



(12)发明专利

(10)授权公告号 CN 106250156 B

(45)授权公告日 2019.03.12

(21)申请号 201610624773.6

(22)申请日 2016.08.01

(65)同一申请的已公布的文献号  
申请公布号 CN 106250156 A

(43)申请公布日 2016.12.21

(73)专利权人 腾讯科技(深圳)有限公司  
地址 518000 广东省深圳市福田区振兴路  
赛格科技园2栋东403室

(72)发明人 伍林

(74)专利代理机构 广州华进联合专利商标代理  
有限公司 44224  
代理人 何平 邓云鹏

(51)Int.Cl.  
G06F 8/35(2018.01)  
G06F 8/38(2018.01)

(56)对比文件

CN 105808237 A,2016.07.27,  
CN 103279358 A,2013.09.04,  
CN 101030139 A,2007.09.05,  
US 2012297360 A1,2012.11.22,  
CN 101751258 A,2010.06.23,

审查员 韩典伯

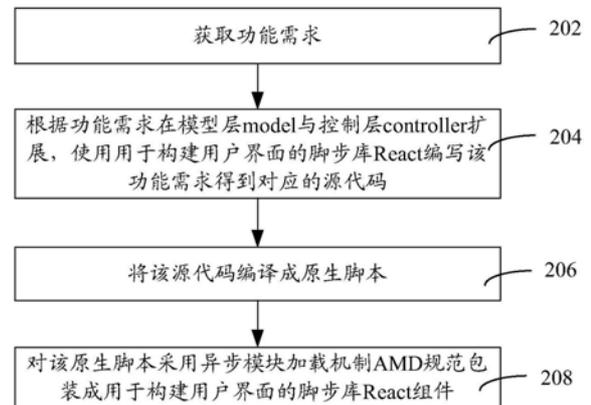
权利要求书2页 说明书10页 附图6页

(54)发明名称

组件构建方法和装置、组件调用方法和装置

(57)摘要

本发明涉及一种组件构建方法和装置、组件调用方法和装置。所述组件构建方法包括:获取功能需求;根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React编写该功能需求得到对应的源代码;将所述源代码编译成原生脚本;对所述原生脚本采用异步模块加载机制AMD规范包装成React组件。上述组件构建方法和装置、组件调用方法和装置,通过使用React的语法编写功能需要对应的源代码,并将源代码编译成原生脚本,将原生脚本通过AMD规范包装成React组件,被使用时,通过简单的调用即可使用React组件,扩大了使用范围,且组件可复用性高,不需要开发者专门学习React即可使用React组件,降低了开发者的学习门槛。



1. 一种组件构建方法,包括:

获取功能需求;

根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚本库React编写所述功能需求得到对应的源代码;

将所述源代码编译成原生脚本;

对所述原生脚本采用异步模块加载机制AMD规范包装成用于构建用户界面的脚本库React组件;

所述用于构建用户界面的脚本库React组件包括按钮组件、下拉组件、数值输入框组件、滑动条组件、气泡提示组件和表格组件。

2. 根据权利要求1所述的方法,其特征在于,所述根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚本库React编写所述功能需求得到对应的源代码,包括:

根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚本库React编写定义所述功能需求对应的脚本库React组件的初始化参数;

定义外部修改所述脚本库React组件的参数时所触发的函数;

定义所述脚本库React组件的值改变时的处理函数;

定义失去焦点时处理函数;

定义声明脚本库React组件的函数。

3. 根据权利要求1所述的方法,其特征在于,所述将所述源代码编译成原生脚本,包括:

采用脚本编译器工具编辑所述源代码生成原生脚本。

4. 一种组件调用方法,包括:

获取业务需求;

根据所述业务需求调用对应的用于构建用户界面的脚本库React组件,所述用于构建用户界面的脚本库React组件为根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚本库React编写所述功能需求得到对应的源代码,并将所述源代码编译成原生脚本,以及对所述原生脚本采用异步模块加载机制AMD规范包装得到的用于构建用户界面的脚本库React组件;

获取对所述用于构建用户界面的脚本库React组件配置的参数;

将所述配置的参数传递给对应的用于构建用户界面的脚本库React组件,形成包含所述参数的用于构建用户界面的脚本库React组件;

展示包含所述参数的所述用于构建用户界面的脚本库React组件;

所述用于构建用户界面的脚本库React组件包括按钮组件、下拉组件、数值输入框组件、滑动条组件、气泡提示组件和表格组件。

5. 一种组件构建装置,其特征在于,包括:

