



(12) 发明专利

(10) 授权公告号 CN 111651158 B

(45) 授权公告日 2024.05.31

(21) 申请号 202010328627.5

(22) 申请日 2020.04.23

(65) 同一申请的已公布的文献号
申请公布号 CN 111651158 A

(43) 申请公布日 2020.09.11

(73) 专利权人 平安国际智慧城市科技股份有限公司

地址 518000 广东省深圳市前海深港合作区妈湾兴海大道3048号前海自贸大厦1-34层

(72) 发明人 洪笃辉

(74) 专利代理机构 深圳市赛恩倍吉知识产权代理有限公司 44334
专利代理师 林丽纯

(51) Int.Cl.

G06F 8/38 (2018.01)

G06F 16/958 (2019.01)

(56) 对比文件

CN 104267949 A, 2015.01.07

CN 106502973 A, 2017.03.15

CN 106598924 A, 2017.04.26

CN 109634598 A, 2019.04.16

CN 109766097 A, 2019.05.17

US 2007198940 A1, 2007.08.23

US 2011321008 A1, 2011.12.29

US 2016057220 A1, 2016.02.25

审查员 龙绘蓝

权利要求书2页 说明书14页 附图2页

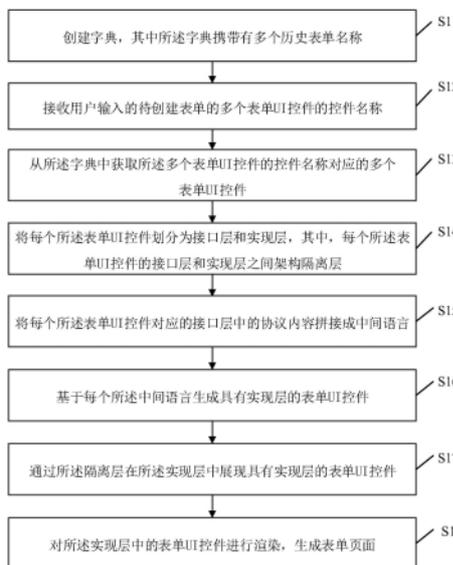
(54) 发明名称

表单页面生成方法、装置、终端及存储介质

(57) 摘要

本发明提供一种表单页面生成方法,包括:创建字典;接收用户输入的待创建表单的多个表单UI控件的控件名称;从字典中获取控件名称对应的多个表单UI控件;将每个表单UI控件划分为接口层和实现层,并在接口层和实现层之间架构隔离层;将每个表单UI控件对应的接口层中的协议内容拼接成中间语言;基于每个中间语言生成具有实现层的表单UI控件;通过隔离层在实现层中展现具有实现层的表单UI控件;对实现层中的表单UI控件进行渲染,生成表单页面。本发明还提供一种表单页面生成装置、终端及存储介质。本发明通过把表单UI控件的技术实现分离出接口层与实现层,从而分离了控件的配置和渲染,提高浏览器的渲染效率,满足了各种场景的定制需求。

CN 111651158 B



1. 一种表单页面生成方法,其特征在于,所述表单页面生成方法包括:
 - 创建字典,其中,所述字典携带有多个历史表单名称;
 - 接收用户输入的待创建表单的多个表单UI控件的控件名称;
 - 从所述字典中获取所述多个表单UI控件的控件名称对应的多个表单UI控件;
 - 将每个所述表单UI控件划分为接口层和实现层,其中,每个所述表单UI控件的接口层和实现层之间架构隔离层;
 - 将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言;
 - 基于每个所述中间语言生成具有实现层的表单UI控件,包括:将每个所述中间语言输入到翻译器中;调用所述翻译器从配置中心请求所述中间语言对应的字典,并将所述字典设置为激活状态;调用所述配置中心从注册表中读取处于激活状态的字典对应的表单UI控件;返回所述表单UI控件至所述翻译器;调用所述翻译器根据所述表单UI控件从每个所述中间语言中读取对应的表单UI控件的控件信息,其中,所述表单UI控件的控件信息对应有关键名称;基于所述表单UI控件的控件名称将从每个所述中间语言读取到的第一表单UI控件与所述配置中心从所述注册表中读取到的第二表单UI控件进行匹配,其中,所述从每个所述中间语言读取到的所述第一表单UI控件对应有关键名称,所述配置中心从所述注册表中读取到的所述第二表单UI控件对应有关键名称;关联相匹配的所述第一表单UI控件的接口层和所述第二表单UI控件的实现层,得到一个对应有关键名称和实现层的表单UI控件列表;调用所述翻译器读取所述表单UI控件列表中每个所述第二表单UI控件生成具有实现层的表单UI控件;
 - 通过所述隔离层在所述实现层中展现具有实现层的表单UI控件;
 - 对所述实现层中的表单UI控件进行渲染,生成表单页面。
2. 如权利要求1所述的表单页面生成方法,其特征在于,所述将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言包括:
 - 接收表单画布中用户拖曳的表单UI控件;
 - 根据所述用户拖曳的顺序读取所述表单画布中的表单UI控件;
 - 每读取到一个表单UI控件,将所述表单UI控件对应的接口层中的协议内容拼接成中间语言。
3. 如权利要求1所述的表单页面生成方法,其特征在于,所述通过所述隔离层在所述实现层中展现具有实现层的表单UI控件包括:
 - 获取所述具有实现层的表单UI控件对应的多个渲染版本;
 - 注册所述多个渲染版本至注册表中,并将所述多个渲染版本的状态修改为挂起状态;
 - 确定所述注册表中的一个渲染版本为目标渲染版本,并将所述目标渲染版本配置到所述配置中心;
 - 修改所述配置中心中的所述目标渲染版本的状态为激活状态;
 - 所述翻译器从所述配置中心中获取所述状态为激活状态对应的渲染版本的实现层信息;
 - 在所述实现层中展现所述实现层信息对应的表单UI控件的实现层。
4. 如权利要求1所述的表单页面生成方法,其特征在于,所述对所述实现层中的表单UI控件进行渲染,生成表单页面包括:

对所述实现层中的表单UI控件进行渲染得到表单UI控件的渲染片；
基于所述渲染片共同构成整张表单页面的渲染图；
基于所述渲染图生成表单页面。

5. 如权利要求1所述的表单页面生成方法,其特征在于,所述创建字典包括:

获取所述多个历史表单名称及每个历史表单名称对应的表单UI控件协议层,每个所述表单UI控件协议层是由所述历史表单名称对应的多个表单UI控件的接口层组成的,每个所述表单UI控件的接口层携带有对应表单UI控件的控件名称;

将所述多个历史表单名称作为字典的第一层级,每个所述历史表单名称对应的多个表单UI控件的接口层作为字典的第二层级,多个表单UI控件的接口层对应的协议层作为字典的第三层级。

6. 如权利要求1所述的表单页面生成方法,其特征在于,所述将每个所述表单UI控件划分为接口层和实现层包括:

将每个所述表单UI控件创建为浏览器能够原生识别的自定义标签;

调用CustomElements.define传入所述自定义标签将每个所述表单UI控件划分为接口层和实现层。

7. 一种表单页面生成装置,其特征在于,所述装置用于实现如权利要求1至6中任意一项所述的表单页面生成方法,所述表单页面生成装置包括:

