

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2017 年 3 月 30 日 (30.03.2017)



(10) 国际公布号
WO 2017/050145 A1

- (51) 国际专利分类号:
G06F 9/44 (2006.01)
- (21) 国际申请号: PCT/CN2016/098813
- (22) 国际申请日: 2016 年 9 月 13 日 (13.09.2016)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201510604273.1 2015 年 9 月 21 日 (21.09.2015) CN
- (71) 申请人: 阿里巴巴集团控股有限公司 (ALIBABA GROUP HOLDING LIMITED) [—/CN]; 英属开曼群岛大开曼资本大厦一座四层 847 号邮箱, Grand Cayman (KY)。
- (72) 发明人: 及
- (71) 申请人 (仅对美国): 熊亮 (XIONG, Liang) [CN/CN]; 中国浙江省杭州市余杭区文一西路 969 号 3 号楼 5 楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。

- (74) 代理人: 北京国昊天诚知识产权代理有限公司 (CO-HORIZON INTELLECTUAL PROPERTY INC.); 中国北京市朝阳区小关北里甲 2 号渔阳置业大厦 B 座 605, Beijing 100029 (CN)。
- (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。
- (84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO,

[见续页]

(54) Title: METHOD AND DEVICE FOR EXECUTING WEB APP UNDER COMPLETE STATE MODE

(54) 发明名称: 一种执行完全状态模式的 web app 的方法和装置

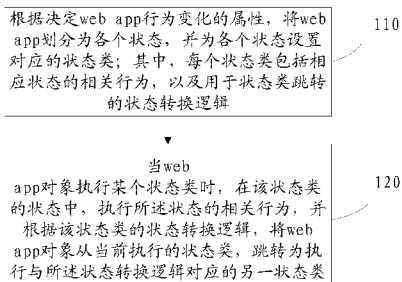


图 1

- 110 DIVIDING A WEB APP INTO VARIOUS STATES ACCORDING TO ATTRIBUTES WHICH DETERMINE THE BEHAVIOUR CHANGE OF THE WEB APP, AND SETTING CORRESPONDING STATE CLASSES FOR THE VARIOUS STATES, WHEREIN EACH OF THE STATE CLASSES COMPRISES A RELATED BEHAVIOUR OF A CORRESPONDING STATE AND A STATE TRANSITION LOGIC FOR STATE CLASS JUMPING
- 120 WHEN A WEB APP OBJECT EXECUTES A CERTAIN STATE CLASS, IN THE STATE OF THE STATE CLASS, EXECUTING THE RELATED BEHAVIOUR OF THE STATE, AND ACCORDING TO THE STATE TRANSITION LOGIC OF THE STATE CLASS, THE WEB APP OBJECT JUMPING FROM THE CURRENT EXECUTED STATE CLASS TO ANOTHER STATE CLASS CORRESPONDING TO THE STATE TRANSITION LOGIC FOR EXECUTION

(57) Abstract: Provided are a method for executing a web app under a complete state mode, a device for executing a web app under a complete state mode and an APP under a complete state mode, which relate to the technical field of computers. The method comprises: pre-dividing a web app into various states according to attributes which determine the behaviour change of the web app, and setting corresponding state classes for the various states, wherein each of the state classes comprises a related behaviour of a corresponding state and a state transition logic for state class jumping; and when a web app object executes a certain state class, in the state of the state class, executing the related behaviour of the state, and according to the state transition logic of the state class, the web app object jumping from the current executed state class to another state class corresponding to the state transition logic for execution. In the present application, a web app can run under a complete state mode, conditional judgement blocks in codes of the web app are small, and are dispersed into specific state classes, and therefore the coupling is low.

(57) 摘要: 本申请提供了一种执行完全状态模式的 web app 的方法、执行完全状态模式的 web app 的装置、一种完全状态模式 APP, 涉及计算机技术领域。所述方法包括: 预先根据决定 web app 行为变化的属性, 将 web app 划分为各个状态, 并为各个状态设置对应的状态类; 其中, 每个状态类包括相应状态的相关行为, 以及用于状态类跳转的状态转换逻辑; 当 web app 对象执行某个状态类时, 在该状态类的状态中, 执行所述状态的相关行为, 并根据该状态类的状态转换逻辑, 将 web app 对象从当前执行的状态类, 跳转为执行与所述状态转换逻辑对应的另一状态类。本申请对于 web app, 能以完全状态模式进行运行, 其代码中条件判断块小, 分散到具体的状态类中, 耦合性低。



RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD,
TG)。

本国际公布:

— 包括国际检索报告(条约第 21 条(3))。

一种执行完全状态模式的web app的方法和装置

技术领域

本申请涉及计算机技术领域，特别是涉及一种执行完全状态模式的 web app 的方法和一种执行完全状态模式的 web app 的装置。

5 背景技术

web app 是一种通过网络（如互联网或内联网）访问的应用程序；也可以指计算机软件承载在浏览器支持环境下或使用浏览器支持语言（如 JavaScript，脚本）并依赖于 web 浏览器来渲染的应用程序。web app 的流行归功于网页浏览器的普及，以及使用这一轻薄客户端方便的用户体验。不必
10 下载安装就可以实现更新和维护，具有支持跨平台的内在属性，是 web app 开始流行的关键原因。典型的 web app 产品包括 web 邮箱、web 商店、wikis 等等。要实现 web app 必须满足以下几点关键条件：

（1）用户交互。web app 必须可以提供界面给用户进行数据展示和用户操作。

15 （2）数据交互。web app 必须可以使用 http 协议通过互联网与 web 服务进行交互，如数据下载和上传

（3）安全保证。web app 必须有能力识别用户身份和权限。

但是在先技术中，对于 web app，其对各个阶段的业务逻辑都在一起实现，其存在大量的条件语句，而巨大的条件语句是不受欢迎的，它们形成一大整块并且使得代码不够清晰，这又使得它们难以修改和扩展，耦合性高。
20 比如，在先技术中，对 web app 的编辑和保存在一个类中，其中，保存时不需要先判断启用哪部分键盘事件和鼠标事件，并且禁用哪部分键盘事件和鼠标事件，而在进行保存行为时也需要先判断启用哪部分键盘事件和鼠标事件，并且禁用哪部分键盘事件和鼠标事件，编辑行为和保存行为对于键盘和
25 鼠标事件的判断则存在耦合，并且代码量较多。

发明内容

鉴于上述问题，提出了本申请实施例以便提供一种克服上述问题或者至少部分地解决上述问题的一种执行完全状态模式的 web app 的方法和相应的一种执行完全状态模式的 web app 的装置。

5

为了解决上述问题，本申请公开了一种执行完全状态模式的 web app 的方法，其特征在于，包括：

预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类；其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑；

当 web app 对象执行某个状态类时，在该状态类的状态中，执行所述状态的相关行为，并根据该状态类的状态转换逻辑，将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的另一状态类。

为了解决上述问题，本申请还公开了一种执行完全状态模式的 web app 的装置，包括：

预置模块，用于预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类；其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑；

状态执行模块，适于当 web app 对象执行某个状态类时，在该状态类的状态中，执行所述状态的相关行为，并根据该状态类的状态转换逻辑，将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的另一状态类。

为了解决上述问题，本申请还公开了一种完全状态模式 web app，包括：

状态执行模块，适于当 web app 对象执行某个状态类时，在该状态类的状态中，执行所述状态的相关行为，并根据该状态类的状态转换逻辑，将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的另一状态类；其中所述各个状态类根据 web app 的状态设置，所述 web app 的状态根据决定 web app 行为变化的属性进行划分；其中，每个状态类包括

相应状态的相关行为，以及用于状态类跳转的状态转换逻辑。

本申请实施例包括以下优点：

本申请实施例预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑，各个状态类之间是平行的，相互独立的，在 web app 执行时，web app 对象的每个状态类独立执行，不和其他的状态存在交叉。因此，本申请实施例对于 web app，能以完全状态模式进行运行，其代码中条件判断块小，即条件判断语句的数量少，分散到具体的状态类中，耦合性低。

比如，对 web app 的编辑和保存，分别设置编辑状态类和保存状态类，在编辑状态类中，直接启用其需要的键盘事件和鼠标事件、禁用其不需要的键盘事件和鼠标事件，然后进入保存状态类后，在保存状态类中直接启用其需要的键盘事件和鼠标事件、禁用其不需要的键盘事件和鼠标事件。由于两个状态是独立的，如此 web app 的保存状态和编辑状态之间对于键盘事件和鼠标事件的逻辑的就解耦了，也不存在大块化的条件判断语句，将大块化的条件判断语句分支化到各个状态类中。

