



(12) 发明专利

(10) 授权公告号 CN 111399806 B

(45) 授权公告日 2023.09.01

(21) 申请号 201910004541.4	US 2016077819 A1,2016.03.17
(22) 申请日 2019.01.03	CN 107733977 A,2018.02.23
(65) 同一申请的已公布的文献号 申请公布号 CN 111399806 A	CN 108664296 A,2018.10.16
(43) 申请公布日 2020.07.10	CN 1828595 A,2006.09.06
(73) 专利权人 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	KR 20020028963 A,2002.04.17
(72) 发明人 董天成	US 2010161777 A1,2010.06.24
(74) 专利代理机构 北京品源专利代理有限公司 11332 专利代理师 孟金喆	US 2006010425 A1,2006.01.12
(51) Int. Cl. G06F 8/20 (2018.01) G06F 16/28 (2019.01)	CN 104657142 A,2015.05.27
(56) 对比文件 CN 108694073 A,2018.10.23	CN 101454766 A,2009.06.10
CN 109117119 A,2019.01.01	CN 107451296 A,2017.12.08
	CN 108319575 A,2018.07.24
	CN 105022630 A,2015.11.04
	CN 107944266 A,2018.04.20
	US 2009064192 A1,2009.03.05
	CN 103544013 A,2014.01.29
	CN 101655789 A,2010.02.24
	US 2009157728 A1,2009.06.18
	US 2018052992 A1,2018.02.22
	曾立鵬.Android系统应用程序组件安全性 分析.《软件》.2014, 2014-06-19.

审查员 周佳利

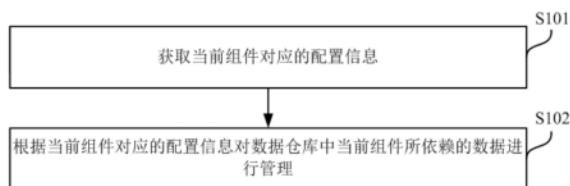
权利要求书2页 说明书10页 附图7页

(54) 发明名称

一种组件管理方法、装置、电子设备及存储介质

(57) 摘要

本发明实施例公开了一种组件管理方法、装置、电子设备及存储介质。所述方法包括：获取当前组件对应的配置信息；根据所述当前组件对应的配置信息对数据仓库中所述当前组件所依赖的数据进行管理。不仅可以减少组件管理成本，还可以提高组件管理效率。



1. 一种组件管理方法,其特征在于,所述方法包括:

获取当前组件对应的配置信息;

根据所述当前组件对应的配置信息对数据仓库中所述当前组件所依赖的数据进行管理,包括:

若所述当前组件对应的配置信息中存在名称name属性的组件配置,则根据所述name属性的组件配置在所述数据仓库中创建所述当前组件所依赖的数据;

若所述当前组件对应的配置信息中被嵌入基于Javascript语法的运行沙箱,则通过所述基于Javascript语法的运行沙箱计算所述当前组件对应的目标值;

在所述数据仓库中将所述当前组件对应的目标值传递给所述当前组件;

若所述数据仓库数据出现变更,触发容器组件更新,读取所述数据仓库数据,获取最新的数据上下文,更新无状态组件,解析所述无状态组件的配置中的属性值,若所述无状态组件的配置中的属性值含有ExpressionString,则导入数据上下文,执行ExpressionString,获取执行结果,将ExpressionString替换成所述执行结果;若所述无状态组件的配置中的属性值不含有ExpressionString,则通过无状态组件获取处理后的配置,实现更新。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

若所述当前组件对应的关联组件产生新的数据,则根据所述name属性的组件配置将所述新的数据写入到所述数据仓库中;

若预先定义的点击操作或者键盘操作被用户触发,则在所述当前组件对应的配置信息中添加与其对应的事件绑定。

3. 根据权利要求2所述的方法,其特征在于,所述将所述新的数据写入到所述数据仓库中,包括:

通过容器组件的响应式接口向所述数据仓库发送超文本传输协议HTTP请求;

通过容器组件的响应式接口接收所述数据仓库返回的HTTP响应;

根据所述数据仓库返回的HTTP响应将所述新的数据写入到所述数据仓库中。

4. 根据权利要求2所述的方法,其特征在于,所述在所述当前组件对应的配置信息中添加与其对应的事件绑定,包括:

若产品定义的软件操作无法使用单个子任务进行操作,则将所述产品定义的软件操作划分为多个软件子操作;

各个软件子操作由各个子任务按照预先设置的执行顺序进行操作,并在所述当前组件对应的配置信息中添加各个子任务对应的事件绑定。

5. 一种组件管理装置,其特征在于,所述装置包括:获取模块和管理模块;其中,

所述获取模块,用于获取当前组件对应的配置信息;

所述管理模块,用于根据所述当前组件对应的配置信息对数据仓库中所述当前组件所依赖的数据进行管理;

其中,所述管理模块,具体用于若所述当前组件对应的配置信息中存在name属性的组件配置,则根据所述name属性的组件配置在所述数据仓库中创建所述当前组件所依赖的数据;若所述当前组件对应的配置信息中被嵌入基于Javascript语法的运行沙箱,则通过所述基于Javascript语法的运行沙箱计算所述当前组件对应的目标值;在所述数据仓库中将所述当前组件对应的目标值传递给所述当前组件;若所述数据仓库数据出现变更,触发容

器组件更新,读取所述数据仓库数据,获取最新的数据上下文,更新无状态组件,解析所述无状态组件的配置中的属性值,若所述无状态组件的配置中的属性值含有ExpressionString,则导入数据上下文,执行ExpressionString,获取执行结果,将ExpressionString替换成所述执行结果;若所述无状态组件的配置中的属性值不含有ExpressionString,则通过无状态组件获取处理后的配置,实现更新。

6. 根据权利要求5所述的装置,其特征在于,所述管理模块,还用于若所述当前组件对应的关联组件产生新的数据,则根据所述name属性的组件配置将所述新的数据写入到所述数据仓库中;若预先定义的点击操作或者键盘操作被用户触发,则在所述当前组件对应的配置信息中添加与其对应的事件绑定。

7. 根据权利要求6所述的装置,其特征在于:

所述管理模块,具体用于通过容器组件的响应式接口向所述数据仓库发送HTTP请求;通过容器组件的响应式接口接收所述数据仓库返回的HTTP响应;根据所述数据仓库返回的HTTP响应将所述新的数据写入到所述数据仓库中。

8. 根据权利要求6所述的装置,其特征在于:

所述管理模块,具体用于若产品定义的软件操作无法使用单个子任务进行操作,则将所述产品定义的软件操作划分为多个软件子操作;各个软件子操作由各个子任务按照预先设置的执行顺序进行操作,并在所述当前组件对应的配置信息中添加各个子任务对应的事件绑定。

9. 一种电子设备,其特征在于,包括:

一个或多个处理器;

存储器,用于存储一个或多个程序,

当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如权利要求1至4中任一项所述的组件管理方法。

10. 一种存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现如权利要求1至4中任一项所述的组件管理方法。

一种组件管理方法、装置、电子设备及存储介质

技术领域

[0001] 本发明实施例涉及计算机技术领域,尤其涉及一种组件管理方法、装置、电子设备及存储介质。

背景技术

[0002] 现在的前端工程师在开发平台系统页面的时候,一般都是使用流行的前端框架,例如:React.js,Vue.js,然后通过编写调用框架API、调用接口以及开发一些特定的业务逻辑代码来完成系统中每个页面的开发。在现有的网页开发过程中,前端工程师不仅要用代码来实现特殊的业务逻辑,还需要基于当前所应用的框架,通过编写代码实现对组件状态的管理,编写每个组件生命周期内的一些实现,完成页面组件在用户交互的时候对状态的操作。如果组件需要调用接口,前端工程师还需要在组件的一些生命周期函数内,调用Ajax来发送请求获取数据,并手动实现逻辑将接口返回的数据合并到页面的状态中。因此前端工程师在页面的开发过程中,需要花费很多时间来维护整个应用的状态。