创建模块,用于创建字典,其中所述字典携带有多个历史表单名称;

接收模块,用于接收用户输入的待创建表单的多个表单UI控件的控件名称;

获取模块,用于从所述字典中获取所述多个表单UI控件的控件名称对应的多个表单UI控件;

划分模块,用于将每个所述表单UI控件划分为接口层和实现层,其中,每个所述表单UI控件的接口层和实现层之间架构隔离层;

拼接模块,用于将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言;

生成模块,用于基于每个所述中间语言生成具有实现层的表单UI控件;

展现模块,用于通过所述隔离层在所述实现层中展现具有实现层的表单UI控件;

渲染模块,用于对所述实现层中的表单UI控件进行渲染,生成表单页面。

8. 一种终端,其特征在于,所述终端包括处理器,所述处理器用于执行存储器中存储的计算机程序时实现如权利要求1至6中任意一项所述的表单页面生成方法。

9. 一种计算机可读存储介质,所述计算机可读存储介质上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至6中任意一项所述的表单页面生成方法。

表单页面生成方法、装置、终端及存储介质

技术领域

[0001] 本发明涉及计算机技术领域,具体涉及一种表单页面生成方法、装置、终端及存储介质。

背景技术

[0002] 表单是用户向系统提交信息的交互方式之一,它几乎承担了网站的信息录入工作。随着信息化建设的日益发展,数据的多样性愈加明显,也使得表单UI开发工作量日渐增多,因而基于可视化操作界面的表单设计器不仅简化了表单UI开发的开发工作,而且降低了表单开发的门槛,使得非开发领域的业务人员也能依照业务需求自行设计可交互的表单UI并发布到系统上,从而降低了开发成本,提高了表单的交付效率。

[0003] 然而,市面上的表单设计器仅提供了单一的技术栈,使得生成的表单页面只能运行于特定技术栈的系统,往往生成的表单页面也只能发布到指定的第三方系统。这不仅限制了表单的二次开发以满足更多定制场景,也加大了与现有系统的技术集成难度;另一方面,把表单托管在第三方系统是不利于系统的信息安全。因此,这类表单设计器生成的表单页面只能解决简单的表单需求,承担简单的信息录入工作,不利于对其定制开发和动态渲染。

发明内容

[0004] 鉴于以上内容,有必要提出一种表单页面生成方法、装置、终端及存储介质,通过把表单UI控件的技术实现分离出接口层与实现层,从而分离了控件的配置和渲染,提高浏览器的渲染效率,满足了各种场景的定制需求。

[0005] 本发明的第一方面提供一种表单页面生成方法,所述方法包括:

[0006] 创建字典,其中所述字典携带有多个历史表单名称;

[0007] 接收用户输入的待创建表单的多个表单UI控件的控件名称;

[0008] 从所述字典中获取所述多个表单UI控件的控件名称对应的多个表单UI控件;

[0009] 将每个所述表单UI控件划分为接口层和实现层,其中,每个所述表单UI控件的接口层和实现层之间架构隔离层;

[0010] 将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言;

[0011] 基于每个所述中间语言生成具有实现层的表单UI控件;

[0012] 通过所述隔离层在所述实现层中展现具有实现层的表单UI控件;

[0013] 对所述实现层中的表单UI控件进行渲染,生成表单页面。

[0014] 优选的,所述基于每个所述中间语言生成具有实现层的表单UI控件包括:

[0015] 将每个所述中间语言输入到翻译器中;

[0016] 调用所述翻译器从配置中心请求所述中间语言对应的字典并将所述字典设置为激活状态;

[0017] 调用所述配置中心从注册表中读取处于激活状态的字典对应的表单UI控件;

- [0018] 返回所述表单UI控件至所述翻译器；
- [0019] 调用所述翻译器根据所述表单UI控件从每个所述中间语言中读取对应的表单UI控件的控件信息,其中,所述表单UI控件的控件信息对应有控件名称；
- [0020] 基于所述表单UI控件的控件名称将从每个所述中间语言读取到的第一表单UI控件与所述配置中心从所述注册表中读取到的第二表单UI控件进行匹配,其中,所述从每个所述中间语言读取到的所述第一表单UI控件对应有接口层,所述配置中心从所述注册表中读取到的所述第二表单UI控件对应有实现层；
- [0021] 关联相匹配的所述第一表单UI控件的接口层和所述第二表单UI控件的实现层,得到一个对应有接口层和实现层的表单UI控件列表；
- [0022] 调用所述翻译器读取所述表单UI控件列表中每个所述第二表单UI控件生成具有实现层的表单UI控件。
- [0023] 优选的,所述将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言包括：
- [0024] 接收表单画布中用户拖曳的表单UI控件；
- [0025] 根据所述用户拖曳的顺序读取所述表单画布中的表单UI控件；
- [0026] 每读取到一个表单UI控件,将所述表单UI控件对应的接口层中的协议内容拼接成中间语言。
- [0027] 优选的,所述通过所述隔离层在所述实现层中展现具有实现层的表单UI控件包括：
- [0028] 获取所述具有实现层的表单UI控件对应的多个渲染版本；
- [0029] 注册所述多个渲染版本至注册表中,并将所述多个渲染版本的状态修改为挂起状态；
- [0030] 确定所述注册表中的一个渲染版本为目标渲染版本,并将所述目标渲染版本配置到所述配置中心；
- [0031] 修改所述配置中心中的所述目标渲染版本的状态为激活状态；
- [0032] 所述翻译器从所述配置中心中获取所述状态为激活状态对应的渲染版本的实现层信息；
- [0033] 在所述实现层中展现所述实现层信息对应的表单UI控件的实现层。
- [0034] 优选的,所述对所述实现层中的表单UI控件进行渲染,生成表单页面包括：
- [0035] 对所述实现层中的表单UI控件进行渲染得到表单UI控件的渲染片；
- [0036] 基于所述渲染片共同构成整张表单页面的渲染图；
- [0037] 基于所述渲染图生成表单页面。
- [0038] 优选的,所述创建字典包括：
- [0039] 获取所述多个历史表单名称及每个历史表单名称对应的表单UI控件协议层,每个所述表单UI控件协议层是由所述历史表单名称对应的多个表单UI控件的接口层组成的,每个所述表单UI控件的接口层携带有对应表单UI控件的控件名称；
- [0040] 将所述多个历史表单名称作为字典的第一层级,每个所述历史表单名称对应的多个表单UI控件的接口层作为字典的第二层级,多个表单UI控件的接口层对应的协议层作为字典的第三层级。