附图说明

图 1 是本申请的一种执行完全状态模式的 web app 的方法实施例的步骤流程图；

图 2 是本申请的另一种执行完全状态模式的 web app 的方法实施例的步骤流程图；

图 2A 是本申请实施例的状态类以及实现状态类相关的基类、环境类之间的关系图；

图 2B 是本申请实施例图 2 实施例的各状态类的转换关系图；

图 3 是本申请的另一种执行完全状态模式的 web app 的方法实施例的步骤流程图；

图 3A 是本申请实施例图 3 实施例的各状态类的转换关系图；

图 3B 是本申请实施例图 3 实施例的弹出窗口示例；

图 4 是本申请的一种执行完全状态模式的 web app 的装置实施例的结构框图；

图 5 是本申请的另一种执行完全状态模式的 web app 的装置实施例的结构框图；

图 6 是本申请的另一种执行完全状态模式的 web app 的装置实施例的结构框图；

图 7 是本申请的一种完全状态模式 web app 的实施例的结构框图。

具体实施方式

10 为使本申请的上述目的、特征和优点能够更加明显易懂，下面结合附图和具体实施方式对本申请作进一步详细的说明。

本申请实施例的核心构思之一在于，基于状态模式，预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑。在 web app 执行时，web app 对象的每个状态类独立执行，不和其他的状态存在交叉。因此，本申请实施例对于 web app，能以完全状态模式进行运行，其代码中条件判断块小，分散到具体的状态类中，耦合性低，并且，可以不用大量的修改原代码，即可很容易的增加新的状态，可扩展性高。

20 其中，所述状态模式的介绍如下：在很多情况下，一个对象的行为取决于一个或多个动态变化的属性，该属性叫做状态，该对象叫做有状态的 (stateful) 对象，该对象状态是从事先定义好的一系列值中取出的。当一个有状态对象，执行该状态类中的行为时，其与外部事件产生互动，继而根据该状态下的状态转换逻辑，该对象的内部状态会改变为其他状态，该对象会执行相应的其他状态类，从而使得对象的行为也随之发生变化。

25 参照图 1，其示出了本申请的一种执行完全状态模式的 web app 的方法实施例的步骤流程图，具体可以包括如下步骤：

步骤 110，预先根据决定 web app 行为变化的属性，将 web app 划分为

各个状态，并为各个状态设置对应的状态类；其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑；

所述状态，指 web app 对象实例的属性的属性值，所述行为，指 web 对象的功能，该行为可以理解为方法。比如方法 `function activateKeyboard ()`，
5 为启用键盘事件的行为。

在本申请实施例中，对于一个 web app，可以根据其逻辑将其分为多个状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑。

需要说明的是，所述状态类比如初始状态类、加载状态类、编辑状态类、
10 保存状态类。当然，实际应用中，状态类可以根据实际需求划分，可以包括任意类，本申请实施例不对其加以限制。

其中，每个状态转换逻辑根据相应状态下的一个具体行为触发状态类的跳转。

步骤 120，当 web app 对象执行某个状态类时，在该状态类的状态中，
15 执行所述状态的相关行为，并根据该状态类的状态转换逻辑，将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的另一状态类。

在实际应用中，web app 启动后，可以创建一个 web app 对象，该 web app 对象可以理解为 web app 的实例。然后为该 web app 对象设置一个最开始的状态类，那么 web app 对象即可首先执行该初始被设置的状态类，然后执行该状态类的相关行为，在执行过程中，可根据该状态类的状态转换逻辑，将 web app 对象跳转为执行另一个状态类。那么跳转后，该 web app 对象执行该另一个状态类的相关行为，在执行过程中根据该状态类的状态转换逻辑再跳转为执行下一个状态类。其他情况以此类推。

25 在实际应用中，步骤 110 可离线进行预置，而步骤 120 在浏览器中执行。

本申请实施例预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑，各个状态类之

间是平行的，相互独立的，在 web app 执行时，web app 对象的每个状态类独立执行，不和其他的状态存在交叉。因此，本申请实施例对于 web app，能以完全状态模式进行运行，其代码中条件判断块小，即条件判断语句的数量少，分散到具体的状态类中，耦合性低。

5 比如对 web app 的编辑和保存，分别设置编辑状态类和保存状态类，在编辑状态类中，直接启用其需要的键盘事件和鼠标事件、禁用其不需要的键盘事件和鼠标事件，然后进入保存状态类后，在保存状态类中直接启用其需要的键盘事件和鼠标事件、禁用其不需要的键盘事件和鼠标事件。由于两个状态是独立的，如此 web app 的保存状态和编辑状态之间
10 对于键盘事件和鼠标事件的逻辑的就解耦了，也不存在大块化的条件判断语句，将大块化的条件判断语句分支化到各个状态类中。

在图 1 实施例的基础上，参照图 2，本申请实施例将状态类划分为了至少下面几个状态类：初始状态类、加载状态类、编辑状态类、保存状态类，
15 以执行本申请实施例的过程。

参照图 2，示出了本申请的一种执行完全状态模式的 web app 的方法实施例的步骤流程图，具体可以包括如下步骤：

步骤 210，预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类；所述状态类包括：初始状态类、加载状态类、编辑状态类、保存状态类；其中，每个状态类包括相应状态
20 的相关行为，以及用于状态类跳转的状态转换逻辑；

所述状态，指 web app 对象实例的属性的属性值，所述行为，指 web 对象的功能，该行为可以理解为方法。比如方法 `function activateKeyboard()`，为启用键盘事件的行为。

25 在本申请实施例中，对于一个 web app，可以根据其逻辑将其分为多个状态类，如初始状态类、加载状态类、编辑状态类、保存状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑。

其中，每个状态转换逻辑根据相应状态下的一个具体行为触发状态类的

跳转。

比如上述初始化状态类的相关行为可包括：为 web app 的各种变量分配空间、为各种变量赋予初始值、初始化鼠标，键盘，网络，以及其他异步处理事件，初始化界面等，其第一状态转换逻辑可包括，在为所有变量赋予初始值之后，跳转至执行加载状态类。

上述加载状态类的相关行为可包括：载入各种信息，渲染 web app 的主界面、获取从上一状态传来的上下文数据、等，其第二状态转换逻辑可包括：在 web app 的主界面渲染完毕之后跳转至执行编辑状态类。

上述编辑状态类的相关行为可包括：启用键盘事件、启用鼠标事件、接收点击保存按钮的事件、获取从上一状态传来的上下文数据、web app 关闭等各种业务逻辑等，其第三状态转换逻辑可包括：接收到对保存按钮的点击事件后，跳转至执行保存状态类。

上述保存状态类的相关行为可包括：保存当前 web app 中的内容至存储区域，取消保存操作（如，接收到 esc 事件）、获取从上一状态传来的上下文数据等，其第四状态转换逻辑可包括：在保存完毕之后或者接收到取消保存操作后，跳转至执行编辑状态类。

在实际应用中，本申请实施例可以如图 2A 设置：

环境类（Context）：定义业务逻辑的接口，如图 2A 的 Request()。维护一个状态类的实例，该环境类生成一个 web app 对象。

基类（State）：定义一个接口如图 2A 的 Handle()，以封装与 Context 的一个状态类相关的行为。

各种状态类（ConcreteState），如图 2A 的 ConcreteStateA、ConcreteStateB 等，每个状态类继承上述基类，也即各个状态类为该基类的子类，每个状态类实现一个与 Context 的一个状态相关的行为。该状态类包括初始状态类、加载状态类、编辑状态类、保存状态类。

步骤 220，根据所述初始状态类将所述 web app 对象设置为初始化状态，在初始化状态的 web app 对象中执行初始化状态的相关行为，并根据第一状态转换逻辑将 web app 对象跳转为执行加载状态类；

在本申请实施例中，浏览器获从服务器侧获取到网页文档后，如 html（Hyper text Markup Language，超文本标记语言）文档后，执行所述文档，则生成的 web app 对象执行初始状态类，此时 web app 的属性为初始状态。

在该初始状态的 web app 对象中，执行初始状态类里封装的各种初始状态的相关行为，如前述为 web app 的各种变量分配空间、为各种变量赋予初始值等。并且在执行到某个具体行为后，跳转至执行加载状态类，如前述在
5 为所有变量赋予初始值之后，跳转至执行加载状态类。

结合图 2B，其介绍了本申请实施例的状态转换过程。

然后，web app 对象的属性更改为加载状态。

10 步骤 230，在加载状态的 web app 对象中，执行加载状态的相关行为，并根据第二状态转换逻辑将 web app 对象跳转为执行编辑状态类；

在 web app 的属性变更为加载状态之后，则执行加载状态类里封装的加载状态的相关行为。如前述载入各种信息，渲染 web app 的主界面等。并且在执行到某个具体行为后，跳转至执行编辑状态类，如前述在 web app 的主
15 界面渲染完毕之后跳转至执行编辑状态类。

然后，web app 对象的属性更改为编辑状态。

步骤 240，在编辑状态的 web app 对象中，执行编辑状态的相关行为，并根据第三状态转换逻辑将 web app 对象跳转为执行保存状态类；

在 web app 对象的属性变更为编辑状态之后，则执行加载状态类里封装
20 的编辑状态的相关行为。如前述启用键盘事件、启用鼠标事件、接收点击保存按钮的事件、web app 关闭等各种业务逻辑等。并且在执行到某个具体行为后，跳转至执行保存状态类，如接收到对保存按钮的点击事件后，跳转至执行保存状态类。