[0003] 此外,复杂的业务系统会依赖非常多的数据接口。每一个数据接口都依赖状态中的某一些字段。在现有的网页开发过程中,前端工程师都需要手动编写代码,来监听某一些字段的更新,以完成重新请求数据的操作。因此,需要通过一定的方式来尽可能地克服这一问题,但是在现有技术中还没有一种有效的解决方式。

发明内容

[0004] 有鉴于此,本发明实施例提供一种组件管理方法、装置、电子设备及存储介质,不仅可以减少组件管理成本,还可以提高组件管理效率。

[0005] 第一方面,本发明实施例提供了一种组件管理方法,所述方法包括:

[0006] 获取当前组件对应的配置信息;

[0007] 根据所述当前组件对应的配置信息对数据仓库中所述当前组件所依赖的数据进行管理。

[0008] 在上述实施例中,所述根据所述当前组件对应的配置信息对数据仓库中所述当前组件所依赖的数据进行管理,包括:

[0009] 若所述当前组件对应的配置信息中被嵌入基于Javascript语法的运行沙箱,则通过所述基于Javascript语法的运行沙箱计算所述当前组件对应的目标值;

[0010] 在所述数据仓库中将所述当前组件对应的目标值传递给所述当前组件。

[0011] 在上述实施例中,所述根据所述当前组件对应的配置信息对数据仓库中所述当前组件所依赖的数据进行管理,包括:

[0012] 若所述当前组件对应的配置信息中存在名称name属性的组件配置,则根据所述name属性的组件配置在所述数据仓库中创建所述当前组件所依赖的数据。

[0013] 在上述实施例中,所述方法还包括:

[0014] 若所述当前组件对应的关联组件产生新的数据,则根据所述name属性的组件配置

将所述新的数据写入到所述数据仓库中；

[0015] 若预先定义的点击操作或者键盘操作被用户触发，则在所述当前组件对应的配置信息中添加与其对应的事件绑定。

[0016] 在上述实施例中，所述将所述新的数据写入到所述数据仓库中，包括：

[0017] 通过所述容器组件的响应式接口向所述数据仓库发送超文本传输协议HTTP请求；

[0018] 通过所述容器组件的响应式接口接收所述数据仓库返回的HTTP响应；

[0019] 根据所述数据仓库返回的HTTP响应将所述新的数据写入到所述数据仓库中。

[0020] 在上述实施例中，所述在所述当前组件对应的配置信息中添加与其对应的事件绑定，包括：

[0021] 若产品定义的软件操作无法使用单个子任务进行操作，则将所述产品定义的软件操作划分为多个软件子操作；

[0022] 各个软件子操作由各个子任务按照预先设置的执行顺序进行操作，并在所述当前组件对应的配置信息中添加各个子任务对应的事件绑定。

[0023] 第二方面，本发明实施例提供了一种组件管理装置，所述装置包括：获取模块和管理模块；其中，

[0024] 所述获取模块，用于获取当前组件对应的配置信息；

[0025] 所述管理模块，用于根据所述当前组件对应的配置信息对数据仓库中所述当前组件所依赖的数据进行管理。

[0026] 在上述实施例中，所述管理模块，具体用于若所述当前组件对应的配置信息中被嵌入基于Javascript语法的运行沙箱，则通过所述基于Javascript语法的运行沙箱计算所述当前组件对应的目标值；在所述数据仓库中将所述当前组件对应的目标值传递给所述当前组件。

[0027] 在上述实施例中，所述管理模块，具体用于若所述当前组件对应的配置信息中存在name属性的组件配置，则根据所述name属性的组件配置在所述数据仓库中创建所述当前组件所依赖的数据。

[0028] 在上述实施例中，所述管理模块，还用于若所述当前组件对应的关联组件产生新的数据，则根据所述name属性的组件配置将所述新的数据写入到所述数据仓库中；若预先定义的点击操作或者键盘操作被用户触发，则在所述当前组件对应的配置信息中添加与其对应的事件绑定。

[0029] 在上述实施例中，所述管理模块，具体用于通过所述容器组件的响应式接口向所述数据仓库发送HTTP请求；通过所述容器组件的响应式接口接收所述数据仓库返回的HTTP响应；根据所述数据仓库返回的HTTP响应将所述新的数据写入到所述数据仓库中。

[0030] 在上述实施例中，所述管理模块，具体用于若产品定义的软件操作无法使用单个子任务进行操作，则将所述产品定义的软件操作划分为多个软件子操作；各个软件子操作由各个子任务按照预先设置的执行顺序进行操作，并在所述当前组件对应的配置信息中添加各个子任务对应的事件绑定。

[0031] 第三方面，本发明实施例提供了一种电子设备，包括：

[0032] 一个或多个处理器；

[0033] 存储器，用于存储一个或多个程序，

[0034] 当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现本发明任意实施例所述的组件管理方法。

[0035] 第四方面,本发明实施例提供了一种存储介质,其上存储有计算机程序,该程序被处理器执行时实现本发明任意实施例所述的组件管理方法。

[0036] 本发明实施例提出了一种组件管理方法、装置、电子设备及存储介质,先获取当前组件对应的配置信息;然后根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。也就是说,在本发明的技术方案中,可以根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。而在现有的组件管理方法中,前端工程师都需要手动编写代码来实现组件管理。因此,和现有技术相比,本发明实施例提出的组件管理方法、装置、电子设备及存储介质,不仅可以减少组件管理成本,还可以提高组件管理效率;并且,本发明实施例的技术方案实现简单方便、便于普及,适用范围更广。

附图说明

[0037] 图1为本发明实施例一提供的组件管理方法的流程示意图;

[0038] 图2为本发明实施例二提供的组件管理方法的流程示意图;

[0039] 图3为本发明实施例三提供的组件管理方法的流程示意图;

[0040] 图4为本发明实施例三提供的组件生成方法的流程示意图;

[0041] 图5为本发明实施例三提供的容器组件通过ExpressionString动态管理组件的方法流程示意图;

[0042] 图6为本发明实施例三提供的含有name属性的无状态组件更新方法的流程示意图;

[0043] 图7为本发明实施例三提供的组件事件触发的方法流程示意图;

[0044] 图8为本发明实施例四提供的组件管理装置的结构示意图;

[0045] 图9为本发明实施例五提供的电子设备的结构示意图。

具体实施方式

[0046] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是,此处所描述的具体实施例仅仅用于解释本发明,而非对本发明的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与本发明相关的部分而非全部内容。

[0047] 实施例一

[0048] 图1为本发明实施例一提供的组件管理方法的流程示意图,该方法可以由组件管理装置或者电子设备来执行,该装置或者电子设备可以由软件和/或硬件的方式实现,该装置或者电子设备可以集成在任何具有网络通信功能的智能设备中。如图1所示,组件管理方法可以包括以下步骤:

[0049] S101、获取当前组件对应的配置信息。

[0050] 在本发明的具体实施例中,电子设备可以获得当前组件对应的配置信息。在RCRE中,组件又分为两种类型:无状态组件和有状态组件。在RCRE中,除了容器container组件之外的其他组件都是无状态组件。开发者只需要把container组件和其他组件分开来看待即可。无状态组件和有状态组件的唯一区别就在于无状态组件不会持有私有的数据模型,它