功能需求获取模块,用于获取功能需求;

编写模块,用于根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚本库React编写所述功能需求得到对应的源代码;

生成模块,用于将所述源代码编译成原生脚本;

封装模块,用于对所述原生脚本采用异步模块加载机制AMD规范包装成用于构建用户

界面的脚本库React组件；

所述用于构建用户界面的脚本库React组件包括按钮组件、下拉组件、数值输入框组件、滑动条组件、气泡提示组件和表格组件。

6. 根据权利要求5所述的装置,其特征在于,所述编写模块还用于根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚本库React编写定义所述功能需求对应的脚本库React组件的初始化参数,定义外部修改所述脚本库React组件的参数时所触发的函数,定义所述脚本库React组件的值改变时的处理函数,定义失去焦点时处理函数,以及定义声明脚本库React组件的函数。

7. 根据权利要求5所述的装置,其特征在于,所述生成模块还用于采用脚本编译器工具编辑所述源代码生成原生脚本。

8. 一种组件调用装置,其特征在于,包括:

业务需求获取模块,用于获取业务需求;

调用模块,用于根据所述业务需求调用对应的用于构建用户界面的脚本库React组件,所述用于构建用户界面的脚本库React组件为根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚本库React编写所述功能需求得到对应的源代码,并将所述源代码编译成原生脚本,以及对所述原生脚本采用异步模块加载机制AMD规范包装得到的用于构建用户界面的脚本库React组件;

参数配置模块,用于获取对所述用于构建用户界面的脚本库React组件配置参数;

传递模块,用于将所述配置参数传递给对应的用于构建用户界面的脚本库React组件,形成包含所述参数的用于构建用户界面的脚本库React组件;

展示模块,用于展示包含所述配置参数的所述用于构建用户界面的脚本库React组件;

所述用于构建用户界面的脚本库React组件包括按钮组件、下拉组件、数值输入框组件、滑动条组件、气泡提示组件和表格组件。

## 组件构建方法和装置、组件调用方法和装置

### 技术领域

[0001] 本发明涉及计算机应用领域,特别是涉及一种组件构建方法和装置、组件调用方法和装置。

### 背景技术

[0002] AngularJS是建立在轻量jQuery之上的一个结构化前端MVVM(Model View View Model)框架,用于构建单页面Web应用。AngularJS是大而全的框架,用来开发组件学习成本大,双向数据绑定在复杂的交互情况下性能陡然下降。

[0003] React是由Facebook推出的一个JavaScript框架,React是一个用于构建组件化UI(User Interface,用户界面)的库,是一个前端界面开发工具。因React只负责view层,开发者在model、controller层需要花费很大的精力,并且JSX语法编写的代码需要通过打包工具webpack、babel编译成原生javascript语法,在开发项目中难以大范围推广使用。

### 发明内容

[0004] 基于此,有必要针对传统构建组件方式难以大范围推广使用的问题,提供一种组件构建方法和装置,能扩大使用范围。

[0005] 此外,还有必要提供一种组件调用方法和装置,能扩大使用范围

[0006] 一种组件构建方法,包括:

[0007] 获取功能需求;

[0008] 根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React的语法编写所述功能需求得到对应的源代码;

[0009] 将所述源代码编译成原生脚本;

[0010] 对所述原生脚本采用异步模块加载机制AMD规范包装成用于构建用户界面的脚步库React组件。

[0011] 一种组件调用方法,包括:

[0012] 获取业务需求;

[0013] 根据所述业务需求调用对应的用于构建用户界面的脚步库React组件,所述用于构建用户界面的脚步库React组件为根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React的语法编写所述功能需求得到对应的源代码,并将所述源代码编译成原生脚本,以及对所述原生脚本采用异步模块加载机制AMD规范包装得到的用于构建用户界面的脚步库React组件;

[0014] 获取对所述用于构建用户界面的脚步库React组件配置的参数;

[0015] 将所述配置的参数传递给对应的用于构建用户界面的脚步库React组件,形成包含所述参数的用于构建用户界面的脚步库React组件;

[0016] 展示包含所述参数的所述用于构建用户界面的脚步库React组件。

[0017] 一种组件构建装置,包括:

- [0018] 功能需求获取模块,用于获取功能需求;
- [0019] 编写模块,用于根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React编写所述功能需求得到对应的源代码;
- [0020] 生成模块,用于将所述源代码编译成原生脚本;
- [0021] 封装模块,用于对所述原生脚本采用异步模块加载机制AMD规范包装成用于构建用户界面的脚步库React组件。
- [0022] 一种组件调用装置,包括:
- [0023] 业务需求获取模块,用于获取业务需求;
- [0024] 调用模块,用于根据所述业务功能需求调用对应的用于构建用户界面的脚步库React组件,所述用于构建用户界面的脚步库React组件为根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React的语法编写所述功能需求得到对应的源代码,并将所述源代码编译成原生脚本,以及对所述原生脚本采用异步模块加载机制AMD规范包装得到的用于构建用户界面的脚步库React组件;
- [0025] 参数配置模块,用于获取对所述用于构建用户界面的脚步库React组件配置的参数;
- [0026] 传递模块,用于将所述配置的参数传递给对应的用于构建用户界面的脚步库React组件,形成包含所述参数的用于构建用户界面的脚步库React组件;
- [0027] 展示模块,用于展示包含所述配置参数的所述用于构建用户界面的脚步库React组件。
- [0028] 上述组件构建方法和装置,通过使用用于构建用户界面的脚步库React编写功能需要对应的源代码,并将源代码编译成原生脚本,将原生脚本通过AMD规范包装成React组件,被使用时,通过简单的调用即可使用React组件,扩大了使用范围,且组件可复用性高,能够被AMD、webpack模块管理器加载,功能丰富,扩展了原生html不具备的特性,且不需要开发者专门学习React即可使用React组件,降低了开发者的学习门槛,同时组件提供的丰富功能能满足开发组日常需求,提升了新产品售卖开发效率,缩短了产品上线周期。
- [0029] 上述组件调用方法和装置,根据业务需求调用对应的React组件,并获取React组件配置的参数,将参数传递给React组件,形成包含该参数的React组件,并展示该React组件,调用方式简单,不需要开发者专门学习React即可使用React组件,降低了开发者的学习门槛,同时组件提供的丰富功能能满足开发组日常需求,提升了新产品售卖开发效率,缩短了产品上线周期。

## 附图说明

- [0030] 图1为一个实施例中电子设备的内部结构示意图;
- [0031] 图2为一个实施例中组件构建方法的流程图;
- [0032] 图3A为按钮组件的示意图;
- [0033] 图3B为下拉组件的示意图;
- [0034] 图3C为数值输入框组件的示意图;
- [0035] 图3D为滑动条组件的示意图;
- [0036] 图3E为气泡提示组件的示意图;

- [0037] 图3F为表格组件的示意图；
- [0038] 图4为根据功能需求在model层与controller层扩展,使用React的语法编写该功能需求得到对应的源代码的具体流程示意图；
- [0039] 图5为一个实施例中组件调用方法的流程图；
- [0040] 图6为调用组件形成页面的示意图；
- [0041] 图7为一个实施例中组件调用方法应用于构建云产品售卖页的架构图；
- [0042] 图8为一个实施例中组件构建装置的结构框图；
- [0043] 图9为一个实施例中组件调用装置的结构框图。

### 具体实施方式

[0044] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0045] 可以理解,本发明所使用的术语“第一”、“第二”等可在本文中用于描述各种元件,但这些元件不受这些术语限制。这些术语仅用于将第一个元件与另一个元件区分。

[0046] 图1为一个实施例中电子设备的内部结构示意图。如图1所示,该电子设备包括通过系统总线连接的处理器、非易失性存储介质、内存储器、网络接口、显示屏和输入装置。其中,电子设备的非易失性存储介质存储有操作系统,还包括一种组件构建装置和组件调用装置,该组件构建装置用于实现一种组件构建装置方法,该组件调用装置用于实现一种组件调用装置方法。该处理器用于提供计算和控制能力,支撑整个电子设备的运行。电子设备中的内存储器为非易失性存储介质中的组件构建装置和组件调用装置的运行提供环境,该内存储器中可储存有计算机可读指令,该计算机可读指令被所述处理器执行时,可使得所述处理器执行一种组件构建方法和组件调用方法。网络接口用于与其他设备进行网络通信。电子设备的显示屏可以是液晶显示屏或者电子墨水显示屏等,输入装置可以是显示屏上覆盖的触摸层,也可以是电子设备外壳上设置的按键、轨迹球或触控板,也可以是外接的键盘、触控板或鼠标等。该电子设备可以是手机、个人计算机、平板电脑、个人数字助理、穿戴式设备或服务器等。本领域技术人员可以理解,图1中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的电子设备的限定,具体的电子设备可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0047] 图2为一个实施例中组件构建方法的流程图。如图2所示,一种组件构建方法,包括:

[0048] 步骤202,获取功能需求。

[0049] 功能需求是指提供某一服务所需要的功能。如地址选择服务,可采用按钮选择功能或下拉框功能等实现。

[0050] 步骤204,根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React编写该功能需求得到对应的源代码。

[0051] 本实施例中,在model层与controller层扩展,React内部使用JSX语法编写该功能需求得到对应的源代码。JSX是一个看起来很像XML (Extensible Markup Language,可扩展

标记语言)的JavaScript语法扩展。

[0052] model层是模型层,是应用程序中用于处理应用程序数据逻辑的部分,通常模型对象负责在数据库中存取数据。模型表示企业数据和业务规则。controller层是控制器层,是应用程序中处理用户交互的部分,通常控制器负责从视图读取数据,控制用户输入,并向模型层发送数据。

[0053] React是由Facebook推出的一个JavaScript框架,用于构建用户界面的Javascript(脚步)库。React实现了单向响应的数据流。

[0054] 步骤206,将该源代码编译成原生脚本。

[0055] 本实施例中,将源代码编译成原生JavaScript(脚本)。

[0056] 步骤208,对该原生脚本采用异步模块加载机制AMD规范包装成用于构建用户界面的脚步库React组件。

[0057] 本实施例中,AMD规范是指Asynchronous Module Definition,异步模块加载机制。AMD规范完整描述了模块的定义,依赖关系,引用关系以及加载机制。

[0058] 该用于构建用户界面的脚步库React组件包括按钮组件、下拉组件、数值输入框组件、滑动条组件、气泡提示组件和表格组件。React组件通过一个render()方法,接收输入的参数并返回展示的对象。

[0059] 如图3A所示,按钮组件可用于获取调用者传入渲染组件的数据后,在按钮某一项选中的时候通知调用方当前选中的值。调用者还可以传入参数将按钮分类、禁用灰掉、上方气泡提示等。按钮包括广州、上海、上海金融、北京、香港、多伦多。

[0060] 如图3B所示,下拉组件可用于获取调用者传入渲染组件的数据后,在下拉框中某一项选中的时候通知调用方当前选中的值。调用者还可以传入参数将下拉某一项禁用、右侧气泡提示、每一项支持最右侧展示折扣促销图标等。

[0061] 如图3C所示,数值输入框组件用于获取调用者传入的最小值、最大值以限制用户输入的数值的区间。比如传入{min:1,max:50}。用户只能输入1至50这个区间的值。若用户输入值不在该区间内,数值输入框组件自动将值修复成最小值或最大值。

[0062] 如图3D所示,滑动条组件可用于获取调用者传入的最小值、最大值以限制用户输入的数值的区间。滑动条更新会同步输入框更新,输入框数值更新会同步滑动条的位置。数值更新会通知调用者当前的值。

[0063] 如图3E所示,气泡提示组件可用于展示气泡提示。用户可自由选择气泡展示在目标元素的位置(例如上、下、左、右)。当浏览器缩小宽高时,气泡会自适应调整位置,避免被挡住看不见。

[0064] 如图3F所示,表格组件可获取调用者传入展示的json数据即可展示表格。表格支持单选、多选、排序等。当用户选中一行或多行时,表格组件会通知调用者此时选中的数据。

[0065] 上述组件构建方法,通过使用React的语法编写功能需要对应的源代码,并将源代码编译成原生脚本,将原生脚本通过AMD规范包装成React组件,被使用时,通过简单的调用即可使用React组件,扩大了使用范围,且组件可复用性高,能够被AMD、webpack模块管理器加载,功能丰富,扩展了原生html不具备的特性,且不需要开发者专门学习React即可使用React组件,降低了开发者的学习门槛,同时组件提供的丰富功能能满足开发组日常需求,提升了新产品售卖开发效率,缩短了产品上线周期。

[0066] 如图4所示,在一个实施例中,该根据功能需求在model层与controller层扩展,使用React的语法编写该功能需求得到对应的源代码,包括:

[0067] 步骤402,根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React编写定义该功能需求对应的组件的初始化参数。