然后，web app 对象的属性更改为保存状态。

25 可以理解，在本申请实施例中，编辑状态类里可以封装各种与外界交互的行为，该行为可以根据实际需要封装，本申请不对其加以限制。

步骤 250，在保存状态的 web app 对象中，执行保存状态的相关行为，并根据第四状态转换逻辑将 web app 对象跳转为执行编辑状态类。

在 web app 的属性变更为保存状态之后，则执行保存状态类里封装的保存状态的相关行为。如前述保存当前 web app 中的内容至存储区域，取消保存操作（如，接收到 esc 事件）等等。并且在执行到某个具体行为后，跳转至执行保存状态类，如接在保存完毕之后或者接收到取消保存操作后，跳转至执行编辑状态类。

在状态类模式中，当 web app 一个状态调整至另外一个状态之后，前一个状态的信息则会被清除，如此后一个状态则不会与前一个状态产生交叉，不存在耦合。

在实际应用中，步骤 210 可离线进行预置，而步骤 220-250 在浏览器中执行。

本申请实施例预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类，即初始化状态类、加载状态类、编辑状态类、保存状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑，各个状态类之间是平行的，相互独立的，在 web app 执行时，web app 对象的每个状态类独立执行，不和其他的状态存在交叉。因此，本申请实施例对于 web app，能以完全状态模式进行运行，其代码中条件判断块小，分散到具体的状态类中，耦合性低。

在图 2 所示实施例的基础上，本申请还提供了一个优选的实施例：

参照图 3，示出了本申请的另一种执行完全状态模式的 web app 的方法实施例的步骤流程图，具体可以包括如下步骤：

步骤 310，预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类；所述状态类包括：初始状态类、加载状态类、编辑状态类、保存状态类，至少一个弹出窗口状态类；其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑；

在通常情况下，所有与外部交互的行为或者说业务逻辑，都被封装在编

辑状态类中,但实际上,这些行为之间从整体层面上很可能还是具备平行性,所谓平行性指的是各个状态的行为所处的层次是一样的,相互独立的、没有关联的,是可以根据不同的状态来决定到底走平行线的哪一条。那么本申请实施例,则可以根据行为之间的平行性,将与外部交互的行为拆分为多个状态类,比如,将在 web app 主页面本身中与外部交互的行为保留在编辑状态类中,将需要从 web app 主页面弹出的窗口中与外部交互的行为单独设置一个弹出窗口状态类,独立于该编辑状态类执行。

而由于可能从 web app 主页面中弹出的窗口有多个,本申请则可以对每个弹出窗口单独设置一个弹出窗口状态类。

相应的,在编辑状态类中,针对每个弹出窗口状态类,设置了相应的第五状态转换逻辑,其为主页面中接收到相应的操作之后,则将 web app 对象跳转至执行相应的弹出窗口状态类。

在本申请实施例的一种优选实施例中,本申请实施例还包括:

步骤 302,在编辑状态类中加入 sticky 属性,所述 sticky 属性用于 web app 对象的编辑状态类跳转为执行其他状态类时,保留编辑状态所对应的界面元素。

在实际应用中,因为一个状态转换到另外一个状态之后,前一个状态会清理,那么如果编辑状态转换到弹出窗口状态之后,编辑状态则会清理,则其主页面就不存在了,该种情况可能会误导用户该 web app 出现了问题,并且影响界面的可视度,不方便用户观看主页面上弹窗覆盖部分之外的信息。那么本申请则添加了 sticky 属性,该属性可以保证编辑状态跳转至其他状态类之后,主界面的 UI 界面还在显示。

需要说明的是,该 sticky 属性通过基类的接口实现,如果某个状态类为该 sticky 属性的值设置为 1,则不对该状态类进行清理,保留编辑状态所对应的界面元素,如保留图 3B 中 web app 的弹出窗口“新建 bucket”后的 web app 主页面。在本申请实施例中,对编辑状态类执行时将 sticky 属性设置为 1,其他状态类可将其值设置为 null。

当然，在本申请实施例中，也会定义前述环境类（Context）、基类（State）、以及各个状态类（ConcreteState）继承基类。

基类比如：

```

    public class State{
5      public function onEnter();
      public function onExit();
      // 切换状态， state 是目标状态名称， sticky 表示是否保留当前状态
      public function setState(state, sticky) {
      // setState 具体实现
10    }
    }

```

编辑类如：

```

    public class EditState extends State {
        .....
15    }

```

弹出窗口状态类如：

```

弹出窗口状态类 1: public class windowPopup1State extends State {
        .....
    }
20 弹出窗口状态类 2: public class windowPopup2State extends State {
        .....
    }

```

当然本申请实施例中，根据实际的业务逻辑，还可以设置其他状态类，比如预览状态栏等，本申请实施例不对其加以限制。

25 步骤 320，根据所述初始状态类将所述 web app 对象设置为初始化状态，在初始化状态的 web app 对象中执行初始化状态的相关行为，并根据第一状态转换逻辑将 web app 对象跳转为执行加载状态类；

本申请实施例结合图 3A 进行描述。

本步骤与图 2 实施例的步骤 220 类似，在此不再赘叙。

步骤 330，在加载状态的 web app 对象中，执行加载状态的相关行为，并根据第二状态转换逻辑将 web app 对象跳转为执行编辑状态类；

本步骤与图 2 实施例的步骤 330 类似，在此不再赘叙。

- 5 步骤 340，在编辑状态的 web app 对象中，执行编辑状态的相关行为，并根据各第五状态转换逻辑将 web app 对象跳转为执行相应的弹出窗口状态，根据第三状态转换逻辑将 web app 对象跳转为执行保存状态类；

如前述例子有在编辑状态之后有两个弹出窗口状态，即 windowPopup1State 和 windowPopup1State。

- 10 对于 windowPopup1State，如图 3B，假使其对应 bucket 管理栏，其第五状态转换逻辑可以为：当接收到用户对 bucket 管理栏的点击操作，则弹出新建 bucket 窗口，web app 对象的属性变为弹出窗口状态 windowPopup1State。假使其对应资源监控报表栏，其第五状态转换逻辑可以为：当接收到用户对资源监控报表栏的点击操作，则弹出资源监控报表窗口 web app 对象的属性变为弹出窗口状态 windowPopup1State。
- 15

当然，在前述在编辑状态类中加入 sticky 属性的情况下，步骤 340 还包括：

在 web app 对象的状态跳转到其他状态之后，保留编辑状态类对应的界面元素。

- 20 该界面元素可以理解为 web app 的主页面。

在本申请实施例的一种优选实施例中，本申请实施例还包括：

步骤 342 在编辑状态类中，判断是否将 web app 对象由编辑状态类跳转至弹出窗口状态类；如果是跳转至弹出窗口状态，则保留编辑状态所对应的界面元素。

- 25 在本申请实施例中，还可以在编辑状态类中在跳转到其他状态时，还可以判断跳转的对象，如果跳转的对象不是弹出窗口状态，则对编辑状态进行清理。比如判断跳转的对象是保存状态，则将 sticky 设置为 null，那么 web app 的状态变为保存状态后，则该编辑状态被清理。

如果跳转的对象是弹出窗口状态，则保留编辑状态所对应的界面元素。比如判断跳转的对象是保存状态，则将 sticky 设置为 1，那么 web app 的状态变为保存状态后，则该编辑状态不会被完全清理，会保留 web app 的主页面。

5 步骤 350 在弹出窗口状态类中，执行弹出窗口状态类的相关行为，并根据第七状态转换逻辑将 web app 对象跳转为执行编辑状态类。

在 web app 对象的属性变更为弹出窗口状态之后，则执行弹出窗口状态类里封装的弹出窗口状态的相关行为，如前图 3B 的 windowPopup1State extends State 下，使用蒙板将底部进行遮盖、不允许用户进行操作的行为，
10 允许用户进行输入、提交、取消等行为。

当然，在弹出窗口状态类中，还设置第七状态转换逻辑，用于将 web app 对象的状态从弹出窗口状态跳转为编辑状态，以执行编辑状态类的各种操作。如图 3B 中的弹出窗口状态 windowPopup1State extends State，该第七逻辑可以为，当接收到对提交按钮的点击操作或者接收到对取消按钮的点击操作，
15 则将 web app 对象跳转为执行编辑状态类。

