务指标。

[0010] 根据本发明的一个方面,所述信息列表模块配置成将所述信息数据分为置顶信息列表和正常排序信息列表;

[0011] 根据本发明的一个方面,所述信息列表模块配置成可对其中存储的信息数据进行去重和/或排序操作。

[0012] 根据本发明的一个方面,所述主页面视图模块采用React-Native的ListView组件渲染被显示的信息数据。

[0013] 根据本发明的一个方面,当监听所述Feed流应用的用户事件后,所述主页面视图模块执行与所述用户时间相对应的业务逻辑。

[0014] 根据本发明的一个方面,所述信息列表模块配置成:当其中存储的信息数据的任意属性字段变化之后,触发持久化操作,将一定数量的信息数据存储在本地设备上。

[0015] 本发明还涉及一种Feed流信息推送的方法,包括:拉取信息数据;在页面上显示所述信息数据,所述页面包括多个视图组件;

[0016] 监听用户事件;监听所述多个视图组件的状态变量,当所述状态变量发生变动时,触发所述页面视图的更新。

[0017] 根据本发明的一个方面,所述在页面上显示所述信息数据的步骤包括:采用React-Native的ListView组件渲染所述信息数据

[0018] 本发明还涉及一种计算机可读存储介质,包括存储于其上的计算机可执行指令,所述可执行指令在被处理器执行时实施如上所述的智能取色方法。

[0019] 本发明还涉及一种信息推送系统,包括:信息服务器,配置成提供信息数据;如上所述的构建Feed流应用的装置,所述装置的网络模块可与所述信息服务器通讯并从所述信息服务器拉取信息数据。

附图说明

[0020] 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:

[0021] 图1示出了根据本发明第一实施例的一种构建Feed流应用的装置;

[0022] 图2示出了根据本发明第二实施例的一种Feed流信息推送的方法;

[0023] 图3示出了根据本发明第三实施例的一种信息推送系统;

[0024] 图4示出了根据本发明第四实施例的计算机程序产品;和

[0025] 图5示出了一个视图组件的实例。

具体实施方式

[0026] 在下文中,仅简单地描述了某些示例性实施例。正如本领域技术人员可认识到的那样,在不脱离本发明的精神或范围的情况下,可通过各种不同方式修改所描述的实施例。因此,附图和描述被认为本质上是示例性的而非限制性的。

[0027] 在本发明的描述中,需要理解的是,术语“中心”、“纵向”、“横向”、“长度”、“宽度”、“厚度”、“上”、“下”、“前”、“后”、“左”、“右”、“竖直”、“水平”、“顶”、“底”、“内”、“外”、“顺时针”、“逆时针”等指示的方位或位置关系为基于附图所示的方位或位置关系,仅是为了便于描述本发明和简化描述,而不是指示或暗示所指的装置或元件必须具有特定的方位、以特定的方位构造和操作,因此不能理解为对本发明的限制。此外,术语“第一”、“第二”仅用于描述目的,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数

量。由此,限定有“第一”、“第二”的特征可以明示或者隐含地包括一个或者更多个所述特征。在本发明的描述中,“多个”的含义是两个或两个以上,除非另有明确具体的限定。

[0028] 在本发明的描述中,需要说明的是,除非另有明确的规定和限定,术语“安装”、“相连”、“连接”应做广义理解,例如,可以是固定连接,也可以是可拆卸连接,或一体地连接;可以是机械连接,也可以是电连接或可以相互通讯;可以是直接相连,也可以通过中间媒介间接相连,可以是两个元件内部的连通或两个元件的相互作用关系。对于本领域的普通技术人员而言,可以根据具体情况理解上述术语在本发明中的具体含义。

[0029] 在本发明中,除非另有明确的规定和限定,第一特征在第二特征之“上”或之“下”可以包括第一和第二特征直接接触,也可以包括第一和第二特征不是直接接触而是通过它们之间的另外的特征接触。而且,第一特征在第二特征“之上”、“上方”和“上面”包括第一特征在第二特征正上方和斜上方,或仅仅表示第一特征水平高度高于第二特征。第一特征在第二特征“之下”、“下方”和“下面”包括第一特征在第二特征正上方和斜上方,或仅仅表示第一特征水平高度小于第二特征。