所需要的任何数据,都会通过一种特殊的方式,从父级组件中的有状态组件上获取。不管是无状态组件还是有状态组件,任何一个组件都必然会包含类型type这个属性。type属性代表着组件的类型。比如type为container的组件就是container组件,type为text的组件就是text文本组件。有状态组件不仅能够持有自己的私有数据模型,也可以通过dataProvider配置,来指定通过什么样的方式来获取新的数据,再合并到原有的数据模型中。dataProvider支持多种获取数据的方法,不同获取数据方式会通过mode属性来进行区分,不同类型的获取数据方式最终都会把获取到的数据合并到原有的数据模型中。因此,在本发明的具体实施例中,电子设备可以预先设置当前组件对应的配置信息,然后在当前组件的管理过程中就可以直接获取当前组件对应的配置信息。

[0051] S102、根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。

[0052] 在本发明的具体实施例中,电子设备可以根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。具体地,若当前组件对应的配置信息中被嵌入基于Javascript语法的运行沙箱,则电子设备通过基于Javascript语法的运行沙箱计算当前组件对应的目标值;然后在数据仓库中将当前组件对应的目标值传递给当前组件。较佳地,在本发明的具体实施例中,若当前组件对应的配置信息中存在name属性的组件配置,则电子设备可以根据name属性的组件配置在数据仓库中创建当前组件所依赖的数据。

[0053] 本发明实施例提出的组件管理方法,先获取当前组件对应的配置信息;然后根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。也就是说,在本发明的技术方案中,可以根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。而在现有的组件管理方法中,前端工程师都需要手动编写代码来实现组件管理。因此,和现有技术相比,本发明实施例提出的组件管理方法,不仅可以减少组件管理成本,还可以提高组件管理效率;并且,本发明实施例的技术方案实现简单方便、便于普及,适用范围更广。

[0054] 实施例二

[0055] 图2为本发明实施例二提供的组件管理方法的流程示意图。如图2所示,组件管理方法可以包括以下步骤:

[0056] S201、获取当前组件对应的配置信息。

[0057] 在本发明的具体实施例中,电子设备可以获取当前组件对应的配置信息。在RCRE中,组件又分为两种类型:无状态组件和有状态组件。在RCRE中,除了container组件之外的其他组件都是无状态组件。开发者只需要把container组件和其他组件分开来看待即可。无状态组件和有状态组件的唯一区别就在于无状态组件不会持有私有的数据模型,它所需要的任何数据,都会通过一种特殊的方式,从父级组件中的有状态组件上获取。不管是无状态组件还是有状态组件,任何一个组件都必然会包含type这个属性。type属性代表着组件的类型。比如type为container的组件就是container组件,type为text的组件就是text文本组件。有状态组件不仅能够持有自己的私有数据模型,也可以通过dataProvider配置,来指定通过什么样的方式来获取新的数据,再合并到原有的数据模型中。dataProvider支持多种获取数据的方法,不同获取数据方式会通过mode属性来进行区分,不同类型的获取数据方式最终都会把获取到的数据合并到原有的数据模型中。因此,在本发明的具体实施例中,

电子设备可以预先设置当前组件对应的配置信息,然后在当前组件的管理过程中就可以直接获取当前组件对应的配置信息。

[0058] S202、若当前组件对应的配置信息中被嵌入基于Javascript语法的运行沙箱,则通过基于Javascript语法的运行沙箱计算当前组件对应的目标值。

[0059] 在本发明的具体实施例中,若当前组件对应的配置信息中被嵌入基于Javascript语法的运行沙箱,则电子设备可以通过基于Javascript语法的运行沙箱计算当前组件对应的目标值。具体地,电子设备可以先在当前组件对应的配置信息中获取到基于Javascript语法的运行沙箱;然后通过基于Javascript语法的运行沙箱计算当前组件对应的目标值。

[0060] S203、在数据仓库中将当前组件对应的目标值传递给当前组件。

[0061] 在本发明的具体实施例中,电子设备可以在数据仓库中将当前组件对应的目标值传递给当前组件。具体地,电子设备可以先在数据仓库中确定出当前组件,然后在数据仓库中将当前组件对应的目标值传递给当前组件。

[0062] 本发明实施例提出的组件管理方法,先获取当前组件对应的配置信息;然后根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。也就是说,在本发明的技术方案中,可以根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。而在现有的组件管理方法中,前端工程师都需要手动编写代码来实现组件管理。因此,和现有技术相比,本发明实施例提出的组件管理方法,不仅可以减少组件管理成本,还可以提高组件管理效率;并且,本发明实施例的技术方案实现简单方便、便于普及,适用范围更广。

[0063] 实施例三

[0064] 图3为本发明实施例三提供的组件管理方法的流程示意图。如图3所示,组件管理方法可以包括以下步骤:

[0065] S301、获取当前组件对应的配置信息。

[0066] 在本发明的具体实施例中,电子设备可以获取当前组件对应的配置信息。在RCRE中,组件又分为两种类型:无状态组件和有状态组件。在RCRE中,除了container组件之外的其他组件都是无状态组件。开发者只需要把container组件和其他组件分开来看待即可。无状态组件和有状态组件的唯一区别就在于无状态组件不会持有私有的数据模型,它所需要的任何数据,都会通过一种特殊的方式,从父级组件中的有状态组件上获取。不管是无状态组件还是有状态组件,任何一个组件都必然会包含type这个属性。type属性代表着组件的类型。比如type为container的组件就是container组件,type为text的组件就是text文本组件。有状态组件不仅能够持有自己的私有数据模型,也可以通过dataProvider配置,来指定通过什么样的方式来获取新的数据,再合并到原有的数据模型中。dataProvider支持多种获取数据的方法,不同获取数据方式会通过mode属性来进行区分,不同类型的获取数据方式最终都会把获取到的数据合并到原有的数据模型中。因此,在本发明的具体实施例中,电子设备可以预先设置当前组件对应的配置信息,然后在当前组件的管理过程中就可以直接获取当前组件对应的配置信息。

[0067] S302、若当前组件对应的配置信息中存在name属性的组件配置,则根据name属性的组件配置在数据仓库中创建当前组件所依赖的数据。

[0068] 在本发明的具体实施例中,若当前组件对应的配置信息中存在name属性的组件配

置,则电子设备可以根据name属性的组件配置在数据仓库中创建当前组件所依赖的数据。具体地,电子设备可以先在当前组件对应的配置信息中获取到name属性的组件配置;然后根据name属性的组件配置在数据仓库中创建当前组件所依赖的数据。

[0069] S303、若当前组件对应的关联组件产生新的数据,则根据name属性的组件配置将新的数据写入到数据仓库中。

[0070] 在本发明的具体实施例中,若当前组件对应的关联组件产生新的数据,则电子设备可以根据name属性的组件配置将新的数据写入到数据仓库中。具体地,电子设备可以先通过容器组件的响应式接口向数据仓库发送HTTP请求;然后通过容器组件的响应式接口接收数据仓库返回的HTTP响应;再根据数据仓库返回的HTTP响应将新的数据写入到数据仓库中。

[0071] S304、若预先定义的点击操作或者键盘操作被用户触发,则在当前组件对应的配置信息中添加与其对应的事件绑定。

[0072] 在本发明的具体实施例中,若预先定义的点击操作或者键盘操作被用户触发,则电子设备在当前组件对应的配置信息中添加与其对应的事件绑定。具体地,若产品定义的软件操作无法使用单个子任务进行操作,则电子设备将产品定义的软件操作划分为多个软件子操作;各个软件子操作由各个子任务按照预先设置的执行顺序进行操作,并在当前组件对应的配置信息中添加各个子任务对应的事件绑定。