步骤 360，在保存状态的 web app 对象中，执行保存状态的相关行为，并根据第四状态转换逻辑将 web app 对象跳转为执行编辑状态类。

本步骤与图 2 实施例的步骤 250 类似，在此不再赘叙。

在实际应用中，步骤 310 可离线进行预置，而不在 320-360 在浏览器中
20 执行。

本申请实施例预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑，各个状态类之间是平行的，相互独立的，在 web app 执行时，web app 对象的每个状态
25 类独立执行，不和其他的状态存在交叉。因此，本申请实施例对于 web app，能以完全状态模式进行运行，其代码中条件判断块小，分散到具体的状态类中，耦合性低。比如对 web app 的编辑和保存，分别设置编辑状态类和保存状态类，在编辑状态类中，直接启用其需要的键盘事件和鼠

标事件、禁用其不需要的键盘事件和鼠标事件，然后进入保存状态类后，在保存状态类中直接启用其需要的键盘事件和鼠标事件、禁用其不需要的键盘事件和鼠标事件。由于两个状态是独立的，如此 web app 的保存状态和编辑状态之间对于键盘事件和鼠标事件的逻辑的就解耦了，也不存在大块化的条件判断语句，将大块化的条件判断语句分支化到各个状态类中。

尤其是，本申请实施例将弹出窗口状态从编辑状态中独立出来，进一步对 web app 进行了解耦，下面以一个为代码示例对该提取出的弹出窗口状态能够进一步解耦的原理进行论述：

10 1、将各个弹出窗口的行为都放在一个编辑状态类中的伪代码如下：

```
public class EditState extends State{  
    private function activateMouse () {  
        // 在这里启用鼠标事件  
    }  
    private function activateKeyboard () {  
15         // 在这里启用键盘事件  
    }  
    private function deactivateMouse () {  
        // 在这里禁用鼠标事件  
20    }  
    private function deactivateKeyboard () {  
        // 在这里禁用键盘事件  
    }  
    public function onEnter() {  
25        this.activateMouse();  
        this.activateKeyboard();  
        // ... 这里有其他的业务逻辑  
        // 处理窗口 1 弹出
```

```

        this.windowPopup1 = function () {
            // 这里要禁用某些键盘和鼠标事件，然后再启用某些键盘和鼠
            标事件

            // 因此这里和外部代码产生了耦合
5          }
            // 处理窗口 2 弹出

            this.windowPopup2 = function () {
                // 这里要禁用某些键盘和鼠标事件，然后再启用某些键盘和鼠
                标事件

10             // 因此这里和外部代码产生了耦合

                }
            }

            public function onExit() {
                this.deactivateMouse();
                15         this.deactivateKeyboard();
            }
        }
    }

```

在通常情况下，将弹出窗口和其他业务逻辑一起放在编辑状态中，会使用蒙版将底部的内容区域遮盖，不允许用户操作（如图 3B 中 2 所述），但实际上用户可以通过 tab 导航快捷键的方式进行操作，此时 Web app 无法区分出这种操作是否合法。并且如上述示例，有两个弹出窗口，而两个窗口都要禁用某些键盘和鼠标事件，然后再启用某些键盘和鼠标事件，两者和外部代码之间就产生了耦合。特别是，窗口中需要处理的一般都是异步操作，那么代码中将会出现大量异步处理的代码，增大了代码耦合性，增加了开发难度。

而本申请实施例，对应上述伪代码，其将各个弹出窗口独立为弹出窗口状态类，其编辑状态类、弹出窗口状态类 1 和弹出窗口状态类 2 的伪代码如下：

(1)、编辑状态类

```
public class EditState extends State {  
    private function activateMouse ()  
    {  
5        // 在这里启用鼠标事件  
    }  
    private function activateKeyboard () {  
        // 在这里启用键盘事件  
    }  
10    private function deactivateMouse () {  
        // 在这里禁用鼠标事件  
    }  
    private function deactivateKeyboard () {  
        // 在这里禁用键盘事件  
15    }  
    public function onEnter() {  
        this.activateMouse();  
        this.activateKeyboard();  
        // ... 这里有其他的业务逻辑  
20        // 处理窗口 1 弹出  
        this.windowPopup1 = function () {  
            // 仅仅是切换到另外一个状态  
            this.setState(windowPopup1State)  
        }  
25        // 处理窗口 2 弹出  
        this.windowPopup2 = function () {  
            // 仅仅是切换到另外一个状态  
            this.setState(windowPopup2State)
```

```

        }
    }
    public function onExit() {
        this.deactivateMouse();
5        this.deactivateKeyboard();
    }
}

```

(2)、弹出窗口状态类 1

```

public class windowPopup1State extends State {
10    public function onEnter() {
        // 这里只需要启用所需的事件
    }
    public function onExit() {
        // 这里只需要禁用在 onEnter 中启用的事件
15    }
}

```

(3)、弹出窗口状态类 2

```

public class windowPopup1State extends State {
    public function onEnter() {
20        // 这里只需要启用所需的事件
    }
    public function onExit() {
        // 这里只需要禁用在 onEnter 中启用的事件
    }
25    }
}

```

在 (1) 的编辑状态类中，其处理弹出窗口 windowPopup1 的方法中，仅仅是将 wep app 对象的状态切换到 (2) 的弹出窗口状态 1；其处理弹出窗口 windowPopup2 的方法中，仅仅是将 wep app 对象的状态切换到 (3) 的弹出

窗口状态 2 中。在编辑状态中，并不执行弹出窗口的具体行为。

由于各个状态之间，因此可以很容易的屏蔽用户对底部内容区域的操作。并且，弹出窗口状态类（2）和弹出窗口状态类（3）与编辑状态类（1）之间的逻辑就解耦了，对鼠标和键盘事件的代码也解耦了，大大降低了开发
5 难度。

需要说明的是，对于方法实施例，为了简单描述，故将其都表述为一系列的动作组合，但是本领域技术人员应该知悉，本申请实施例并不受所描述的动作顺序的限制，因为依据本申请实施例，某些步骤可以采用其他顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书中所描述的实施例
10 均属于优选实施例，所涉及的动作并不一定是本申请实施例所必须的。

参照图 4，示出了本申请的一种执行完全状态模式的 web app 的装置实施例的结构框图，具体可以包括如下模块：

预置模块 410，用于预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类；其中，每个状态类包括
15 相应状态的相关行为，以及用于状态类跳转的状态转换逻辑；

状态执行模块 420，适于当 web app 对象执行某个状态类时，在该状态类的状态中，执行所述状态的相关行为，并根据该状态类的状态转换逻辑，将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的另一状态类。

本申请实施例预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类，每个状态类包括相应
20 状态的相关行为，以及用于状态类跳转的状态转换逻辑，各个状态类之间是平行的，相互独立的，在 web app 执行时，web app 对象的每个状态类独立执行，不和其他的状态存在交叉。因此，本申请实施例对于 web
25 app，能以完全状态模式进行运行，其代码中条件判断块小，即条件判断语句的数量少，分散到具体的状态类中，耦合性低。

参照图 5，示出了本申请的一种执行完全状态模式的 web app 的装置实

施例的结构框图，具体可以包括如下模块：

预置模块 510，用于预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类；所述状态类包括：初始状态类、加载状态类、编辑状态类、保存状态类；其中，每个状态类包括
5 相应状态的相关行为，以及用于状态类跳转的状态转换逻辑；

状态执行模块 520，具体包括：

初始化状态模块 521，用于根据所述初始状态类将所述 web app 对象设置为初始化状态，在初始化状态的 web app 对象中执行初始化状态的相关行为，并根据第一状态转换逻辑将 web app 对象跳转为执行加载状态类；

10 加载状态模块 522，用于在加载状态的 web app 对象中，执行加载状态的相关行为，并根据第二状态转换逻辑将 web app 对象跳转为执行编辑状态类；

编辑状态模块 523，用于在编辑状态的 web app 对象中，执行编辑状态的相关行为，并根据第三状态转换逻辑将 web app 对象跳转为执行保存状态
15 类；

保存状态模块 524，用于在保存状态的 web app 对象中，执行保存状态的相关行为，并根据第四状态转换逻辑将 web app 对象跳转为执行编辑状态类。

本申请实施例预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类，即初始化状态类、加载状态类、编辑状态类、保存状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑，各个状态类之间是平行的，相互独立的，在 web app 执行时，web app 对象的每个状态类独立执行，不和其他的状态存在交叉。因此，本申请实施例对于 web app，能以完全
20 状态模式进行运行，其代码中条件判断块小，分散到具体的状态类中，
25 耦合性低。

参照图 6，示出了本申请的另一种执行完全状态模式的 web app 的装置

实施例的结构框图，具体可以包括如下模块：

预置模块 610，用于预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类；所述状态类包括：初始状态类、加载状态类、编辑状态类、保存状态类、至少一个弹出窗口状态类；其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑；

在本申请的另外一个优选实施例中，所述预置模块 410 还用于：在编辑状态类中加入 sticky 属性，所述 sticky 属性用于 web app 对象的编辑状态类跳转为执行其他状态类时，保留编辑状态所对应的界面元素。

其中，每个所述状态转换逻辑根据相应状态下的一个或多个具体行为触发状态类的跳转。

状态执行模块 620，具体包括：

初始化状态模块 621，用于根据所述初始状态类将所述 web app 对象设置为初始化状态，在初始化状态的 web app 对象中执行初始化状态的相关行为，并根据第一状态转换逻辑将 web app 对象跳转为执行加载状态类；