- [0041] 优选的,所述将每个所述表单UI控件划分为接口层和实现层包括:
- [0042] 将每个所述表单UI控件创建为浏览器能够原生识别的自定义标签;
- [0043] 调用CustomElements.define传入所述自定义标签将每个所述表单UI控件划分为接口层和实现层。
- [0044] 本发明的第二方面提供一种表单页面生成装置,所述装置包括:
- [0045] 创建模块,用于创建字典,其中所述字典携带有多个历史表单名称;
- [0046] 接收模块,用于接收用户输入的待创建表单的多个表单UI控件的控件名称;
- [0047] 获取模块,用于从所述字典中获取所述多个表单UI控件的控件名称对应的多个表单UI控件;
- [0048] 划分模块,用于将每个所述表单UI控件划分为接口层和实现层,其中,每个所述表单UI控件的接口层和实现层之间架构隔离层;
- [0049] 拼接模块,用于将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言;
- [0050] 生成模块,用于基于每个所述中间语言生成具有实现层的表单UI控件;
- [0051] 展现模块,用于通过所述隔离层在所述实现层中展现具有实现层的表单UI控件;
- [0052] 渲染模块,用于对所述实现层中的表单UI控件进行渲染,生成表单页面。
- [0053] 本发明的第三方面提供一种终端,所述终端包括处理器,所述处理器用于执行存储器中存储的计算机程序时实现所述表单页面生成方法。
- [0054] 本发明的第四方面提供一种计算机可读存储介质,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现所述表单页面生成方法。
- [0055] 综上所述,本发明所述的表单页面生成方法、装置、终端及存储介质,一方面通过将表单UI控件划分为接口层与实现层,从而分离了表单UI控件的配置和渲染,为定制开发提供了技术基础,又通过架设隔离层,使得每个控件的实现层彼此独立、互不干扰,通过切换实现层实现不同渲染结果的目的,提高浏览器的渲染效率;最后,将所述表单UI控件对应的接口层中的协议内容翻译成中间语言,便于后期运用在不同的技术语言中,同时基于所述中间语言生成具有实现层的表单UI控件,在不改变任意表单UI控件协议的条件对任意表单UI控件进行二次开发,满足了各种场景的定制需求。

附图说明

- [0056] 图1是本发明实施例一提供的表单页面生成方法的流程图。
- [0057] 图2是本发明实施例二提供的表单页面生成装置的结构图。
- [0058] 图3是本发明实施例三提供的终端的结构示意图。
- [0059] 如下具体实施方式将结合上述附图进一步说明本发明。

具体实施方式

[0060] 为了能够更清楚地理解本发明的上述目的、特征和优点,下面结合附图和具体实施例对本发明进行详细描述。需要说明的是,在不冲突的情况下,本发明的实施例及实施例中的特征可以相互组合。

[0061] 除非另有定义,本文所使用的所有的技术和科学术语与属于本发明的技术领域的

技术人员通常理解的含义相同。本文中在本发明的说明书中所使用的术语只是为了描述具体的实施例的目的,不是旨在于限制本发明。

[0062] 实施例一

[0063] 图1是本发明实施例一提供的表单页面生成方法的流程图。

[0064] 在本实施例中,所述表单页面生成方法可以应用于终端中,对于需要进行表单页面生成的终端,可以直接在终端上集成本发明的方法所提供的表单页面生成的功能,或者以软件开发工具包(Software Development Kit,SKD)的形式运行在终端中。

[0065] 如图1所示,所述表单页面生成方法具体包括以下步骤,根据不同的需求,该流程图中步骤的顺序可以改变,某些可以省略。

[0066] S11:创建字典,其中所述字典携带有多个历史表单名称。

[0067] 本实施例中,所述字典是指多个历史表单的表单UI控件对应的一系列协议层的集合,所述字典中存储了每个历史表单名称、每个历史表单的表单UI控件、表单UI控件对应的协议及对应的协议层,其中,所述协议包括每个表单UI控件的控件名称、协议内容、HTMLElement标签名称、属性、方法、接口层、实现层及其参数列表。

[0068] 优选的,所述创建字典包括:

[0069] 获取所述多个历史表单名称及每个历史表单名称对应的表单UI控件协议层,每个所述表单UI控件协议层是由所述历史表单名称对应的多个表单UI控件的接口层组成的,每个所述表单UI控件的接口层携带有对应表单UI控件的控件名称;

[0070] 将所述多个历史表单名称作为字典的第一层级,每个所述历史表单名称对应的多个表单UI控件的接口层作为字典的第二层级,多个表单UI控件的接口层对应的协议层作为字典的第三层级。

[0071] 本实施例中,每个历史表单名称对应一个协议层,每个协议层对应多个表单UI控件的接口层,每个所述表单UI控件的接口层对应一个表单UI控件,每个表单UI控件对应应有表单UI控件的控件名称,将所述历史表单名称、每个历史表单名称对应的表单UI控件及表单UI控件的接口层对应的协议层定义为不同的层级,提高了获取表单UI控件的效率。

[0072] S12:接收用户输入的待创建表单的多个表单UI控件的控件名称。

[0073] 本实施例中,当用户需要制作表单时,在浏览器中输入待创建表单的表单名称及所述表单名称对应的多个表单UI控件的控件名称。

[0074] S13:从所述字典中获取所述多个表单UI控件的控件名称对应的多个表单UI控件。

[0075] 本实施例中,所述字典中存储有每个历史表单的表单名称及每个历史表单的表单名称对应的多个表单UI控件及表单UI控件的控件名称,当接收到用户输入的待创建表单的多个表单UI控件的控件名称时,从所述字典中获取每个所述表单UI控件的控件名称对应的表单UI控件。

[0076] S14:将每个所述表单UI控件划分为接口层和实现层,其中,每个所述表单UI控件的接口层和实现层之间架构隔离层。

[0077] 本实施例中,所述接口层是指定义所述表单UI控件的属性和行为,就是将所述表单UI控件的内部操作和外部环境在数据处理过程中相互独立,例如,修改所述表单UI控件的内部操作的时候不会影响到所述表单UI控件的外部环境;所述实现层是指执行所述表单UI控件的接口层定义的协议内容来实现内部操作,为了防止所述实现层的内部细节污染到

全局环境,在将所述实现层和所述接口层之间架构隔离层。

[0078] 优选的,所述将每个所述表单UI控件划分为接口层和实现层包括:

[0079] 将每个所述表单UI控件创建为浏览器能够原生识别的自定义标签;

[0080] 调用CustomElements.define传入所述自定义标签将每个所述表单UI控件划分为接口层和实现层。

[0081] 本实施例中,通过采用Web Components技术中的Custom Elements技术,把表单UI控件创建为浏览器能够原生识别的HTML Element(自定义)标签,调用customElements.define方法传入所述HTML Element标签以及使用JavaScript Class实现每个表单UI控件对应的接口层得到的实用层,从而把每个表单UI控件分为接口层和实现层,当将所述表单UI控件分为接口层和实现层之后,采用Shadow DOM技术在每个所述表单UI控件的接口层和实现层之间架构隔离层。

[0082] 本实施例中,通过将每个控件分为接口层和实现层,是为了将配置和渲染分开,其中,所述配置通过接口层进行实现,所述渲染通过实现层中进行实现,提高了渲染效率,架构隔离层是由于浏览器提供的默认运行环境没有隔离性,实现层的内部细节很容易污染到全局环境,导致其他的UI控件受到干扰而无法正常工作,而在所述实现层和所述接口层之间架构隔离层提高了渲染质量。

[0083] S15:将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言。

[0084] 本实施例中,所述表单UI控件对应的接口层中的协议内容以代码的形式既不利于保存,又不利于传输,为了方便存储和传输,将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言。

[0085] 优选的,所述将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言包括:

[0086] 接收表单画布中用户拖曳的表单UI控件;

[0087] 根据所述用户拖曳的顺序读取所述表单画布中的表单UI控件;