[0068] 步骤404,定义外部修改该组件参数时所触发的函数。

[0069] 步骤406,定义该组件的值改变时的处理函数。

[0070] 步骤408,定义该失去焦点时处理函数。

[0071] 步骤410,定义声明组件的函数。

[0072] 通过定义组件的参数触发函数、改变时处理函数、失去焦点时处理函数等,便于对组件实现数据处理。

[0073] 在一个实施例中,该将该源代码编译成原生脚本,包括:采用bable工具编辑该源代码生成原生脚本。

[0074] 以一个数值输入框组件的实现代码进行说明,具体代码如下:

```
define('widget/number', function (require, exports, module) {
```

```
    var $ = require('$');
```

```
    var _ = require('underscore');
```

```
    var React = require('React');
```

```
    //NumberComponent 是 React 的实现
```

```
    var NumberComponent = React.createClass({
```

```
        getInitialState: function() {
```

```
            //处理组件初始化参数设置
```

[0075]

```
        },
```

```
        componentWillReceiveProps: function(nextProps) {
```

```
            //外部修改组件参数时触发
```

```
        },
```

```
        clickHandle: function(e) {
```

```
            //点击数值组件的加一减一按钮时处理
```

```
    },  
  
    changeHandle: function() {  
        //数值输入框组件的值改变时处理函数  
    },  
  
    blurHandle: function() {  
        //失去焦点时处理函数  
    },  
  
    render: function() {  
        //此函数通地 jsx 语方声明组件, bable 工具会编辑成原生 html  
    }  
});  
[0076] //NumberWidget 是对 react 组件 NumberComponent 类的 amd 包装  
var NumberWidget = function(options) {  
    var defaults = {};  
    var options = this.options = $.extend({}, defaults, options);  
    if (options.value < options.min) options.value = options.min;  
    if (options.value > options.max) options.value = options.max;  
    this.render();  
};  
NumberWidget.prototype = {  
    //render 渲染 react 组件  
    render: function() {  
        var self = this;  
        var options = this.options;  
        self.reactComponent = React.render(  

```

[0077]

```
        <NumberComponent
          onChange={self.change.bind(self)}
          {...self.options}/>,
        $(self.options.renderTo)[0]
      );
    },
    //组件的值变化时会通知组件调用方
    change: function(value, triggerType) {

    },
    //获取或者设置组件的值
    val: function(value) {

    },
    //更新组件的参数
    setOptions: function(options) {

    }
  };
  //真正返回的是一个标准 AMD 组件,所以 React 内部的 jsx 语方对调用者是
  屏蔽的
  exports.create = function(options) {
    var mod = new NumberWidget(options);
    return mod;
  };
  }。
```

[0078] 其他React组件实现方式类似上述实现数值输入框组件。

[0079] 在一个实施例中,上述组件构建方法还包括:获取包含组件标识的组件更新指令,根据该组件更新指令对该组件标识对应的组件进行更新。

[0080] 组件标识是用来唯一表示组件的字符串。

[0081] React组件更新,不是全局刷新,而是通过它内部的dom(Document Object Model, 文档对象模型)diff算法计算出不同点,然后以最小粒度进行更新,所以组件渲染效率和性能非常高。

[0082] 图5为一个实施例中组件调用方法的流程图。如图5所示,一种组件调用方法,包括步骤502~步骤510。其中:

[0083] 步骤502,获取业务需求。

[0084] 本实施例中,业务需求是指网页构建所提供的服务需求。

[0085] 步骤504,根据该业务需求调用对应的用于构建用户界面的脚步库React组件,该用于构建用户界面的脚步库React组件为根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React编写该功能需求得到对应的源代码,并将该源代码编译成原生脚本,以及对该原生脚本采用异步模块加载机制AMD规范包装得到的用于构建用户界面的脚步库React组件。

[0086] 该React组件包括按钮组件、下拉组件、数值输入框组件、滑动条组件、气泡提示组件和表格组件。

[0087] 采用异步模块加载机制AMD模块加载器加载对应的用于构建用户界面的脚步库React组件即可展示在界面上。

[0088] 步骤506,获取对该用于构建用户界面的脚步库React组件配置的参数。

[0089] 本实施例中,对于不同的React组件可配置不同的参数。例如React组件为数值输入框组件,配置的参数的数值为1。

[0090] 步骤508,将该配置的参数传递给对应的用于构建用户界面的脚步库React组件,形成包含该参数的用于构建用户界面的脚步库React组件。

[0091] 例如,将参数的数值1传递给数值输入框组件,然后显示该数值输入框的数值为1。

[0092] 步骤510,展示包含该参数的该用于构建用户界面的脚步库React组件。

[0093] 上述组件调用方法,根据业务需求调用对应的用于构建用户界面的脚步库React组件,并获取React组件配置的参数,将参数传递给用于构建用户界面的脚步库React组件,形成包含该参数的用于构建用户界面的脚步库React组件,并展示该React组件,调用方式简单,不需要开发者专门学习React即可使用React组件,降低了开发者的学习门槛,同时组件提供的丰富功能能满足开发组日常需求,提升了新产品售卖开发效率,缩短了产品上线周期。