加载状态模块 622，用于在加载状态的 web app 对象中，执行加载状态的相关行为，并根据各第五状态转换逻辑将 web app 对象跳转为执行相应的弹出窗口状态，根据第二状态转换逻辑将 web app 对象跳转为执行编辑状态类；

编辑状态模块 623，用于在编辑状态的 web app 对象中，执行编辑状态的相关行为，并根据第三状态转换逻辑将 web app 对象跳转为执行保存状态类；

在本申请的另外一个优选实施例中，所述编辑状态模块 440 还用于：

判断是否将 web app 对象由编辑状态类跳转至弹出窗口状态类；如果是跳转至弹出窗口状态，则保留编辑状态所对应的界面元素；

如果不是，则清理编辑状态类。

弹出窗口状态模块 624，用于在弹出窗口状态类中，执行弹出窗口状态类的相关行为，并根据第七状态转换逻辑将 web app 对象跳转为执行编辑状

态类。

保存状态模块 625，用于在保存状态的 web app 对象中，执行保存状态的相关行为，并根据第四状态转换逻辑将 web app 对象跳转为执行编辑状态类。

- 5 本申请实施例预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑，各个状态类之间是平行的，相互独立的，在 web app 执行时，web app 对象的每个状态类独立执行，不和其他的状态存在交叉。因此，本申请实施例对于 web
10 app，能以完全状态模式进行运行，其代码中条件判断块小，分散到具体的状态类中，耦合性低。

尤其是，本申请实施例将各弹出窗口状态从编辑状态中独立出来，进一步对 web app 进行了解耦，在弹出窗口状态下可以很容易的屏蔽用户对底部内容区域的操作，大大降低了开发难度。

15

参照图 7，示出了本申请的一种完全状态模式 web app 的实施例的结构框图，具体可以包括如下模块：

- 状态执行模块 710，适于当 web app 对象执行某个状态类时，在该状态类的状态中，执行所述状态的相关行为，并根据该状态类的状态转换逻辑，
20 将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的另一状态类；其中所述各个状态类根据 web app 的状态设置，所述 web app 的状态根据决定 web app 行为变化的属性进行划分；其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑。

- 当然，在本发明实施例中，可以预先根据决定 web app 行为变化的属性，
25 将 web app 划分为各个状态，并为各个状态设置对应的状态类，其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑，从而可以构建一个 web app。然后即可得到状态执行模块 710。

当然在状态执行模块 710，还可包括 web app 对象创建模块，用于在启

动 web app 时，创建 web app 对象。

还可包括起始状态赋予模块，用于为创建的 web app 对象赋予一个最初始的状态类。

其中，每个所述状态转换逻辑根据相应状态下的一个或多个具体行为触
5 发状态类的跳转。

在本申请另外一个优选的实施例中，所述状态类包括： 初始状态类、加载状态类、编辑状态类、保存状态类；

所述状态执行模块 710 包括：

初始化状态模块，用于根据所述初始状态类将所述 web app 对象设置为
10 初始化状态，在初始化状态的 web app 对象中执行初始化状态的相关行为，并根据第一状态转换逻辑将 web app 对象跳转为执行加载状态类；

加载状态模块，用于在加载状态的 web app 对象中，执行加载状态的相关行为，并根据第二状态转换逻辑将 web app 对象跳转为执行编辑状态类；

编辑状态模块，用于在编辑状态的 web app 对象中，执行编辑状态的相关行为，并根据第三状态转换逻辑将 web app 对象跳转为执行保存状态类；
15

保存状态模块，用于在保存状态的 web app 对象中，执行保存状态的相关行为，并根据第四状态转换逻辑将 web app 对象跳转为执行编辑状态类。

在本申请另外一个优选的实施例中，所述状态类还包括：至少一个弹出窗口状态类；

20 所述编辑状态模块，还用于：

根据各第五状态转换逻辑将 web app 对象跳转为执行相应的弹出窗口状态。

在本申请另外一个优选的实施例中，还包括：

弹出窗口状态模块，用于在弹出窗口状态类中，执行弹出窗口状态类的相关行为，并根据第七状态转换逻辑将 web app 对象跳转为执行编辑状态类。
25

在本申请另外一个优选的实施例中，在所述编辑状态类中还包括 sticky 属性，所述 sticky 属性用于 web app 对象的编辑状态类跳转为执行其他状态类时，保留编辑状态所对应的界面元素。

在本申请另外一个优选的实施例中，所述编辑状态模块还用于：

判断是否将 web app 对象由编辑状态类跳转至弹出窗口状态类；如果是跳转至弹出窗口状态，则保留编辑状态所对应的界面元素；

如果不是，则清理编辑状态类。

- 5 本申请实施例预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑，各个状态类之间是平行的，相互独立的，在 web app 执行时，web app 对象的每个状态类独立执行，不和其他的状态存在交叉。因此，本申请实施例对于 web
10 app，能以完全状态模式进行运行，其代码中条件判断块小，即条件判断语句的数量少，分散到具体的状态类中，耦合性低。

- 比如对 web app 的编辑和保存，分别设置编辑状态类和保存状态类，在编辑状态类中，直接启用其需要的键盘事件和鼠标事件、禁用其不需要的键盘事件和鼠标事件，然后进入保存状态类后，在保存状态类中
15 直接启用其需要的键盘事件和鼠标事件、禁用其不需要的键盘事件和鼠标事件。由于两个状态是独立的，如此 web app 的保存状态和编辑状态之间对于键盘事件和鼠标事件的逻辑的就解耦了，也不存在大块化的条件判断语句，将大块化的条件判断语句分支化到各个状态类中。

- 对于装置实施例而言，由于其与方法实施例基本相似，所以描述的比较
20 简单，相关之处参见方法实施例的部分说明即可。

 本说明书中的各个实施例均采用递进的方式描述，每个实施例重点说明的都是与其他实施例的不同之处，各个实施例之间相同相似的部分互相参见即可。

- 25 本领域内的技术人员应明白，本申请实施例的实施例可提供为方法、装置、或计算机程序产品。因此，本申请实施例可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请实施例可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介

质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

在一个典型的配置中，所述计算机设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM) 和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不
10 限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他
15 非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括非持续性的电脑可读媒体(transitory media)，如调制的数据信号和载波。

本申请实施例是参照根据本申请实施例的方法、终端设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、
20 专用计算机、嵌入式处理机或其他可编程数据处理终端设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理终端设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。
25

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理终端设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个

流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能。

5 这些计算机程序指令也可装载到计算机或其他可编程数据处理终端设备上,使得在计算机或其他可编程终端设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程终端设备上执行的指令提供用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的步骤。

10 尽管已描述了本申请实施例的优选实施例,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例做出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本申请实施例范围的所有变更和修改。

15 最后,还需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者终端设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者终端设备所固有的要素。在没有更多限制的情况下,由语句“包括一个.....”限定的要素,并不排除在包括所述要素的过程、方法、物品或者终端设备中还存在另外的相同要素。

20

以上对本申请所提供的执行完全状态模式的 web app 的方法、执行完全状态模式的 web app 的装置、一种完全状态模式 APP,进行了详细介绍,本文中应用了具体个例对本申请的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本申请的方法及其核心思想;同时,对于本领域的一般技术人员,依据本申请的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本申请的限制。

25

权 利 要 求 书

1、一种执行完全状态模式的 web app 的方法，其特征在于，包括：

预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类；其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑；

当 web app 对象执行某个状态类时，在该状态类的状态中，执行所述状态的相关行为，并根据该状态类的状态转换逻辑，将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的另一状态类。

2、根据权利要求 1 所述的方法，其特征在于，所述状态类包括：初始状态类、加载状态类、编辑状态类、保存状态类；

所述当 web app 对象执行某个状态类时，在该状态类的状态中，执行所述状态的相关行为，并根据该状态类的状态转换逻辑，将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的另一状态类的步骤包括：

根据所述初始状态类将所述 web app 对象设置为初始化状态，在初始化状态的 web app 对象中执行初始化状态的相关行为，并根据第一状态转换逻辑将 web app 对象跳转为执行加载状态类；