[0088] 每读取到一个表单UI控件,将所述表单UI控件对应的接口层中的协议内容拼接成中间语言。

[0089] 示例性的,用户将表单UI控件列表“我”“平安”“来自”拖曳到表单画布中,根据用户拖曳的顺序读取每个表单UI控件,读取的第一个表单UI控件对应的接口层的协议内容为“我”,读取的第二个表单UI控件对应的接口层的协议内容为“来自”,读取的第三个表单UI控件对应的接口层的协议内容为“平安”,按照读取的先后顺序将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言“我来自平安”。

[0090] S16:基于每个所述中间语言生成具有实现层的表单UI控件。

[0091] 本实施例中,在生成具有实现层的表单UI控件的过程中,每个协议层对应多个实现层,为了达到定制的目的,设计了翻译器、配置中心和注册表,所述翻译器、配置中心和注册表协同作用在翻译层。

[0092] 优选的,所述基于每个所述中间语言生成具有实现层的表单UI控件包括:

[0093] 将每个所述中间语言输入到翻译器中;

[0094] 调用所述翻译器从配置中心请求所述中间语言对应的字典并将所述字典设置为激活状态;

- [0095] 调用所述配置中心从注册表中读取处于激活状态的字典对应的表单UI控件；
- [0096] 返回所述表单UI控件至所述翻译器；
- [0097] 调用所述翻译器根据所述表单UI控件从每个所述中间语言中读取对应的表单UI控件的控件信息,其中,所述表单UI控件的控件信息对应有控件名称；
- [0098] 基于所述表单UI控件的控件名称将从每个所述中间语言读取到的第一表单UI控件与所述配置中心从所述注册表中读取到的第二表单UI控件进行匹配,其中,所述从每个所述中间语言读取到的所述第一表单UI控件对应有接口层,所述配置中心从所述注册表中读取到的所述第二表单UI控件对应有实现层；
- [0099] 关联相匹配的所述第一表单UI控件的接口层和所述第二表单UI控件的实现层,得到一个对应有接口层和实现层的表单UI控件列表；
- [0100] 调用所述翻译器读取所述表单UI控件列表中每个所述第二表单UI控件生成具有实现层的表单UI控件。
- [0101] 本实施例中,为了将每个所述中间语言生成具有实现层的表单UI控件,将中间语言输入到翻译器,翻译器在配置中心中根据所述中间语言请求对应的字典,在配置中心中将所述字典设置为激活状态,然后从注册表中查找到所述激活状态字典对应的表单UI控件,返回所述表单UI控件至翻译器,翻译器从每个所述中间语言读取对应所述第一表单UI控件,从所述中间语言读取的所述第一表单UI控件只对应有接口层,所述配置中心从所述注册表中读取到的所述第二表单UI控件只对应有实现层,将所述第一表单UI控件与所述第二表单UI控件进行匹配,将匹配后的所述第一表单UI控件与所述第二表单UI控件进行关联,得到一个携带有接口层和实现层的表单UI控件列表,调用翻译器逐条读取所述列表中的每个所述第一表单UI控件,通过每个所述第一表单UI控件的接口层获取对应所述第二表单UI控件的实现层,最终,所述翻译器完成从中间语言生成具有实现层的表单UI控件。
- [0102] 示例性的,将所述中间语言:“我来自平安”,输入到翻译器,调用所述翻译器从所述配置中心中请求所述中间语言对应的字典并设置为激活状态,调用所述配置中心从所述注册表中读取所述字典对应的表单UI控件的控件名称:“我”“来自”“平安”,将“我”“来自”“平安”返回给翻译器,调用翻译器根据“我”“来自”“平安”将从所述中间语言读取到的第一表单UI控件“我”“来自”“平安”,所述配置中心从所述注册表中读取到的第二表单UI控件“I”、“come from”、“PingAn”,将所述第一表单UI控件与所述第二表单UI控件进行匹配得到“我”与“I”对应、“来自”与“come from”对应、“平安”与“PingAn”对应的表单UI控件列表,调用所述翻译器读取所述第二表单UI控件“I”、“come from”、“PingAn”生成具有英语实现层的表单控件,例如:所述中间语言“我”对应的具有英语实现层的表单控件为“I”,所述中间语言“来自”对应的具有英语实现层的表单控件为“come here”,所述中间语言“平安”对应的具有英语实现层的表单控件为“PingAn”。
- [0103] 本实施例中,翻译层基于所述中间语言生成具有实现层的表单UI控件,所述表单UI控件中包含了具体的样式代码和交互逻辑代码,开发人员可以在所述翻译层对每个所述表单UI控件进行二次开发,也可以对每个所述表单UI控件进行不同实现层切换,满足了各种定制场景的需求。
- [0104] S17:通过所述隔离层在所述实现层中展现具有实现层的表单UI控件。
- [0105] 本实施例中,一个表单UI控件协议层包括多个表单UI控件,每个表单UI控件对应

多个实现层,每个实现层在所述隔离层中独立展现。

[0106] 优选的,所述通过所述隔离层在所述实现层中展现具有实现层的表单UI控件包括:

[0107] 获取所述具有实现层的表单UI控件对应的多个渲染版本;

[0108] 注册所述多个渲染版本至注册表中,并将所述多个渲染版本的状态修改为挂起状态;

[0109] 确定所述注册表中的一个渲染版本为目标渲染版本,并将所述目标渲染版本配置到所述配置中心;

[0110] 修改所述配置中心中的所述目标渲染版本的状态为激活状态;

[0111] 所述翻译器从所述配置中心中获取所述状态为激活状态对应的渲染版本的实现层信息;

[0112] 在所述实现层中展现所述实现层信息对应的表单UI控件的实现层。

[0113] 本实施例中,每个表单UI控件对应多个实现层,每个实现层对应一个渲染版本,获取所述具有实现层的表单UI控件对应的多个渲染版本,将所述多个渲染版本注册到注册表中,并将状态设置为挂起状态;在配置中心中把指定的渲染版本状态设置为激活状态,这样“翻译器”会从“配置中心”获得唯一的激活状态的渲染版本的实现层信息。

[0114] 本实施例中,所述注册表、配置中心及翻译器在所述翻译层中共同作用,协同负责动态维护每个表单UI控件不同实现层之间的关系,实现不同实现层的切换。由于在每个所述表单UI控件的接口层和实现层之间架构隔离层,故每个所述表单UI控件的实现层是彼此独立,互不干扰,避免了由于浏览器提供的默认运行环境没有隔离性,实现层的内部细节很容易污染到全局环境,导致其他的UI控件受到干扰而无法正常工作。

[0115] S18:对所述实现层中的表单UI控件进行渲染,生成表单页面。

[0116] 本实施例中,所述表单页面是通过对所实现层中的表单UI控件进行渲染成不同的渲染片后进行连接生成的。

[0117] 优选的,所述对所实现层中的表单UI控件进行渲染,生成表单页面包括:

[0118] 对所实现层中的表单UI控件进行渲染得到表单UI控件的渲染片;

[0119] 基于所述渲染片共同构成整张表单页面的渲染图;

[0120] 基于所述渲染图生成表单页面。

[0121] 本实施例中,隔离层将每个表单UI控件包裹成独立的渲染片,每张渲染片在特定条件下可以独立的动态渲染而不影响其他渲染片,实现不同渲染结果,提高了浏览器的渲染效率,减少了整张表单页面的渲染成本。