[0030] 下文的公开提供了许多不同的实施方式或例子用来实现本发明的不同结构。为了简化本发明的公开,下文中对特定例子的部件和设置进行描述。当然,它们仅为示例,并且目的不在于限制本发明。此外,本发明可以在不同例子中重复参考数字和/或参考字母,这种重复是为了简化和清楚的目的,其本身不指示所讨论各种实施方式和/或设置之间的关系。此外,本发明提供了的各种特定的工艺和材料的例子,但是本领域普通技术人员可以意识到其他工艺的应用和/或其他材料的使用。

[0031] 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。

[0032] 图1示出了根据本发明第一实施例的一种构建Feed流应用的装置100。Feed流例如是一个可通过上拉或者下拉的无限信息(例如新闻)的排列。下面参考图1详细描述装置100。

[0033] 如图1所示,根据本发明第一实施例的构建Feed流应用的装置100包括信息列表模块105、网络模块101、主页面视图模块103和状态分发模块107。

[0034] 其中,主页面视图模块103是入口模块,是面向用户的模块,负责渲染Feed流应用的整体视图,并响应视图的各种用户操作事件。

[0035] 在Feed流应用启动时,主页面视图模块103从原生客户端的启动参数中获取一些必要的变量,包括但不限于操作系统信息、硬件信息等基础常量,也可能是变量,比如,当前时刻的网络环境。而对于需要异步获取的数据,例如当前网络状态等随时变化的值,此时开始向客户端请求。视图例如由多个并列、嵌套的组件构成。例如如图5所示,其中,中间位置被框出的部分就是一个组件,与之并列的是其它的组件。

[0036] 之后,主页面视图模块103向信息列表模块105查询当前可用的信息数据,无论结果如何,例如使用React-Native的ListView组件渲染Feed列表,并传入各种可以响应用户事件的回调函数,保证业务逻辑都冒泡到该模块中进行统一处理。根据本发明的一个实施例,主页面视图模块103可以直接向信息列表模块105查询调取当前可用的信息数据,并进行渲染和显示。根据本发明的一个变形,主页面视图模块103也可以通过状态分发模块107向信息列表模块105查询调取当前可用的信息数据,并进行相应的处理,例如对信息列表模

块105中的信息数据进行排序和/或去重,这些都在本发明的保护范围内。

[0037] 在渲染显示了Feed列表之后,主页面视图模块103监听Feed的各种点击、滑动等用户事件,并执行对应的业务逻辑。例如,当用户点击某一条信息的标题时,显示该信息具体的内容;当用户意图删除其中一条信息数据时,删除页面上的该条信息。

[0038] 网络模块101配置成负责从远程服务器上拉取下一批信息列表,拉取的信息列表的具体数量由服务器依据请求参数等因素来控制。网络模块101拉取信息列表可以由用户事件触发拉取请求,例如用户在主页面上进行下滑或者上拉操作时,主页面视图模块103可以请求网络模块101拉取下一批次的信息数据列表。网络模块101拉取信息列表也可以由其他事件触发,包括但不限于例如缓存失效后需要自动强制拉取新数据的情形。

[0039] 所述网络模块101例如通过使用NativeModules这个React-Native提供的API,从原生客户端获取一些必要的实时参数,与业务参数合并;

[0040] 然后,通过fetch API向远程服务器请求新的批次数据,经过解析和过滤,推送给信息列表模块105。

[0041] 所述网络模块101锁拉取的信息数据可以直接存储到所述信息列表模块105,也可以通过状态分发模块107,进行一定的预处理之后,再存储到所述信息列表模块105,这些都在本发明的保护范围内。状态分发模块107将在下面详细描述。

[0042] 状态分发模块107是一个标准的有限状态机,可以有逻辑清晰且运行正确的程序来实现,也可以由包含软件的硬件来实现。状态分发模块107负责维护整个Feed应用的诸多状态,并将状态的变动精准高效地反映到视图上去。所述状态比如是否正在访问网络加载数据、是通过哪种方式触发的拉取新数据,最近一次拉取数据的结果和时间等等。

[0043] 下面描述状态分发模块107的工作方式。

[0044] 首先,在应用的首次运行期间,状态分发模块107会记录每个视图组件所使用到的具体状态变量。所述状态变量例如:已经持有的所有feed数据数组,是否还有更多数据需要拉取,是否已经在拉取数据,上一次是否拉取失败,上一次拉取数据的时间戳。所述视图组件典型的即为模板,从Feed流中可以看到,模板例如是从上到下排列的视觉模块,一般由一条分割线分隔。模板之间可以有复用的子组件,比如底部信息栏,标题等等。组件之间例如可能存在树形关系。