在加载状态的 web app 对象中，执行加载状态的相关行为，并根据第二状态转换逻辑将 web app 对象跳转为执行编辑状态类；

在编辑状态的 web app 对象中，执行编辑状态的相关行为，并根据第三状态转换逻辑将 web app 对象跳转为执行保存状态类；

在保存状态的 web app 对象中，执行保存状态的相关行为，并根据第四状态转换逻辑将 web app 对象跳转为执行编辑状态类。

3、根据权利要求 2 所述的方法，其特征在于，

所述状态类还包括：至少一个弹出窗口状态类；

在编辑状态的 web app 对象中，还包括：

根据各第五状态转换逻辑将 web app 对象跳转为执行相应的弹出窗口状态。

4、根据权利要求3所述的方法，其特征在于，还包括：

在弹出窗口状态类中，执行弹出窗口状态类的相关行为，并根据第七状态转换逻辑将 web app 对象跳转为执行编辑状态类。

5、根据权利要求3或4所述的方法，其特征在于，还包括：

5 在编辑状态类中加入 sticky 属性，所述 sticky 属性用于 web app 对象的编辑状态类跳转为执行其他状态类时，保留编辑状态所对应的界面元素。

6、根据权利要求4所述的方法，其特征在于，在编辑状态中，还包括：

判断是否将 web app 对象由编辑状态类跳转至弹出窗口状态类；如果是跳转至弹出窗口状态，则保留编辑状态所对应的界面元素；

10 如果不是，则清理编辑状态类。

7、根据权利要求1所述的方法，其特征在于，每个所述状态转换逻辑根据相应状态下的一个或多个具体行为触发状态类的跳转。

8、一种执行完全状态模式的 web app 的装置，其特征在于，包括：

15 预置模块，用于预先根据决定 web app 行为变化的属性，将 web app 划分为各个状态，并为各个状态设置对应的状态类；其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑；

20 状态执行模块，适于当 web app 对象执行某个状态类时，在该状态类的状态中，执行所述状态的相关行为，并根据该状态类的状态转换逻辑，将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的另一状态类。

9、根据权利要求8所述的装置，其特征在于，所述状态类包括： 初始状态类、加载状态类、编辑状态类、保存状态类；

所述状态执行模块包括：

25 初始化状态模块，用于根据所述初始状态类将所述 web app 对象设置为初始化状态，在初始化状态的 web app 对象中执行初始化状态的相关行为，并根据第一状态转换逻辑将 web app 对象跳转为执行加载状态类；

加载状态模块，用于在加载状态的 web app 对象中，执行加载状态的相关行为，并根据第二状态转换逻辑将 web app 对象跳转为执行编辑状态类；

编辑状态模块，用于在编辑状态的 web app 对象中，执行编辑状态的相关行为，并根据第三状态转换逻辑将 web app 对象跳转为执行保存状态类；

保存状态模块，用于在保存状态的 web app 对象中，执行保存状态的相关行为，并根据第四状态转换逻辑将 web app 对象跳转为执行编辑状态类。

5 10、根据权利要求 8 所述的装置，其特征在于，

所述状态类还包括：至少一个弹出窗口状态类；

所述编辑状态模块，还用于：

根据各第五状态转换逻辑将 web app 对象跳转为执行相应的弹出窗口状态。

10 11、根据权利要求 10 所述的装置，其特征在于，所述状态执行模块还包括：

弹出窗口状态模块，用于在弹出窗口状态类中，执行弹出窗口状态类的相关行为，并根据第七状态转换逻辑将 web app 对象跳转为执行编辑状态类。

15 12、根据权利要求 10 或 11 所述的装置，其特征在于，所述预置模块还用于：

在编辑状态类中加入 sticky 属性，所述 sticky 属性用于 web app 对象的编辑状态类跳转为执行其他状态类时，保留编辑状态所对应的界面元素。

13、根据权利要求 11 所述的装置，其特征在于，所述编辑状态模块还用于：

20 判断是否将 web app 对象由编辑状态类跳转至弹出窗口状态类；如果是跳转至弹出窗口状态，则保留编辑状态所对应的界面元素；

如果不是，则清理编辑状态类。

14、根据权利要求 8 所述的装置，其特征在于，每个所述状态转换逻辑根据相应状态下的一个或多个具体行为触发状态类的跳转。

25 15、一种完全状态模式 web app，其特征在于，包括：

状态执行模块，适于当 web app 对象执行某个状态类时，在该状态类的状态中，执行所述状态的相关行为，并根据该状态类的状态转换逻辑，将 web app 对象从当前执行的状态类，跳转为执行与所述状态转换逻辑对应的

另一状态类；其中所述各个状态类根据 web app 的状态设置，所述 web app 的状态根据决定 web app 行为变化的属性进行划分；其中，每个状态类包括相应状态的相关行为，以及用于状态类跳转的状态转换逻辑。