[0122] 综上所述,本实施例所述的表单页面生成方法,通过创建字典;接收用户输入的待创建表单的多个表单UI控件的控件名称;从所述字典中获取所述多个表单UI控件的控件名称对应的多个表单UI控件;将每个所述表单UI控件划分为接口层和实现层,其中,每个所述表单UI控件的接口层和实现层之间架构隔离层;将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言;基于每个所述中间语言生成具有实现层的表单UI控件;通过所述隔离层在所述实现层中展现具有实现层的表单UI控件;对所实现层中的表单UI控件进行渲染,生成表单页面。本实施例一方面通过将表单UI控件划分为接口层与实现层,从而分离了表单UI控件的配置和渲染,为定制开发提供了技术基础,又通过架设隔离层,使得每个

控件的实现层彼此独立、互不干扰,通过切换实现层实现不同渲染结果的目的,提高浏览器的渲染效率;最后,将所述表单UI控件对应的接口层中的协议内容翻译成中间语言,便于后期运用在不同的技术语言中,同时基于所述中间语言生成具有实现层的表单UI控件,在不改变任意表单UI控件协议的条件对任意表单UI控件进行二次开发,满足了各种场景的定制需求。

[0123] 实施例二

[0124] 图2是本发明实施例二提供的表单页面生成装置的结构图。

[0125] 在一些实施例中,所述表单页面生成装置20可以包括多个由程序代码段所组成的功能模块。所述表单页面生成装置20中的各个程序段的程序代码可以存储于终端的存储器中,并由所述至少一个处理器所执行,以执行(详见图1描述)表单页面生成。

[0126] 本实施例中,所述表单页面生成装置20根据其所执行的功能,可以被划分为多个功能模块。所述功能模块可以包括:创建模块201、接收模块202、获取模块203、划分模块204、拼接模块205、生成模块206、展现模块207及渲染模块208。本发明所称的模块是指一种能够被至少一个处理器所执行并且能够完成固定功能的一系列计算机程序段,其存储在存储器中。在本实施例中,关于各模块的功能将在后续的实施例中详述。

[0127] 创建模块201:用于创建字典,其中所述字典携带有多个历史表单名称。

[0128] 本实施例中,所述字典是指多个历史表单的表单UI控件对应的一系列协议层的集合,所述字典中存储了每个历史表单名称、每个历史表单的表单UI控件、表单UI控件对应的协议及对应的协议层,其中,所述协议包括每个表单UI控件的控件名称、协议内容、HTMLElement标签名称、属性、方法、接口层、实现层及其参数列表。

[0129] 优选的,所述创建模块201创建字典包括:

[0130] 获取所述多个历史表单名称及每个历史表单名称对应的表单UI控件协议层,每个所述表单UI控件协议层是由所述历史表单名称对应的多个表单UI控件的接口层组成的,每个所述表单UI控件的接口层携带有对应表单UI控件的控件名称;

[0131] 将所述多个历史表单名称作为字典的第一层级,每个所述历史表单名称对应的多个表单UI控件的接口层作为字典的第二层级,多个表单UI控件的接口层对应的协议层作为字典的第三层级。

[0132] 本实施例中,每个历史表单名称对应一个协议层,每个协议层对应多个表单UI控件的接口层,每个所述表单UI控件的接口层对应一个表单UI控件,每个表单UI控件对应应有表单UI控件的控件名称,将所述历史表单名称、每个历史标点名称对应的表单UI控件及表单UI控件的接口层对应的协议层定义为不同的层级,提高了获取表单UI控件的效率。

[0133] 接收模块202:用于接收用户输入的待创建表单的多个表单UI控件的控件名称。

[0134] 本实施例中,当用户需要制作表单时,在浏览器中输入待创建表单的表单名称及所述表单名称对应的多个表单UI控件的控件名称。

[0135] 获取模块203:用于从所述字典中获取所述多个表单UI控件的控件名称对应的多个表单UI控件。

[0136] 本实施例中,所述字典中存储有每个历史表单的表单名称及每个历史表单的表单名称对应的多个表单UI控件及表单UI控件的控件名称,当接收到用户输入的待创建表单的多个表单UI控件的控件名称时,从所述字典中获取每个所述表单UI控件的控件名称对应的

表单UI控件。

[0137] 划分模块204:用于将每个所述表单UI控件划分为接口层和实现层,其中,每个所述表单UI控件的接口层和实现层之间架构隔离层。

[0138] 本实施例中,所述接口层是指定义所述表单UI控件的属性和行为,就是将所述表单UI控件的内部操作和外部环境在数据处理过程中相互独立,例如,修改所述表单UI控件的内部操作的时候不会影响到所述表单UI控件的外部环境;所述实现层是指执行所述表单UI控件的接口层定义的协议内容来实现内部操作,为了防止所述实现层的内部细节污染到全局环境,在将所述实现层和所述接口层之间架构隔离层。

[0139] 优选的,所述划分模块204将每个所述表单UI控件划分为接口层和实现层包括:

[0140] 将每个所述表单UI控件创建为浏览器能够原生识别的自定义标签;

[0141] 调用CustomElements.define传入所述自定义标签将每个所述表单UI控件划分为接口层和实现层。

[0142] 本实施例中,通过采用Web Components技术中的Custom Elements技术,把表单UI控件创建为浏览器能够原生识别的HTML Element(自定义)标签,调用customElements.define方法传入所述HTML Element标签以及使用JavaScript Class实现每个表单UI控件对应的接口层得到的实用层,从而把每个表单UI控件分为接口层和实现层,当将所述表单UI控件分为接口层和实现层之后,采用Shadow DOM技术在每个所述表单UI控件的接口层和实现层之间架构隔离层。

[0143] 本实施例中,通过将每个控件分为接口层和实现层,是为了将配置和渲染分开,其中,所述配置通过接口层进行实现,所述渲染通过实现层中进行实现,提高了渲染效率,架构隔离层是由于浏览器提供的默认运行环境没有隔离性,实现层的内部细节很容易污染到全局环境,导致其他的UI控件受到干扰而无法正常工作,而在所述实现层和所述接口层之间架构隔离层提高了渲染质量。

[0144] 拼接模块205:用于将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言。

[0145] 本实施例中,所述表单UI控件对应的接口层中的协议内容以代码的形式既不利于保存,又不利于传输,为了方便存储和传输,将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言。

[0146] 优选的,所述拼接模块205将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言包括:

[0147] 接收表单画布中用户拖曳的表单UI控件;

[0148] 根据所述用户拖曳的顺序读取所述表单画布中的表单UI控件;

[0149] 每读取到一个表单UI控件,将所述表单UI控件对应的接口层中的协议内容拼接成中间语言。

[0150] 示例性的,用户将表单UI控件列表“我”“平安”“来自”拖曳到表单画布中,根据用户拖曳的顺序读取每个表单UI控件,读取的第一个表单UI控件对应的接口层的协议内容为“我”,读取的第二个表单UI控件对应的接口层的协议内容为“来自”,读取的第三个表单UI控件对应的接口层的协议内容为“平安”,按照读取的先后顺序将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言“我来自平安”。