[0045] 其次,状态分发模块107监听每个具体状态变量值的改动,并将新的变量值推送到对应的视图组件中去,完成视图的更新。

[0046] 基于以上原理,主页面视图模块103上的业务逻辑无需主动操作视图,而是通过修改状态变量值,自动触发视图的更新。根据本发明的一个优选实施例,当其中一些组件的状态变量发生变化时,仅触发这些发生变化的组件的视图更新,而非整个视图的更新。另外,这种触发行为具有一定的事务性优化,即对于短时间内状态的多次变动,视图的更新仅限于最多一次,这样可以尽量避免无谓的视图再更新,因此它是相当消耗性能的。对于Feed应用来说,典型的操作是上拉和下拉操作,也包括用户通过叉号删除其中元素。这些操作都是直接去操作数据,并自动反应到视图来实现的。例如当用户点击其中一条信息条目上的叉号以指示删除该信息条目时,动画启动,将该信息高度渐变为0,然后从全局feed数组中删除此条目,状态分发模块会自动删除此数据对应的视图模板。

[0047] 上面所述事务性优化,例如可以设置一个计时器,可以设定计时器的计时长度,在

计时长度到达时，重渲染视图。根据本发明的一个优选实施例，只有被分配了CPU时间片才能到达计时长度末尾，因为只要业务逻辑中拥有CPU时间间隙，计时器便会获得机会。

[0048] 状态分发模块107能够捕获到状态变量与视图的对应关系。在状态变化时，寻找依赖此变量的所有视图组件并令其重新渲染，这是效率最高的更新视图方式，避免了高昂的无效渲染。例如状态变量onWIFI表示当前是否处于WIFI环境，在此环境下，所有动图组件(GIF)都将依赖它，在它为真的时候，自动下载和播放。状态分发模块会尝试在程序启动时渲染一次所有视图，在视图的渲染逻辑中可以使用到这些状态变量，而这些状态变量通常都是可以追踪到它被使用的视图组件，因此在这个时机就可以得知依赖改变量的视图组件。从抽象角度来讲，信息数据的首次展现也是一种状态更新的行为。业务逻辑的行为无非是修改各种状态变量，只不过有些变量会触发视图更新，有些不会，但这是状态分发模块需要关心的事情，而不是业务逻辑。换句话说，业务逻辑最终都是操作了状态分发模块，但如何分发的，是由视图的依赖决定的。以上描述的是状态分发模块107更新与发生变化的状态变量相关联的组件的视图。本领域技术人员能够理解，在其中一个或一些状态变量发生变化时，本发明也可以更新全部的视图，虽然这样的处理方式并非是最优的选择。

[0049] 信息列表模块105负责存储所有信息，例如通过类似于数组的数据结构来存储信息数据。所述信息例如是新闻信息，当然也可以是运营商提供的其他信息、广告等。每条信息在本发明中可以成为一条feed。下面以新闻数据为例进行说明。本领域技术人员能够理解，本发明可以应用于其他类型的信息，而不仅限于新闻信息。

[0050] 根据本发明的一个优选实施例，为了满足置顶信息的需求，信息列表模块105将所有信息拆分为置顶信息列表H和正常排序信息列表N。

[0051] 在网络模块101拉取到新的信息批次后，所拉取的信息数据，通过状态分发模块107格式转换为统一的格式，便于后续逻辑统一处理。信息列表模块105接收到转换后的信息数据后，依次遍历所有信息条目，从中摘取需要置顶的信息放入H中，如果需要唯一，则还需要进行去重，去除H中的同类信息，然后把余下的信息放入N中，放入的位置取决于具体用户的操作方法。例如，当用户上拉刷新时，将信息放入末尾。当用户下拉刷新时，将信息放到起始位置。

[0052] 信息列表模块105还可配置成监听置顶信息列表H和正常排序信息列表N中的所有信息属性。所述信息属性例如该条信息是否已经被读取过。这些状态往往会影响到视觉展现，并且在运行时是动态改变的。在任意属性字段改变之后，信息列表模块105触发持久化操作，将从列表顶部开始的一定数量的有限信息存储在本地设备上，例如存储在移动客户端的本地化缓存中，该存储不受断网断电的影响，但可能有一定的时效性。