[0073] 图4为本发明实施例三提供的组件生成方法的流程示意图。如图4所示,组件生成方法可以包括以下步骤:S401、创建顶级组件;S402、读取子组件的配置;S403、是否是容器组建;若是,执行S404;否则,执行S412;S404、在依据容器组件的model属性,在数据仓库中分配存储;S405、初始化data属性上的默认值;S406、初始化子组件的默认值;S407、初始化响应式接口;S408、初始化无状态组件控制函数;S409、创建容器组件;S410、是否含有子组件;若是,执行S411;否则,执行S402;S411、组件生成完毕,开始监听用户操作。

[0074] 组件的管理是用户编写了包含整个页面所有组件的关系和逻辑的配置之后,将完成的配置导入引擎进行组件渲染。在渲染初期,引擎会创建一个顶级组件用于挂载子级组件。然后依次递归解析配置来渲染组件。如果组件是一个容器组件,则会先根据容器组件配置上的model属性在数据仓库中预先分配一个存储空间,用于存储当前容器组件以及容器组件下所有无状态组件的数据。容器组件在初始化的时候,还会初始化填写在组件配置上的初始化数据以及控制子组件状态的相关函数和方法,以及初始化响应式接口。如果组件是一个无状态组件,那么直接创建组件即可,然后将组件挂载到父级组件下面。组件的渲染是一个递归的过程,因为组件和组件之间可以嵌套,嵌套的深度由用户编写的组件配置来决定。

[0075] 图5为本发明实施例三提供的容器组件通过ExpressionString动态管理组件的方法流程示意图。如图5所示,容器组件通过ExpressionString动态管理组件的方法可以包括以下步骤:S501、数据仓库数据出现变更;S502、触发容器组件更新;S503、读取数据仓库数据,获取最新的数据上下文;S504、开始更新无状态组件;S505、解析无状态组件的配置中的属性值;S506、判断是否含有ExpressionString;若是,执行S507;否则,执行S510;S507、导入数据上下文,执行ExpressionString;S508、获取执行结果;S509、将ExpressionString替换成执行结果;S510、无状态组件获取处理后的配置,实现更新。

[0076] ExpressionString是一个基于JavaScript语法的运行沙箱,用户在这个沙箱内部的编写的代码,可以被动态计算出结果。假如这个时候,数据仓库中的值发生的变更,会触发容器组件发生更新,容器组件的更新进而会带动无状态组件进行更新。每次容器组件更新时,都会将当前状态下最新的数据组装成上下文对象传递到无状态组件。无状态组件每次更新的时,会对用户编写的ExpressionString进行解析,并以此执行ExpressionString中的JavaScript代码。执行javascript代码之前,会将上下文对象导入到代码作用域内,供ExpressionString中的javascript代码调用。ExpressionString执行完毕后,会将原有的ExpressionString替换成执行结果,无状态组件拿到被解析之后的配置,重新进行更新和渲染。

[0077] 图6为本发明实施例三提供的含有name属性的无状态组件更新方法的流程示意图。如图6所示,含有name属性的无状态组件更新方法可以包括以下步骤:S601、数据仓库数据出现变更;S602、触发容器组件更新;S603、触发无状态组件更新;S604、无状态组件读取配置中的name属性;S605、以name作为参数,调用容器组件提供的函数,获取当前组件的值;S606、无状态组件渲染。

[0078] 容器组件向无状态组件传递数据除了ExpressionString还有通过无状态组件配置上的name属性来完成。同样,当数据容器发生更新时,容器组件开始更新,容器组件的更新会带动内部所有的无状态组件更新。无状态组件的更新的时候,如果组件的配置中存在name属性,组件会调用容器组件向下传递的函数,传入name的值作为参数,即可得到这个在数据仓库中key为name属性值的值。无状态拿到值之后就会立即进行更新并渲染。

[0079] 图7为本发明实施例三提供的组件事件触发的方法流程示意图。如图7所示,组建事件触发的方法可以包括以下步骤:S701、用户点击按钮,触发点击事件;S702、触发无状态组件的事件回调;S703、解析事件绑定上的ExpressionString,获取事件触发的参数;S704、消息派发到任务组控制器;S705、判断事件绑定的任务是普通任务还是任务组;若是任务组,执行S706;若是普通任务,执行S708;S706、执行第1个任务;S707、执行第N个任务;其中,N为大于1的自然数;S708、执行任务。

[0080] 当用户进行页面操作,比如点击某个按钮。按钮的点击事件回调会先被触发,进而触发显示为按钮的这个无状态组件的点击事件回调函数。无状态组件的点击事件回调会读取配置中trigger属性的值,来找到配置中制定需要被执行的任务和传递过去的参数。参数中支持嵌入ExpressionString来实现在点击的时候,动态读取容器组件中的值来计算出一些自定义的值。解析完成之后,无状态组件会依据配置中targetCustomer的定义,将params的值发送到容器组件中name值与其相等的任务。如果name的值是一个任务组的配置,那么会依据steps属性的定义,按照数组的顺序依次执行任务。

[0081] 本发明实施例提出的组件管理方法,先获取当前组件对应的配置信息;然后根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。也就是说,在本发明的技术方案中,可以根据当前组件对应的配置信息对数据仓库中当前组件所依赖的数据进行管理。而在现有的组件管理方法中,前端工程师都需要手动编写代码来实现组件管理。因此,和现有技术相比,本发明实施例提出的组件管理方法,不仅可以减少组件管理成本,还可以提高组件管理效率;并且,本发明实施例的技术方案实现简单方便、便于普及,适用范围更广。

[0082] 实施例四

[0083] 图8为本发明实施例四提供的组件管理装置的结构示意图。如图8所示,本发明实施例所述的组件管理装置可以包括:获取模块801和管理模块802;其中,

[0084] 所述获取模块801,用于获取当前组件对应的配置信息;

[0085] 所述管理模块802,用于根据所述当前组件对应的配置信息对数据仓库中所述当前组件所依赖的数据进行管理。

[0086] 进一步的,所述管理模块802,具体用于若所述当前组件对应的配置信息中被嵌入基于Javascript语法的运行沙箱,则通过所述基于Javascript语法的运行沙箱计算所述当前组件对应的目标值;在所述数据仓库中将所述当前组件对应的目标值传递给所述当前组件。

[0087] 进一步的,所述管理模块802,具体用于若所述当前组件对应的配置信息中存在name属性的组件配置,则根据所述name属性的组件配置在所述数据仓库中创建所述当前组件所依赖的数据。

[0088] 进一步的,所述管理模块802,还用于若所述当前组件对应的关联组件产生新的数据,则根据所述name属性的组件配置将所述新的数据写入到所述数据仓库中;若预先定义的点击操作或者键盘操作被用户触发,则在所述当前组件对应的配置信息中添加与其对应的事件绑定。

[0089] 进一步的,所述管理模块802,具体用于通过所述容器组件的响应式接口向所述数据仓库发送HTTP请求;通过所述容器组件的响应式接口接收所述数据仓库返回的HTTP响应;根据所述数据仓库返回的HTTP响应将所述新的数据写入到所述数据仓库中。

[0090] 进一步的,所述管理模块802,具体用于若产品定义的软件操作无法使用单个子任务进行操作,则将所述产品定义的软件操作划分为多个软件子操作;各个软件子操作由各个子任务按照预先设置的执行顺序进行操作,并在所述当前组件对应的配置信息中添加各个子任务对应的事件绑定。