[0151] 生成模块206:用于基于每个所述中间语言生成具有实现层的表单UI控件。

[0152] 本实施例中,在生成具有实现层的表单UI控件的过程中,每个协议层对应多个实现层,为了达到定制的目的,设计了翻译器、配置中心和注册表,所述翻译器、配置中心和注册表协同作用在翻译层。

[0153] 优选的,所述生成模块206基于每个所述中间语言生成具有实现层的表单UI控件包括:

[0154] 将每个所述中间语言输入到翻译器中;

[0155] 调用所述翻译器从配置中心请求所述中间语言对应的字典并将所述字典设置为激活状态;

[0156] 调用所述配置中心从注册表中读取处于激活状态的字典对应的表单UI控件;

[0157] 返回所述表单UI控件至所述翻译器;

[0158] 调用所述翻译器根据所述表单UI控件从每个所述中间语言中读取对应的表单UI控件的控件信息,其中,所述表单UI控件的控件信息对应有控件名称;

[0159] 基于所述表单UI控件的控件名称将从每个所述中间语言读取到的第一表单UI控件与所述配置中心从所述注册表中读取到的第二表单UI控件进行匹配,其中,所述从每个所述中间语言读取到的所述第一表单UI控件对应有接口层,所述配置中心从所述注册表中读取到的所述第二表单UI控件对应有实现层;

[0160] 关联相匹配的所述第一表单UI控件的接口层和所述第二表单UI控件的实现层,得到一个对应有接口层和实现层的表单UI控件列表;

[0161] 调用所述翻译器读取所述表单UI控件列表中每个所述第二表单UI控件生成具有实现层的表单UI控件。

[0162] 本实施例中,为了将每个所述中间语言生成具有实现层的表单UI控件,将中间语言输入到翻译器,翻译器在配置中心中根据所述中间语言请求对应的字典,在配置中心中将所述字典设置为激活状态,然后从注册表中查找到所述激活状态字典对应的表单UI控件,返回所述表单UI控件至翻译器,翻译器从每个所述中间语言读取对应所述第一表单UI控件,从所述中间语言读取的所述第一表单UI控件只对应有接口层,所述配置中心从所述注册表中读取到的所述第二表单UI控件只对应有实现层,将所述第一表单UI控件与所述第二表单UI控件进行匹配,将匹配后的所述第一表单UI控件与所述第二表单UI控件进行关联,得到一个携带有接口层和实现层的表单UI控件列表,调用翻译器逐条读取所述列表中的每个所述第一表单UI控件,通过每个所述第一表单UI控件的接口层获取对应所述第二表单UI控件的实现层,最终,所述翻译器完成从中间语言生成具有实现层的表单UI控件。

[0163] 示例性的,将所述中间语言:“我来自平安”,输入到翻译器,调用所述翻译器从所述配置中心中请求所述中间语言对应的字典并设置为激活状态,调用所述配置中心从所述注册表中读取所述字典对应的表单UI控件的控件名称:“我”“来自”“平安”,将“我”“来自”“平安”返回给翻译器,调用翻译器读取根据“我”“来自”“平安”将从所述中间语言读取到的第一表单UI控件“我”“来自”“平安”,所述配置中心从所述注册表中读取到的第二表单UI控件生成:“I”、“come from”、“PingAn”,将所述第一表单UI控件与所述第二表单UI控件进行匹配得到“我”与“I”对应、“来自”与“come from”对应、“平安”与“PingAn”对应的表单UI控件列表,调用所述翻译器读取所述第二表单UI控件“I”、“come from”、“PingAn”生成具

有英语实现层的表单控件,例如:所述中间语言“我”对应的具有英语实现层的表单控件为“T”,所述中间语言“来自”对应的具有英语实现层的表单控件为“come here”,所述中间语言“平安”对应的具有英语实现层的表单控件为“PingAn”。

[0164] 本实施例中,翻译层基于所述中间语言生成具有实现层的表单UI控件,所述表单UI控件中包含了具体的样式代码和交互逻辑代码,开发人员可以在所述翻译层对每个所述表单UI控件进行二次开发,也可以对每个所述表单UI控件进行不同实现层切换,满足了各种定制场景的需求。

[0165] 展现模块207:用于通过所述隔离层在所述实现层中展现具有实现层的表单UI控件。

[0166] 本实施例中,一个表单UI控件协议层包括多个表单UI控件,每个表单UI控件对应多个实现层,每个实现层在所述隔离层中独立展现。

[0167] 优选的,所述展现模块207通过所述隔离层在所述实现层中展现具有实现层的表单UI控件包括:

[0168] 获取所述具有实现层的表单UI控件对应的多个渲染版本;

[0169] 注册所述多个渲染版本至注册表中,并将所述多个渲染版本的状态修改为挂起状态;

[0170] 确定所述注册表中的一个渲染版本为目标渲染版本,并将所述目标渲染版本配置到所述配置中心;

[0171] 修改所述配置中心中的所述目标渲染版本的状态为激活状态;

[0172] 所述翻译器从所述配置中心中获取所述状态为激活状态对应的渲染版本的实现层信息;

[0173] 在所述实现层中展现所述实现层信息对应的表单UI控件的实现层。

[0174] 本实施例中,每个表单UI控件对应多个实现层,每个实现层对应一个渲染版本,获取所述具有实现层的表单UI控件对应的多个渲染版本,将所述多个渲染版本注册到注册表中,并将状态设置为挂起状态;在配置中心中把指定的渲染版本状态设置为激活状态,这样“翻译器”会从“配置中心”获得唯一的激活状态的渲染版本的实现层信息。

[0175] 本实施例中,所述注册表、配置中心及翻译器在所述翻译层中共同作用,协同负责动态维护每个表单UI控件不同实现层之间的关系,实现不同实现层的切换。由于在每个所述表单UI控件的接口层和实现层之间架构隔离层,故每个所述表单UI控件的实现层是彼此独立,互不干扰,避免了由于浏览器提供的默认运行环境没有隔离性,实现层的内部细节很容易污染到全局环境,导致其他的UI控件受到干扰而无法正常工作。

[0176] 渲染模块208:用于对所述实现层中的表单UI控件进行渲染,生成表单页面。

[0177] 本实施例中,所述表单页面是通过对所述实现层中的表单UI控件进行渲染成不同的渲染片后进行连接生成的。

[0178] 优选的,所述渲染模块208对所述实现层中的表单UI控件进行渲染,生成表单页面包括:

[0179] 对所述实现层中的表单UI控件进行渲染得到表单UI控件的渲染片;

[0180] 基于所述渲染片共同构成整张表单页面的渲染图;

[0181] 基于所述渲染图生成表单页面。

[0182] 本实施例中,隔离层将每个表单UI控件包裹成独立的渲染片,每张渲染片在特定条件下可以独立的动态渲染而不影响其他渲染片,实现不同渲染结果,提高了浏览器的渲染效率,减少了整张表单页面的渲染成本。