[0053] 当主页面视图模块启动时，将会立即取出并渲染被持久化存储的信息条目，显示在主界面上。

[0054] 根据本发明的一个变形，所述装置100还可以包括单条信息渲染模块109。单条信息渲染模块109负责将单条信息数据转换为对应的视图。为了做到整个Feed流应用对单个信息视图的高度解耦，将每种单条Feed视图所需要的数据抽象并包装为统一的格式。单条信息渲染模块109例如是连接数据与视觉模板的桥梁，它读取单条Feed的数据属性，例如可以是文字、图片、视频等，来选取合适的视觉模板，交给React-Native。

[0055] 此外，单条信息渲染模块109在视图上还负责包裹信息视图，以统一支持一定的动

画效果。

[0056] 根据本发明的一个变形，所述装置100还包括日志统计模块111。日志统计模块111负责记录运行中所涉及的关键业务指标，如页面浏览量PV、浏览用户数UV等。

[0057] 在主页面视图模块启动时，通过传入一些静态参数，日志统计模块111也随之初始化，并记录固有参数到内存中；

[0058] 在指定的业务指标事件发生时，通过事件系统通知该模块，并传递必要参数，该模块将所有参数组合，再通过HTTP请求发回到日志服务器，完成行为监控，服务器通过分析和计算，将指标数据反映出来。

[0059] 根据本发明的一个优选实施例，所述构建Feed流应用的装置100还可包括React-Native模块。所述React-Native模块例如作为基础库存在，提供构建视图、网络等模块的应用软件接口（API），同时也提供了一些基础视图组件可供直接使用。

[0060] 普通技术人员能够理解，根据本发明第一实施例的构建Feed流应用的装置100可以通过软件的方式实现，也可以通过硬件的方式实现，也可以通过软硬件相结合的方式来实现，这些都在本发明的保护范围内。另外，这里描述的装置100的各个模块，既可以集中部署在移动客户端上，也可以分布式的部署在移动客户端和服务器端上。

[0061] 本发明还涉及一种Feed流信息推送的方法200。下面参考图2描述该方法200。

[0062] 在步骤S201，拉取信息数据。例如从网络上的信息数据服务器，通过HTTP请求，单条地或者成批次地拉取信息数据。

[0063] 在步骤S202，在页面上显示信息数据，所述页面包括多个视图组件。

[0064] 在步骤S203，监听用户事件。当监听所述Feed流应用的用户事件后，执行与所述用户事件相对应的业务逻辑。

[0065] 在步骤S204，监听所述多个视图组件的状态变量，当所述状态变量发生变动时，触发所述页面视图的更新。

[0066] 根据本发明的一个优选实施例，所述在页面上显示所述信息数据的步骤包括：采用React-Native的ListView组件渲染所述信息数据。

[0067] 根据本发明的一个优选实施例，在页面上显示信息数据的步骤包括：从本地存储器中查询当前可用的信息数据并显示。

[0068] 根据本发明的一个优选实施例，所述Feed流信息推送的方法200还包括：记录Feed流应用运行中所涉及的业务指标。

[0069] 根据本发明的一个优选实施例，所述Feed流信息推送的方法200还包括：对所述拉取的信息数据进行去重和/或排序操作。

[0070] 根据本发明的一个优选实施例，所述Feed流信息推送的方法200还包括：当信息数据的任意属性字段变化之后，触发持久化操作，将一定数量的信息数据存储在本地设备上。

[0071] 本发明还涉及一种信息推送系统300。下面参考图3描述该信息推送系统300。如图3所示，所述信息推送系统300包括信息服务器301和构建Feed流应用的装置302。信息服务器301配置成提供信息数据。信息服务器301例如可以根据用户的兴趣、偏好、历史搜索记录等信息，提供相应的信息。所述装置302例如通过有线、无线网络与所述信息服务器301通讯，例如以HTTP的方式向所述信息服务器301发送请求，请求信息服务器301提供单条的或者成批次的信息数据。信息服务器301在收到来自装置302的请求之后，提供相应的信息数

据。

[0072] 图4是依照本发明的至少一些实施例布置的计算机程序产品400的框图。信号承载介质402可以被实现为或者包括计算机可读介质406、计算机可记录介质408、计算机通信介质410或者它们的组合，其存储可配置处理单元以执行先前描述的过程中的全部或一些的编程指令404。这些指令可以包括例如用于使一个或多个处理器执行如下处理的一个或多个可执行指令：拉取信息数据；

[0073] 在页面上显示所述信息数据，所述页面包括多个视图组件；监听用户事件；当监听到用户事件后，执行与所述用户事件相对应的业务逻辑；监听所述多个视图组件的状态变量，当所述状态变量发生变动时，触发所述页面视图的更新。