[0091] 上述组件管理装置可执行本发明任意实施例所提供的方法,具备执行方法相应的功能模块和有益效果。未在本实施例中详尽描述的技术细节,可参见本发明任意实施例提供的组件管理方法。

[0092] 实施例五

[0093] 图9为本发明实施例五提供的电子设备的结构示意图。图9示出了适于用来实现本发明实施方式的示例性电子设备的框图。图9显示的电子设备12仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0094] 如图9所示,电子设备12以通用计算设备的形式表现。电子设备12的组件可以包括但不限于:一个或者多个处理器或者处理单元16,系统存储器28,连接不同系统组件(包括系统存储器28和处理单元16)的总线18。

[0095] 总线18表示几类总线结构中的一种或多种,包括存储器总线或者存储器控制器,外围总线,图形加速端口,处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说,这些体系结构包括但不限于工业标准体系结构(ISA)总线,微通道体系结构(MAC)总线,增强型ISA总线、视频电子标准协会(VESA)局域总线以及外围组件互连(PCI)总线。

[0096] 电子设备12典型地包括多种计算机系统可读介质。这些介质可以是任何能够被电

子设备12访问的可用介质,包括易失性和非易失性介质,可移动的和不可移动的介质。

[0097] 系统存储器28可以包括易失性存储器形式的计算机系统可读介质,例如随机存取存储器(RAM)30和/或高速缓存存储器32。电子设备12可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例,存储系统34可以用于读写不可移动的、非易失性磁介质(图9未显示,通常称为“硬盘驱动器”)。尽管图9中未示出,可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器,以及对可移动非易失性光盘(例如CD-ROM,DVD-ROM或其它光介质)读写的光盘驱动器。在这些情况下,每个驱动器可以通过一个或者多个数据介质接口与总线18相连。存储器28可以包括至少一个程序产品,该程序产品具有一组(例如至少一个)程序模块,这些程序模块被配置以执行本发明各实施例的功能。

[0098] 具有一组(至少一个)程序模块42的程序/实用工具40,可以存储在例如存储器28中,这样的程序模块42包括但不限于操作系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块42通常执行本发明所描述的实施例中的功能和/或方法。

[0099] 电子设备12也可以与一个或多个外部设备14(例如键盘、指向设备、显示器24等)通信,还可与一个或者多个使得用户能与该电子设备12交互的设备通信,和/或与使得该电子设备12能与一个或多个其它计算设备进行通信的任何设备(例如网卡,调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口22进行。并且,电子设备12还可以通过网络适配器20与一个或者多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,例如因特网)通信。如图所示,网络适配器20通过总线18与电子设备12的其它模块通信。应当明白,尽管图中未示出,可以结合电子设备12使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0100] 处理单元16通过运行存储在系统存储器28中的程序,从而执行各种功能应用以及数据处理,例如实现本发明实施例所提供的组件管理方法。

[0101] 实施例六

[0102] 本发明实施例六提供了一种计算机存储介质。

[0103] 本发明实施例的计算机可读存储介质,可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0104] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可

读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0105] 计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括——但不限于无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0106] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码,所述程序设计语言包括面向对象的程序设计语言——诸如Java、Smalltalk、C++,还包括常规的过程式程序设计语言——诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)——连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0107] 注意,上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解,本发明不限于这里所述的特定实施例,对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此,虽然通过以上实施例对本发明进行了较为详细的说明,但是本发明不仅仅限于以上实施例,在不脱离本发明构思的情况下,还可以包括更多其他等效实施例,而本发明的范围由所附的权利要求范围决定。