[0183] 综上所述,本实施例所述的表单页面生成装置,通过创建字典;接收用户输入的待创建表单的多个表单UI控件的控件名称;从所述字典中获取所述多个表单UI控件的控件名称对应的多个表单UI控件;将每个所述表单UI控件划分为接口层和实现层,其中,每个所述表单UI控件的接口层和实现层之间架构隔离层;将每个所述表单UI控件对应的接口层中的协议内容拼接成中间语言;基于每个所述中间语言生成具有实现层的表单UI控件;通过所述隔离层在所述实现层中展现具有实现层的表单UI控件;对所述实现层中的表单UI控件进行渲染,生成表单页面。本实施例一方面通过将表单UI控件划分为接口层与实现层,从而分离了表单UI控件的配置和渲染,为定制开发提供了技术基础,又通过架设隔离层,使得每个控件的实现层彼此独立、互不干扰,通过切换实现层实现不同渲染结果的目的,提高浏览器的渲染效率;最后,将所述表单UI控件对应的接口层中的协议内容翻译成中间语言,便于后期运用在不同的技术语言中,同时基于所述中间语言生成具有实现层的表单UI控件,在不改变任意表单UI控件协议的条件对任意表单UI控件进行二次开发,满足了各种场景的定制需求。

[0184] 实施例三

[0185] 参阅图3所示,为本发明实施例三提供的终端的结构示意图。在本发明较佳实施例中,所述终端3包括存储器31、至少一个处理器32、至少一条通信总线33及收发器34。

[0186] 本领域技术人员应该了解,图3示出的终端的结构并不构成本发明实施例的限定,既可以是总线型结构,也可以是星形结构,所述终端3还可以包括比图示更多或更少的其他硬件或者软件,或者不同的部件布置。

[0187] 在一些实施例中,所述终端3是一种能够按照事先设定或存储的指令,自动进行数值计算和/或信息处理的终端,其硬件包括但不限于微处理器、专用集成电路、可编程门阵列、数字处理器及嵌入式设备等。所述终端3还可包括客户设备,所述客户设备包括但不限于任何一种可与客户通过键盘、鼠标、遥控器、触摸板或声控设备等方式进行人机交互的电子产品,例如,个人计算机、平板电脑、智能手机、数码相机等。

[0188] 需要说明的是,所述终端3仅为举例,其他现有的或今后可能出现的电子产品如可适应于本发明,也应包含在本发明的保护范围以内,并以引用方式包含于此。

[0189] 在一些实施例中,所述存储器31用于存储程序代码和各种数据,例如安装在所述终端3中的表单页面生成装置20,并在终端3的运行过程中实现高速、自动地完成程序或数据的存取。所述存储器31包括只读存储器(Read-Only Memory,ROM)、可编程只读存储器(Programmable Read-Only Memory,PROM)、可擦除可编程只读存储器(Erasable Programmable Read-Only Memory,EPR0M)、一次可编程只读存储器(One-time Programmable Read-Only Memory,OTPROM)、电子擦除式可复写只读存储器(Electrically-Erasable Programmable Read-Only Memory,EEPROM)、只读光盘(Compact Disc Read-Only Memory,CD-ROM)或其他光盘存储器、磁盘存储器、磁带存储器、或者能够用于携带或存储数据的计算机可读的任何其他介质。

[0190] 在一些实施例中,所述至少一个处理器32可以由集成电路组成,例如可以由单个

封装的集成电路所组成,也可以是由多个相同功能或不同功能封装的集成电路所组成,包括一个或者多个中央处理器(Central Processing unit,CPU)、微处理器、数字处理芯片、图形处理器及各种控制芯片的组合等。所述至少一个处理器32是所述终端3的控制核心(Control Unit),利用各种接口和线路连接整个终端3的各个部件,通过运行或执行存储在所述存储器31内的程序或者模块,以及调用存储在所述存储器31内的数据,以执行终端3的各种功能和处理数据。

[0191] 在一些实施例中,所述至少一条通信总线33被设置为实现所述存储器31以及所述至少一个处理器32等之间的连接通信。

[0192] 尽管未示出,所述终端3还可以包括给各个部件供电的电源(比如电池),优选的,电源可以通过电源管理装置与所述至少一个处理器32逻辑相连,从而通过电源管理装置实现管理充电、放电、以及功耗管理等功能。电源还可以包括一个或一个以上的直流或交流电源、再充电装置、电源故障检测电路、电源转换器或者逆变器、电源状态指示器等任意组件。所述终端3还可以包括多种传感器、蓝牙模块、Wi-Fi模块等,在此不再赘述。

[0193] 应该了解,所述实施例仅为说明之用,在专利申请范围上并不受此结构的限制。

[0194] 上述以软件功能模块的形式实现的集成的单元,可以存储在一个计算机可读存储介质中。上述软件功能模块存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,终端,或者网络设备等)或处理器(processor)执行本发明各个实施例所述方法的部分。

[0195] 在进一步的实施例中,结合图2(装置模块),所述至少一个处理器32可执行所述终端3的操作装置以及安装的各类应用程序(如所述的表单页面生成装置20)、程序代码等,例如,上述的各个模块。

[0196] 所述存储器31中存储有程序代码,且所述至少一个处理器32可调用所述存储器31中存储的程序代码以执行相关的功能。例如,图2中所述的各个模块是存储在所述存储器31中的程序代码,并由所述至少一个处理器32所执行,从而实现所述各个模块的功能以达到表单页面生成的目的。

[0197] 在本发明的一个实施例中,所述存储器31存储多个指令,所述多个指令被所述至少一个处理器32所执行以实现表单页面的生成。

[0198] 具体地,所述至少一个处理器32对上述指令的具体实现方法可参考图1对应实施例中相关步骤的描述,在此不赘述。

[0199] 在本发明所提供的几个实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述模块的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式。

[0200] 进一步地,所述计算机可读存储介质可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序等;存储数据区可存储根据区块链节点的使用所创建的数据等。

[0201] 本发明所指区块链是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。区块链(Blockchain),本质上是一个去中心化的数据库,是一串使用密码学方法相关联产生的数据块,每一个数据块中包含了一批网络交易的信息,用于验证其信息的有效性(防伪)和生成下一个区块。区块链可以包括区块链底层平台、平台产品

服务层以及应用服务层等。

[0202] 所述作为分离部件说明的模块可以是或者也可以不是物理上分开的,作为模块显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。

[0203] 另外,在本发明各个实施例中的各功能模块可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用硬件加软件功能模块的形式实现。

[0204] 对于本领域技术人员而言,显然本发明不限于上述示范性实施例的细节,而且在不背离本发明的精神或基本特征的情况下,能够以其他的具体形式实现本发明。因此,无论从哪一点来看,均应将实施例看作是示范性的,而且是非限制性的,本发明的范围由所附权利要求而不是上述说明限定,因此旨在将落在权利要求的等同要件的含义和范围内的所有变化涵括在本发明内。不应将权利要求中的任何附图标记视为限制所涉及的权利要求。此外,显然“包括”一词不排除其他单元或,单数不排除复数。装置权利要求中陈述的多个单元或装置也可以由一个单元或装置通过软件或者硬件来实现。第一,第二等词语用来表示名称,而并不表示任何特定的顺序。