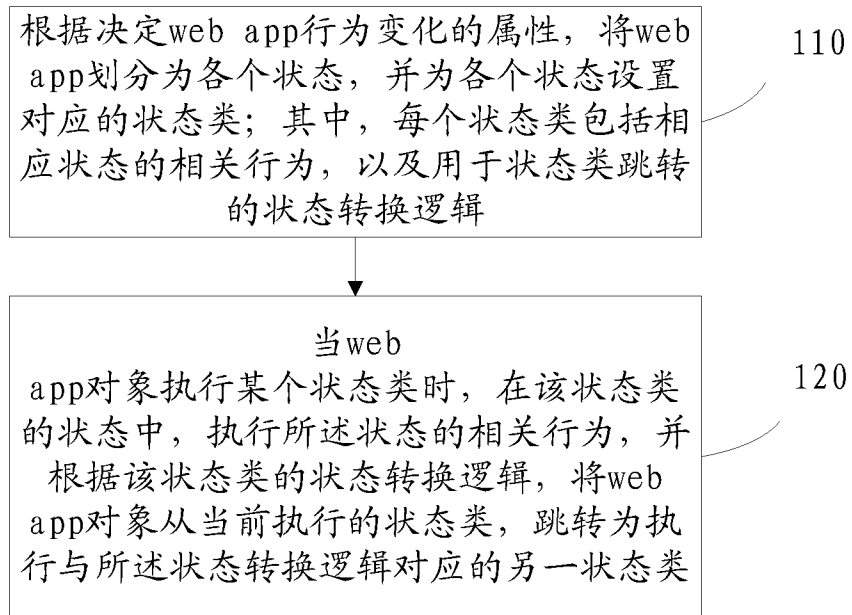


图 1

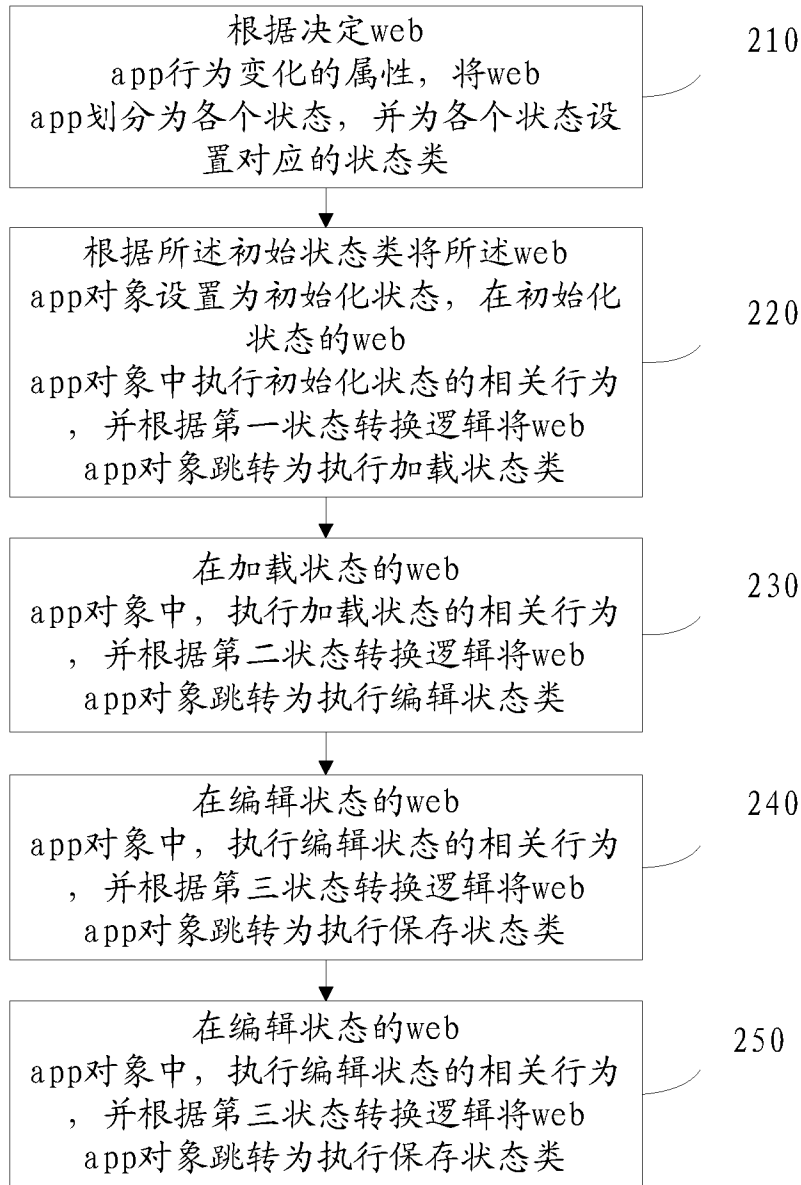


图 2

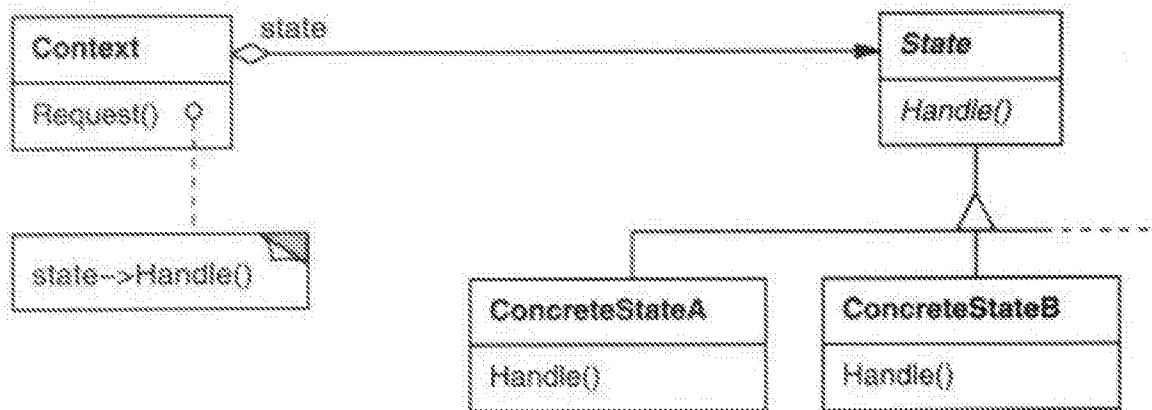


图 2A

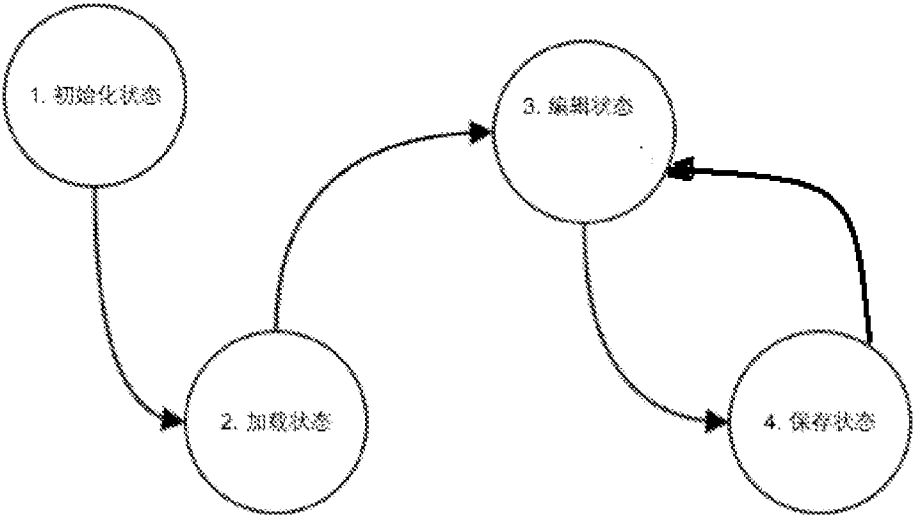


图 2B

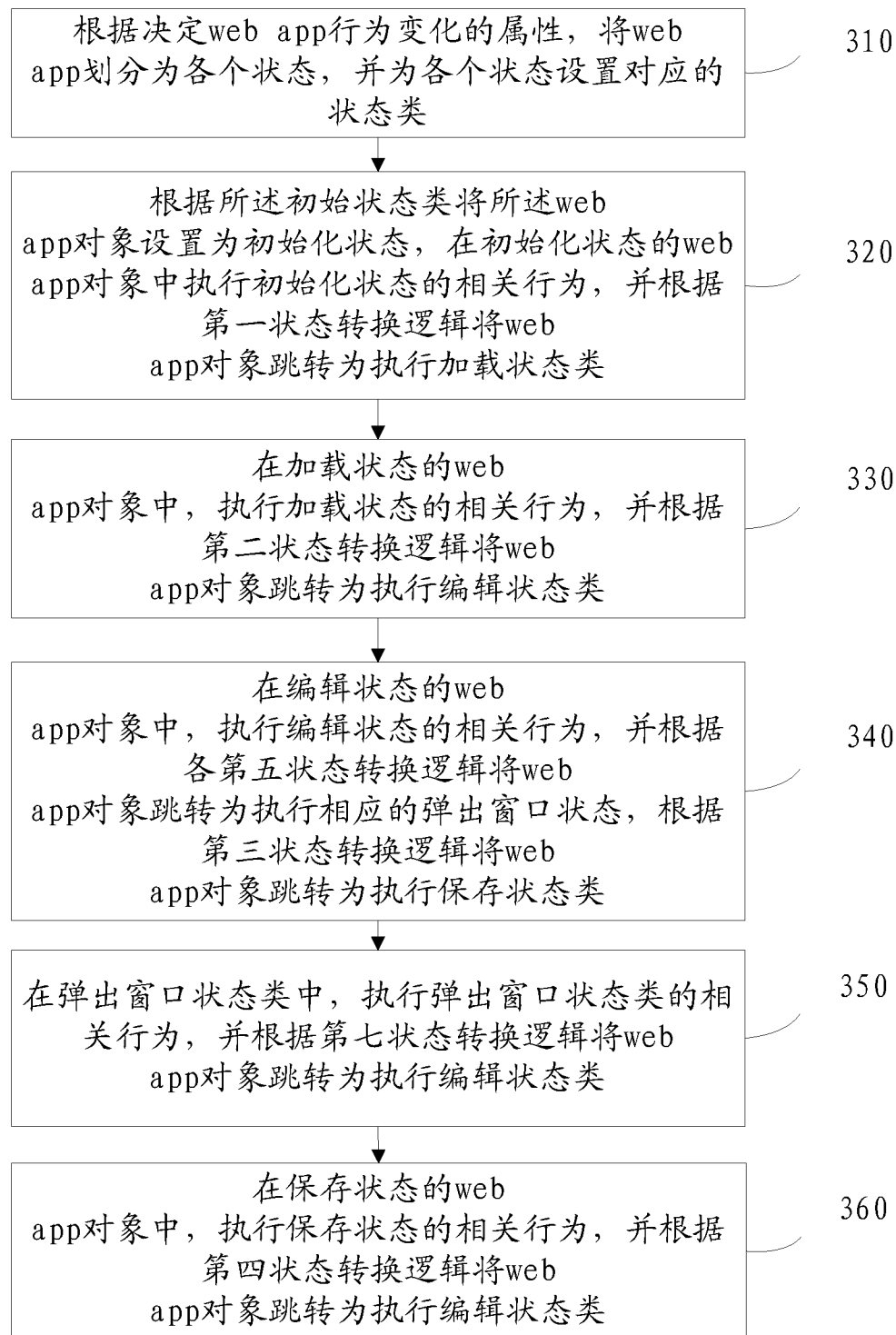


图 3

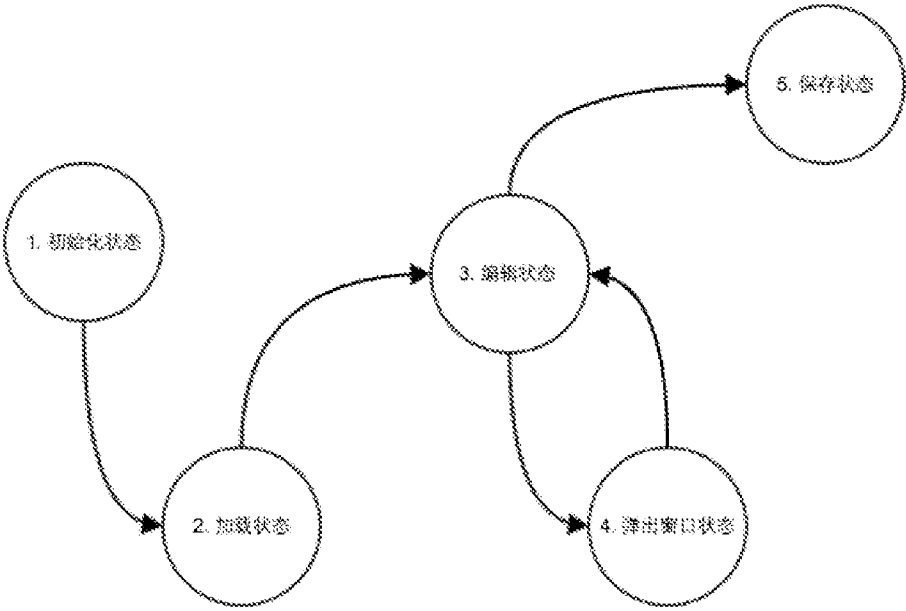


图 3A

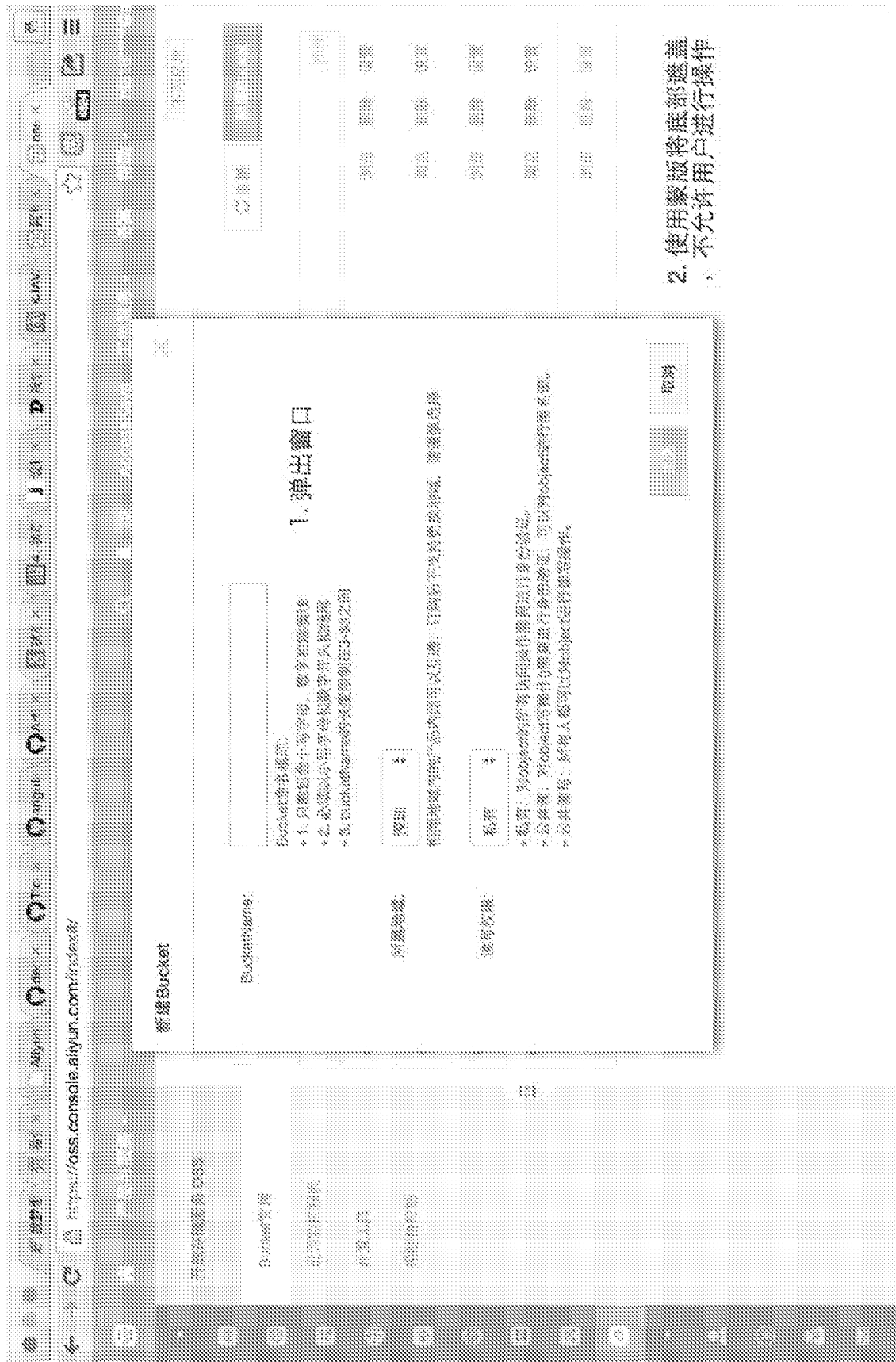


图 3B

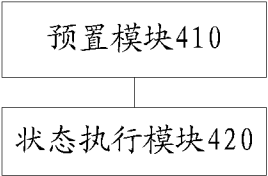


图 4

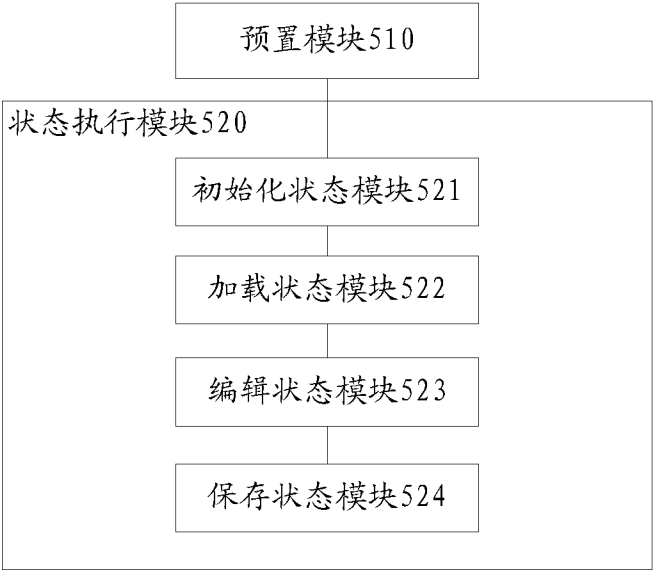


图 5



图 6

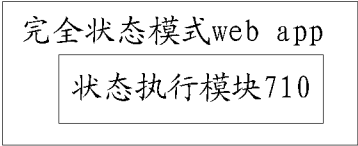


图 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2016/098813

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/44 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS, DWPI, CNKI, SIPOABS: web, app, application, catalog, class, property, state, web page

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2015234798 A1 (GOOGLE INCORPORATION) 20 August 2015 (20.08.2015) the whole document	1-15
A	CN 104216715 A (SHENZHEN QIANHAI NUODE CONSULTATION SERVICE CO., LTD.) 17 December 2014 (17.12.2014) the whole document	1-15
A	CN 104516738 A (BEIJING JINHER SOFTWARE CO., LTD.) 15 April 2015 (15.04.2015) the whole document	1-15

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	
“A” document defining the general state of the art which is not considered to be of particular relevance	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
“E” earlier application or patent but published on or after the international filing date	“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
“O” document referring to an oral disclosure, use, exhibition or other means	“&” document member of the same patent family
“P” document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 29 November 2016	Date of mailing of the international search report 23 December 2016
Name and mailing address of the ISA State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer ZHANG, Yiliang Telephone No. (86-10) 62089422

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/CN2016/098813

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