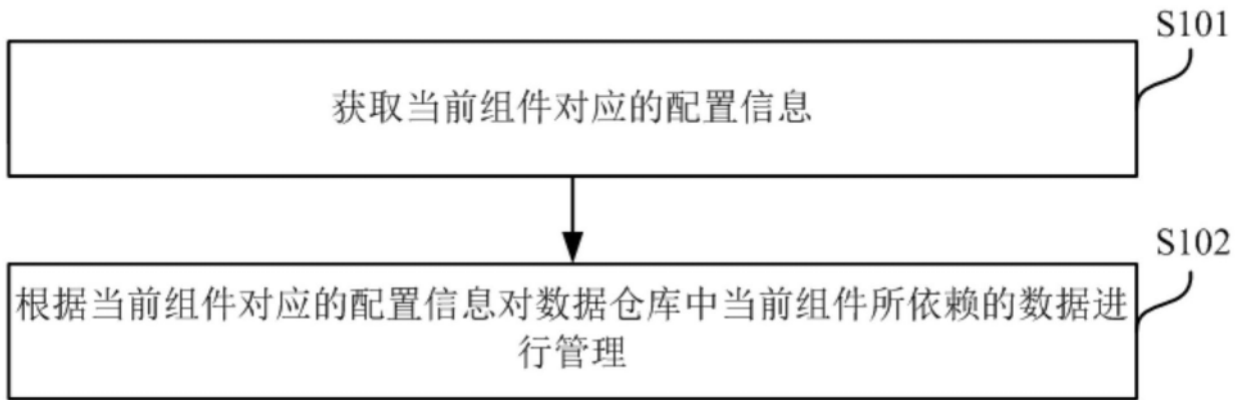


图1

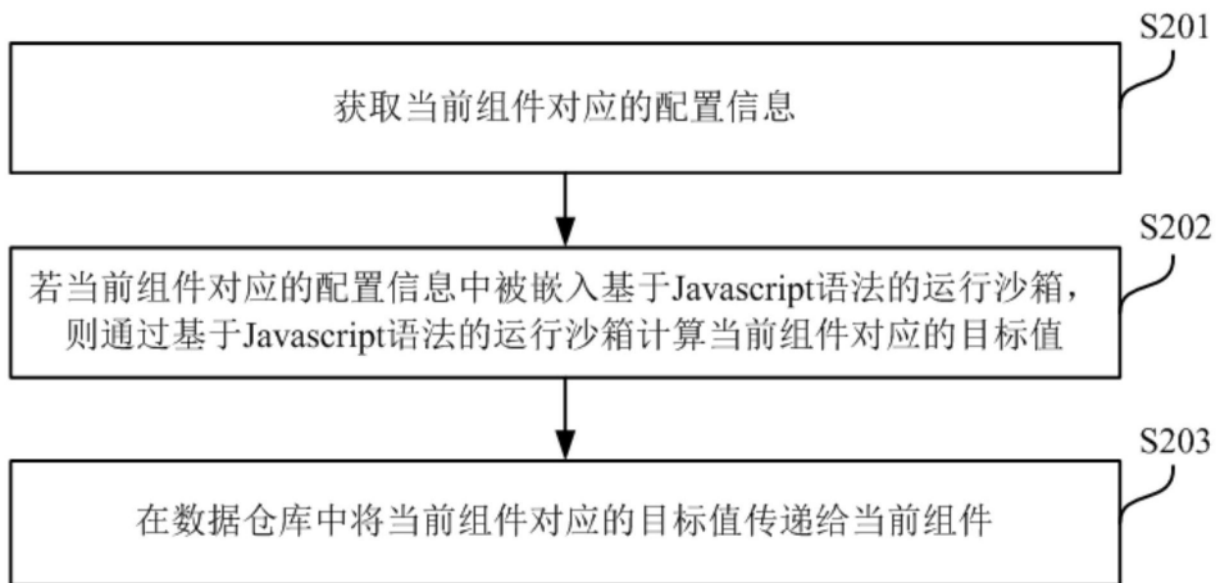


图2

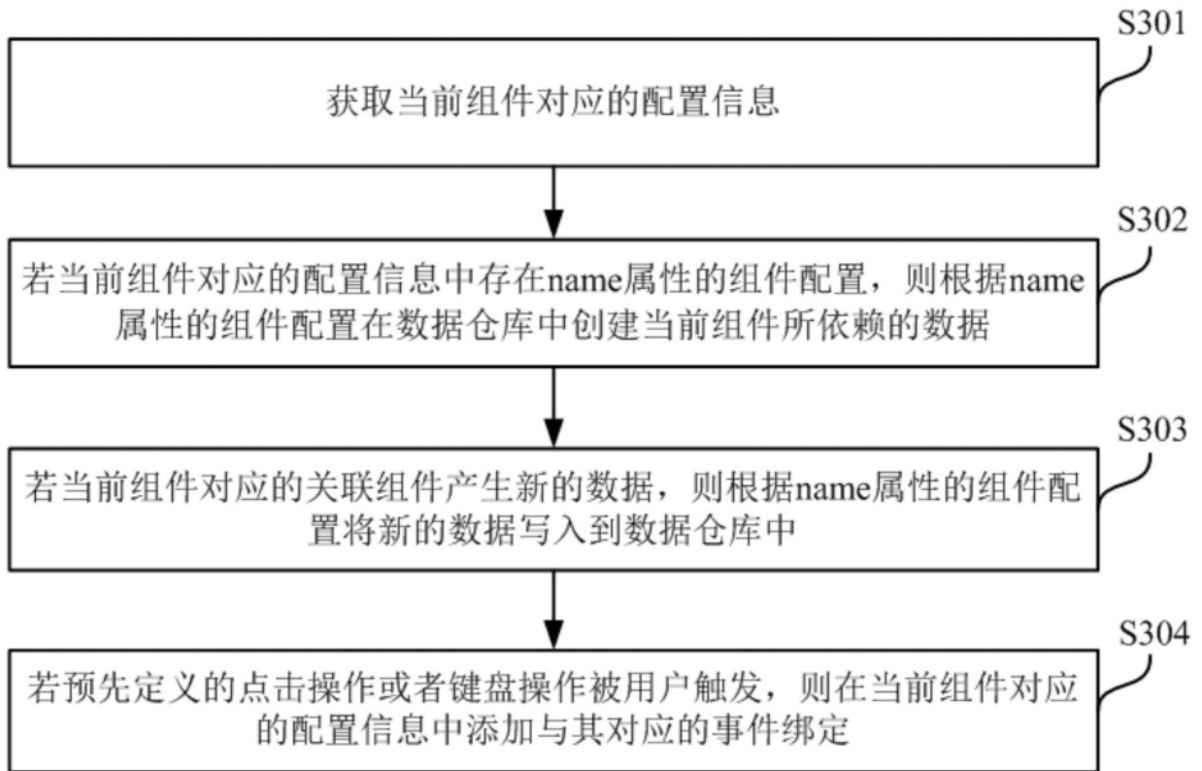


图3