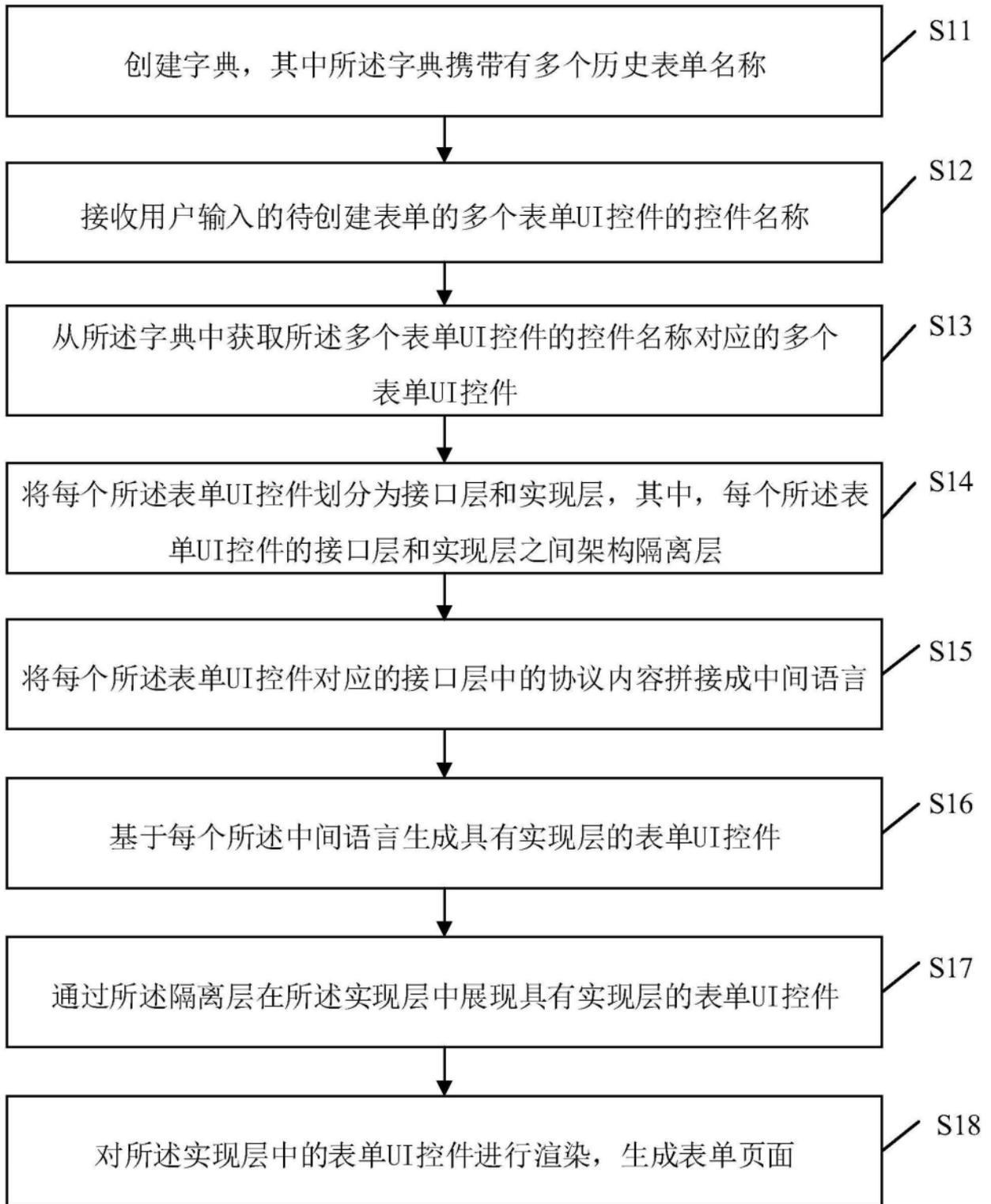


图1

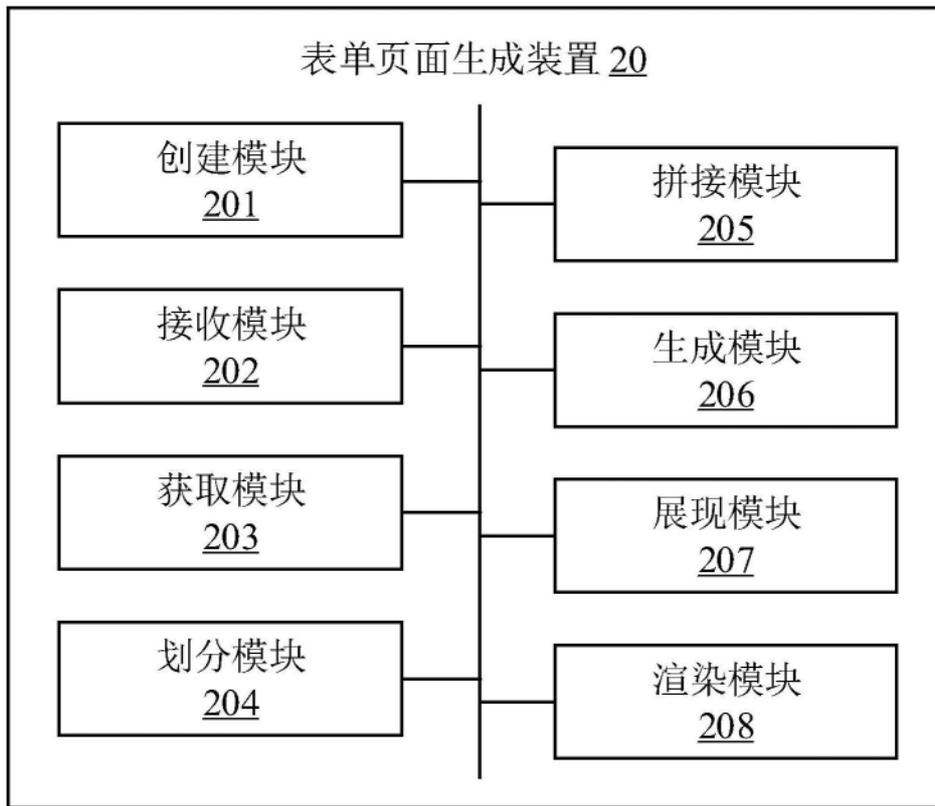


图2

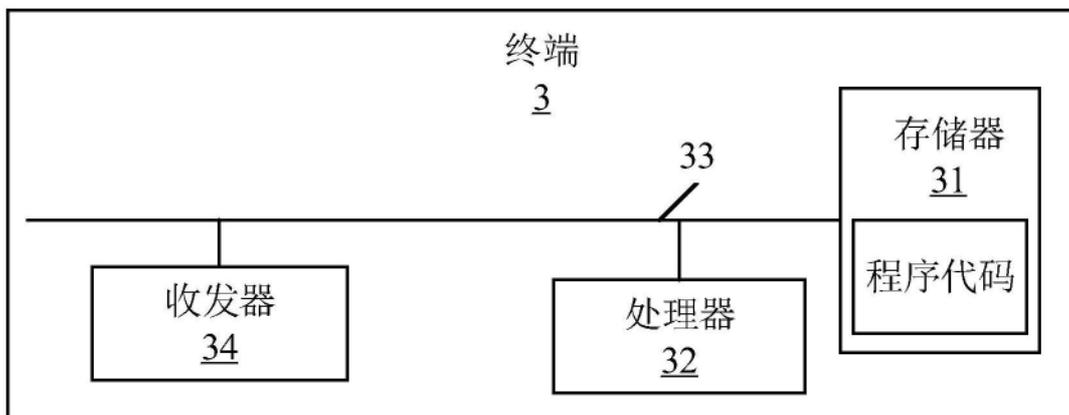


图3