[0074] 本发明的其中一些实施例提出了一种基于React-Native技术建设Feed流应用的方法。在付出一定的基础设施建设成本后，应用的维护成本、效率、用户体验都有了一定的改善。

[0075] 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。

100

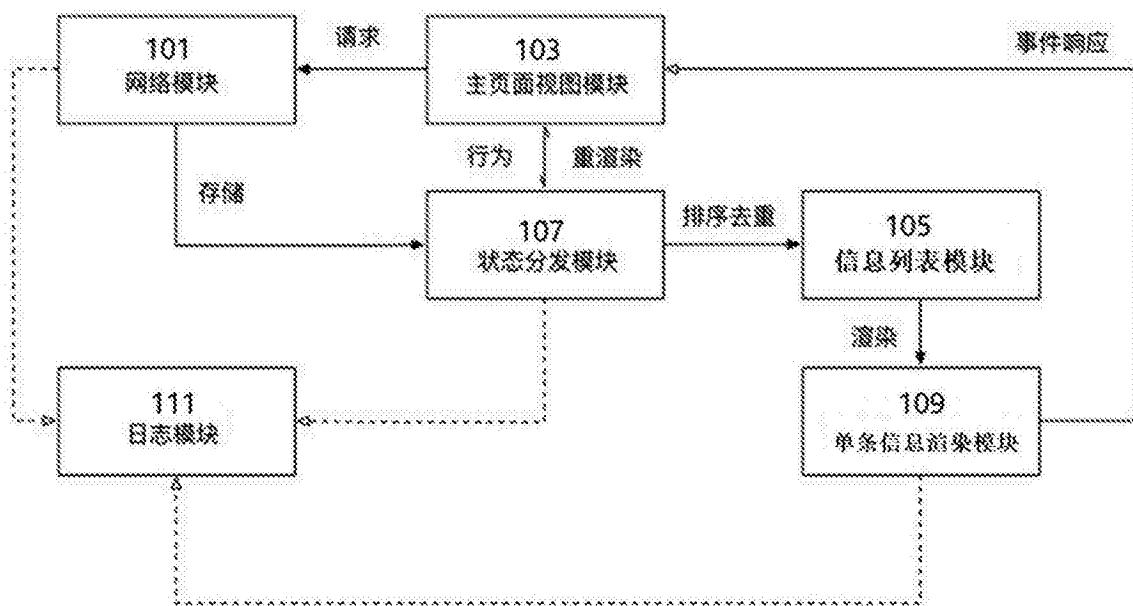


图1

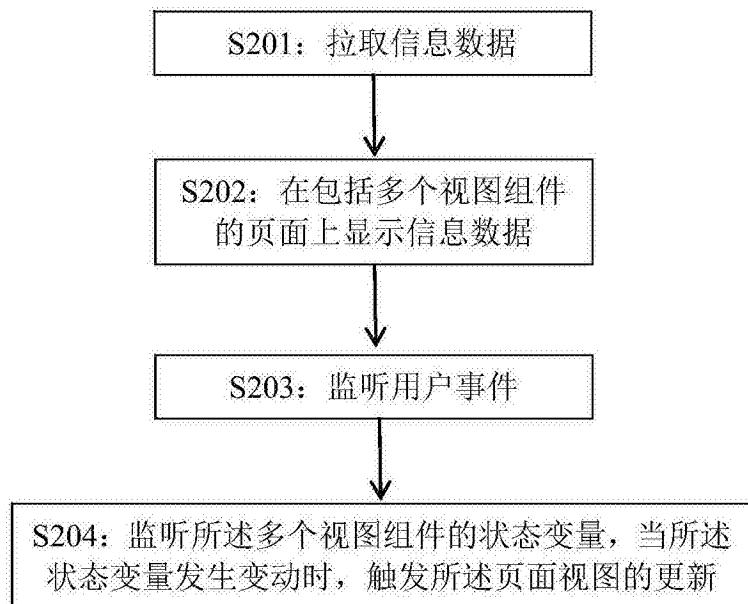


图2

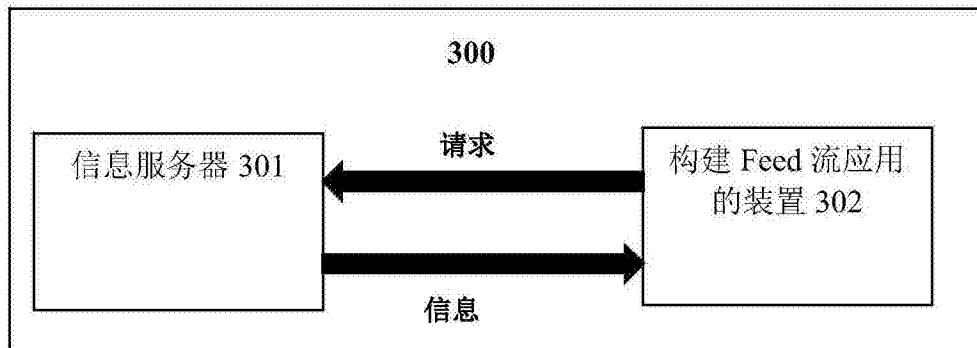


图3

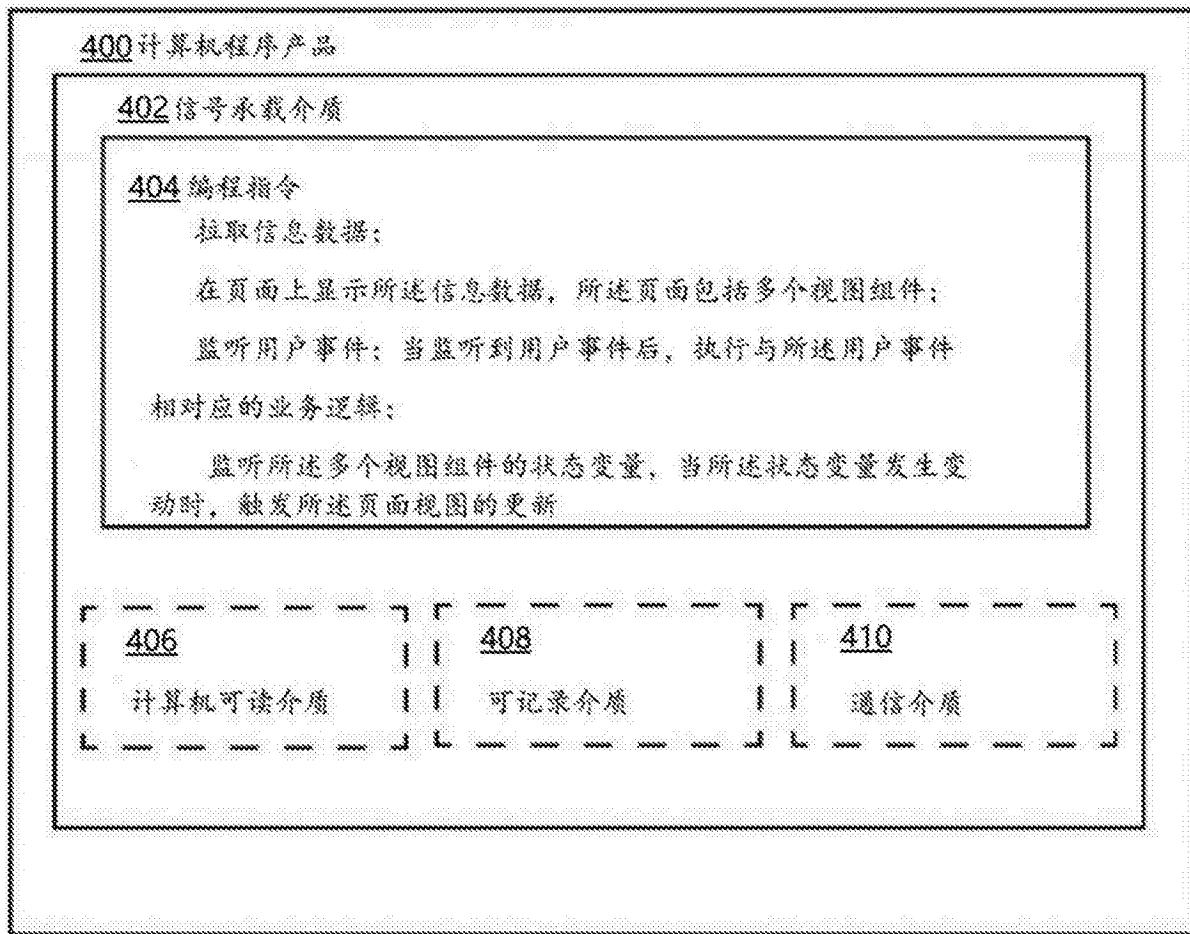


图4



图5