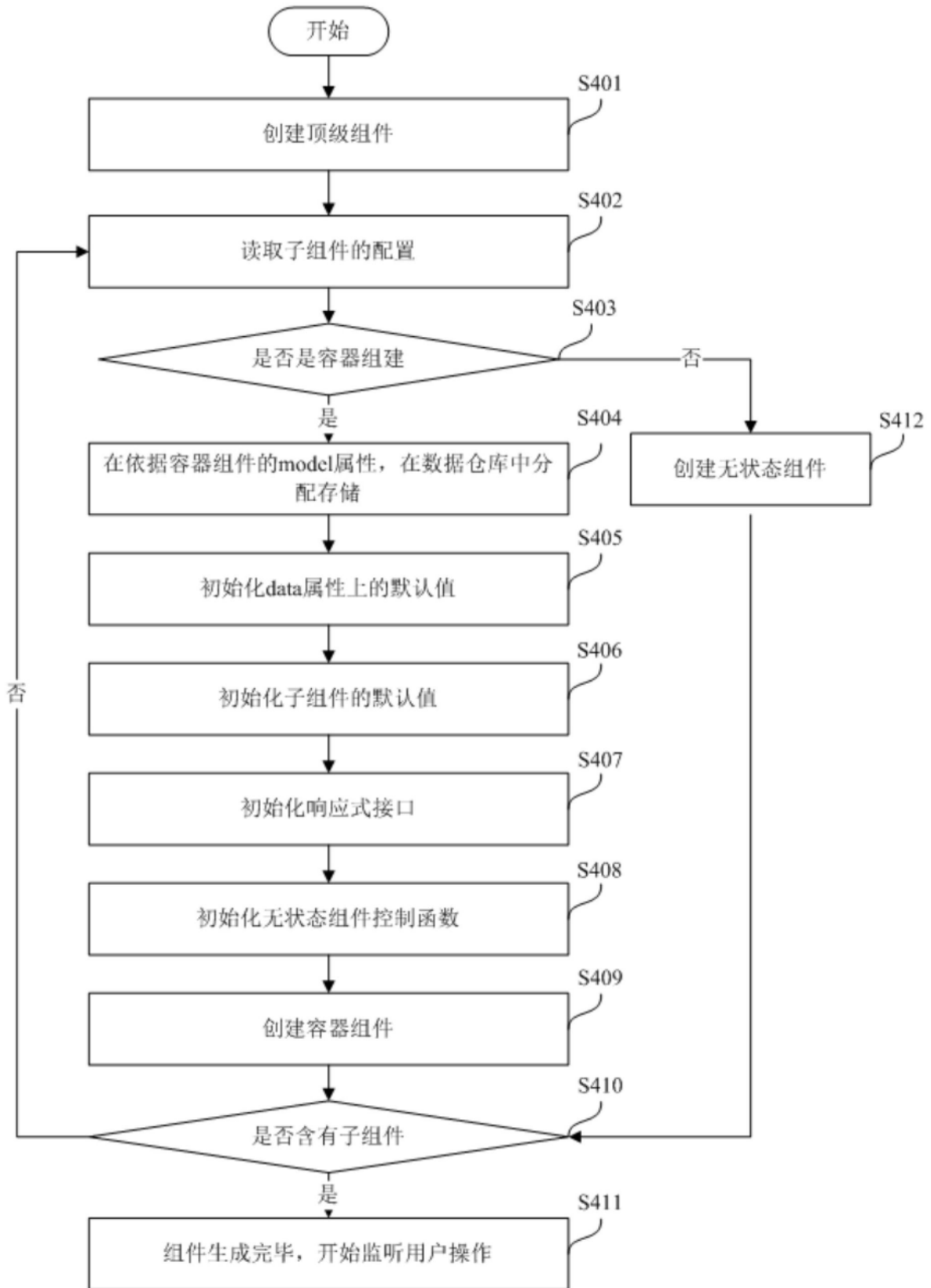


图4

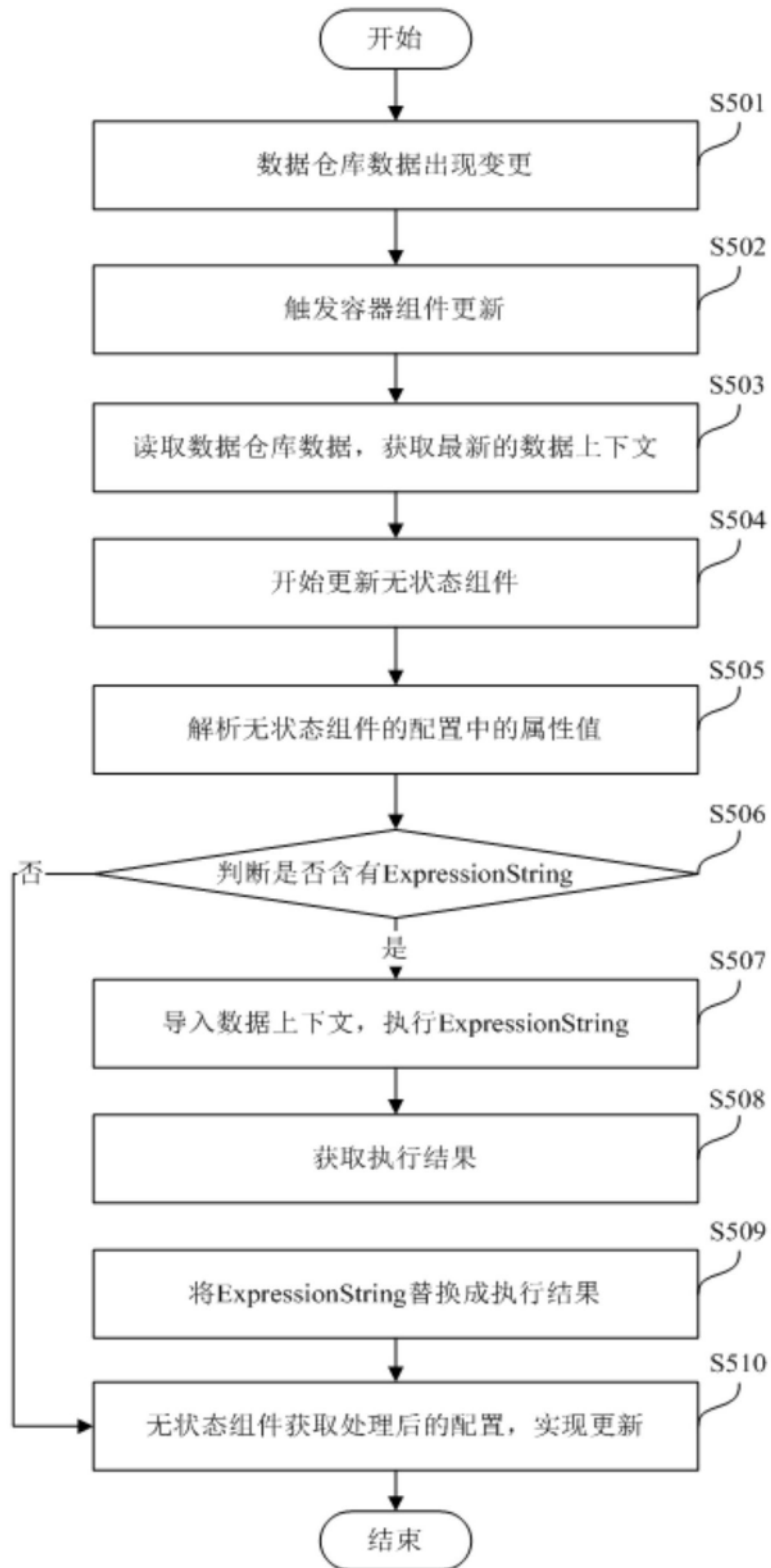


图5

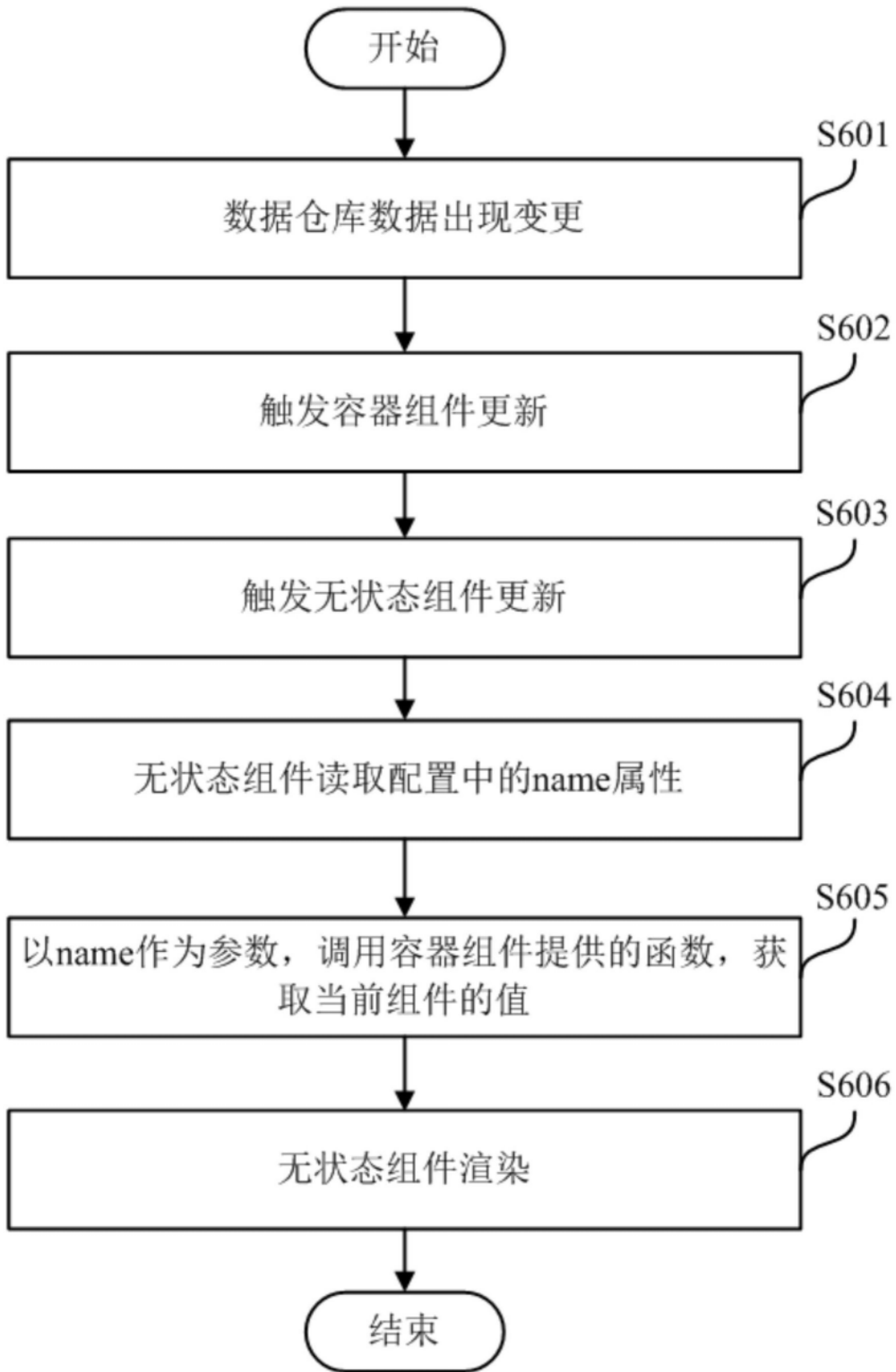


图6