US 2015234798 A1	20 August 2015	None	
CN 104216715 A	17 December 2014	None	
CN 104516738 A	15 April 2015	US 2016170712 A1	16 June 2016

A. 主题的分类 G06F 9/44 (2006.01) i 按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类														
B. 检索领域 检索的最低限度文献 (标明分类系统和分类号) G06F 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用)) CNABS, DWPI, CNKI, SIPOABS: 网页, 应用, 类, 属性, 状态, web, app, application, catalog, class, property, state														
C. 相关文件 <table border="1"> <thead> <tr> <th>类 型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>US 2015234798 A1 (GOOGLE INC.) 2015年 8月 20日 (2015 - 08 - 20) 全文</td> <td>1-15</td> </tr> <tr> <td>A</td> <td>CN 104216715 A (深圳前海诺德咨询服务有限公司) 2014年 12月 17日 (2014 - 12 - 17) 全文</td> <td>1-15</td> </tr> <tr> <td>A</td> <td>CN 104516738 A (北京金和软件股份有限公司) 2015年 4月 15日 (2015 - 04 - 15) 全文</td> <td>1-15</td> </tr> </tbody> </table>			类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求	A	US 2015234798 A1 (GOOGLE INC.) 2015年 8月 20日 (2015 - 08 - 20) 全文	1-15	A	CN 104216715 A (深圳前海诺德咨询服务有限公司) 2014年 12月 17日 (2014 - 12 - 17) 全文	1-15	A	CN 104516738 A (北京金和软件股份有限公司) 2015年 4月 15日 (2015 - 04 - 15) 全文	1-15
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求												
A	US 2015234798 A1 (GOOGLE INC.) 2015年 8月 20日 (2015 - 08 - 20) 全文	1-15												
A	CN 104216715 A (深圳前海诺德咨询服务有限公司) 2014年 12月 17日 (2014 - 12 - 17) 全文	1-15												
A	CN 104516738 A (北京金和软件股份有限公司) 2015年 4月 15日 (2015 - 04 - 15) 全文	1-15												
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。														
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件														
国际检索实际完成的日期 2016年 11月 29日		国际检索报告邮寄日期 2016年 12月 23日												
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10) 62019451		受权官员 张一良 电话号码 (86-10) 62089422												

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2016/098813

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
US	2015234798	A1	2015年 8月 20日	无	
CN	104216715	A	2014年 12月 17日	无	
CN	104516738	A	2015年 4月 15日	US 2016170712 A1	2016年 6月 16日