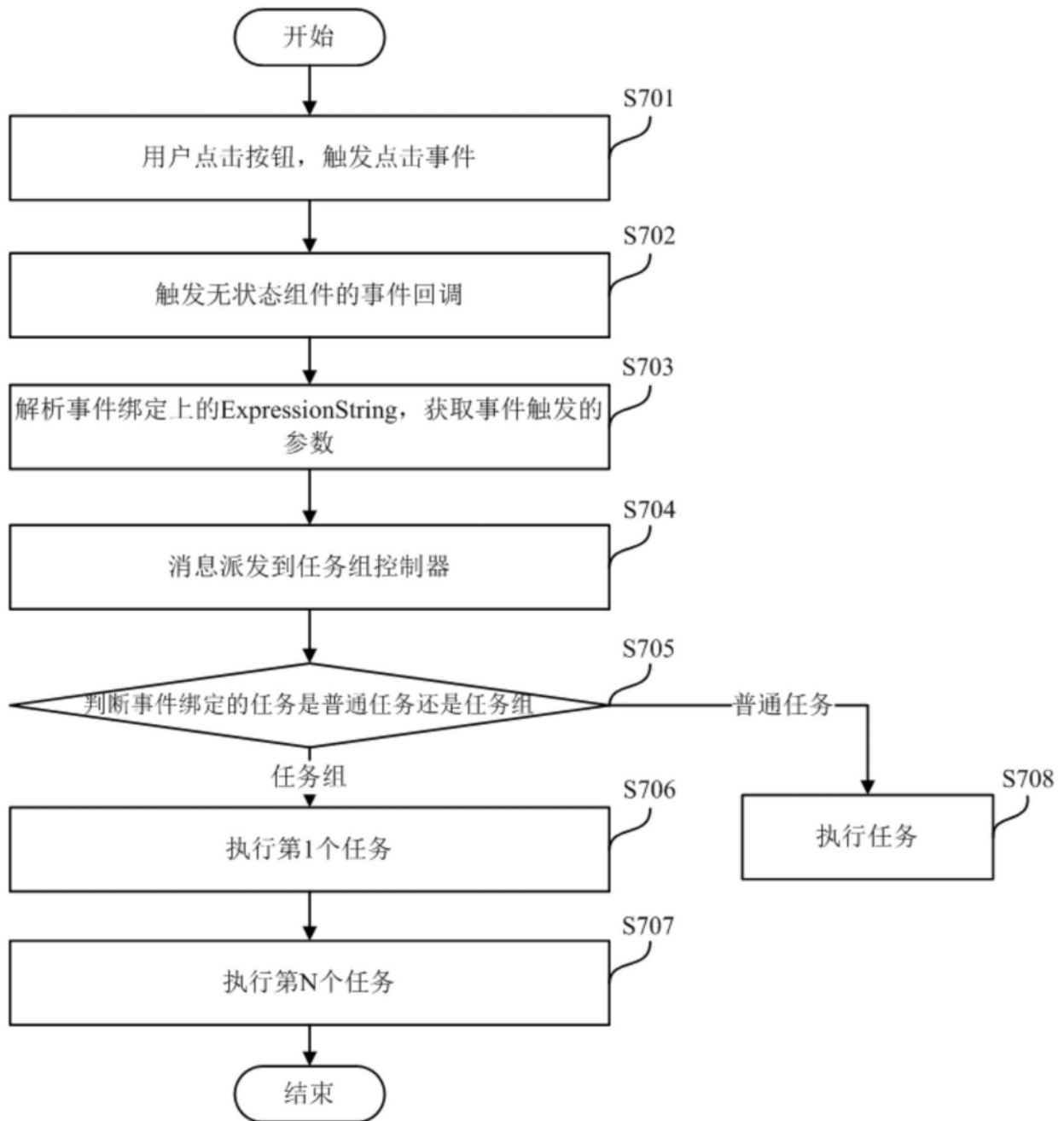


图7

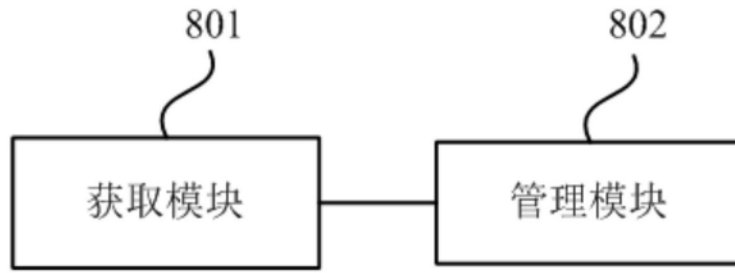


图8

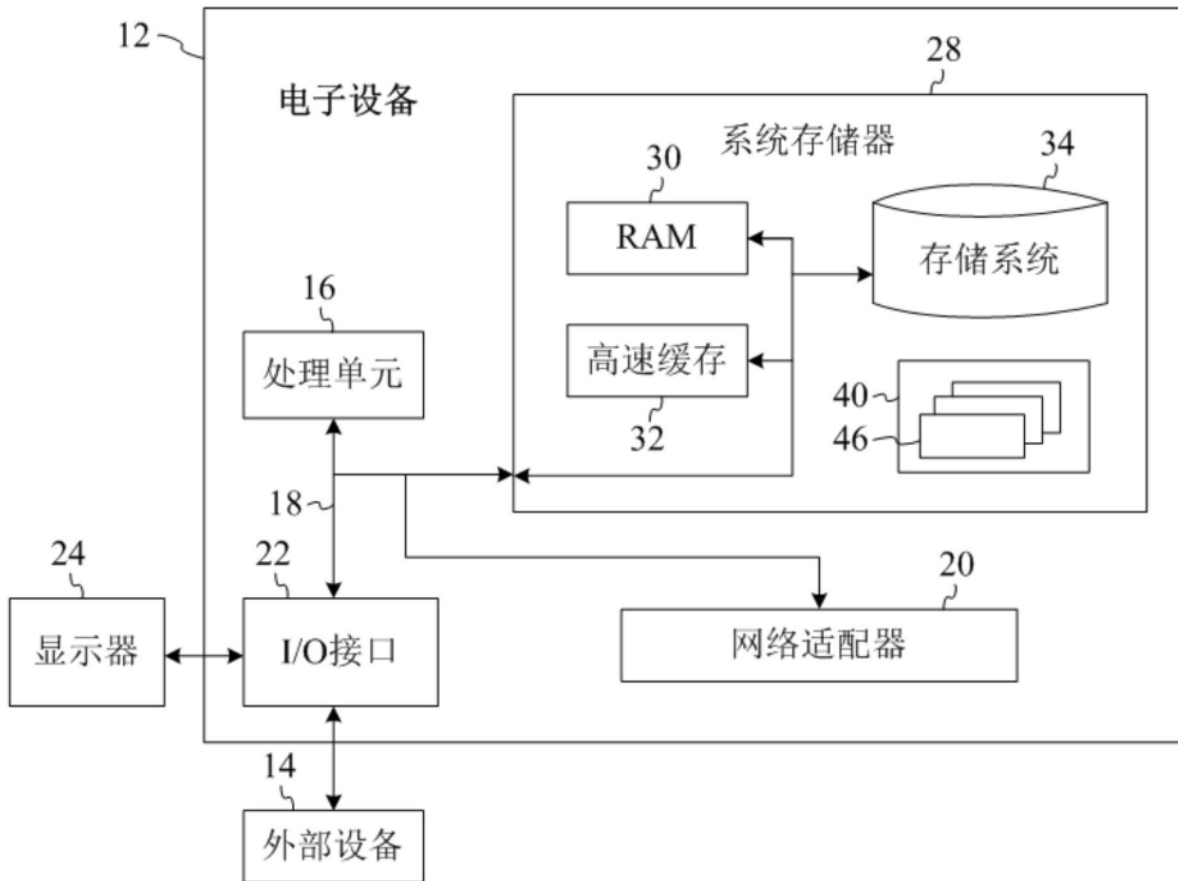


图9