[0094] 图6为调用组件形成页面的示意图。如图6所示,开发是所能用到的组件包括按钮组件、下拉组件、数值输入框组件、滑动条组件、气泡提示组件和表格组件。页面A调用滑动条组件和下拉组件,页面B调用按钮组件、表格组件和数值输入框组件。调用时采用AMD加载器加载组件到页面A或页面B。

[0095] 原React组件调用方式是:

[0096] `React.render(`

[0097] `<dropdown {...options} />,`

[0098] `Document.body`

[0099] `);`

[0100] 开发者需要使用babel等工具将自定义JSX语法编译成原生的javascript语法。

[0101]

```
React.render(  
    React.createElement(DropdownComponent,React._spread({},options)),  
    Document.body  
);
```

[0102] 而采用上述组件调用方法,组件的调用方式少了编译的过程,直接使用AMD模块加载器即可展示在界面如:

```
[0103] var dropdown=require('dropdown');
```

```
[0104] Dropdown.create(options)。
```

[0105] 为了更清楚的说明组件调用方法的实现,以组件调用方法应用于云产品售卖页为例进行详细描述。如图7所示,自助接入页获取云业务开发者配置的云产品售卖页的相关参数,并根据相关参数加载业务方的云产品售卖页的逻辑代码。云产品售卖页是开发者在React框架的基础上构建的售卖逻辑,包括进行询价、下单、支付等。React框架扩展了原生html的组件,异步数据请求组件,页面性能监控组件,云计费接口等。云业务开发者根据云产品售卖页的业务需求调用对应的组件构建云产品售卖页;获取用户在组件上操作选择云服务配置;通过组件提供的接口获取用户选择的配置参数,并将配置参数提交给询价组件,返回价格展示在云产品售卖页的页面上;获取到用户的购买操作,将参数提交给支付页,支付页将参数提交给云后台生成订单,生成订单后跳转至第三方支付平台或银行卡支付界面进行支付完成购买。

[0106] 图8为一个实施例中组件构建装置的结构框图。如图8所示,一种组件构建装置,包括功能需求获取模块802、编写模块804、生成模块806和封装模块808。其中:

[0107] 功能需求获取模块802用于获取功能需求。

[0108] 编写模块804用于根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React编写该功能需求得到对应的源代码。

[0109] 生成模块806用于将该源代码编译成原生脚本。

[0110] 封装模块808用于对该原生脚本采用异步模块加载机制AMD规范包装成用于构建用户界面的脚步库React组件。

[0111] 该用于构建用户界面的脚步库React组件包括按钮组件、下拉组件、数值输入框组件、滑动条组件、气泡提示组件和表格组件。

[0112] 上述组件构建装置,通过使用React的语法编写功能需要对应的源代码,并将源代码编译成原生脚本,将原生脚本通过AMD规范包装成React组件,被使用时,通过简单的调用即可使用React组件,扩大了使用范围,且组件可复用性高,能够被AMD、webpack模块管理器加载,功能丰富,扩展了原生html不具备的特性,且不需要开发者专门学习React即可使用React组件,降低了开发者的学习门槛,同时组件提供的丰富功能能满足开发组日常需求,提升了新产品售卖开发效率,缩短了产品上线周期。

[0113] 在一个实施例中,该编写模块804还用于根据功能需求在model层与controller层扩展,使用React的语法编写定义该功能需求对应的组件的初始化参数,定义外部修改该组件参数时所触发的函数,定义该组件的值改变时的处理函数,定义该失去焦点时处理函数,

以及定义声明组件的函数。

[0114] 在一个实施例中,该生成模块806还用于采用脚本编译器(如babel)工具编辑该源代码生成原生脚本。

[0115] 图9为一个实施例中组件调用装置的结构框图。如图9所示,一种组件调用装置,包括业务需求获取模块902、调用模块904、参数配置模块906、传递模块908和展示模块910。其中:

[0116] 业务需求获取模块902用于获取业务需求。

[0117] 调用模块904用于根据该业务功能需求调用对应的用于构建用户界面的脚步库React组件,该用于构建用户界面的脚步库React组件为根据功能需求在模型层model与控制层controller扩展,使用用于构建用户界面的脚步库React的语法编写该功能需求得到对应的源代码,并将该源代码编译成原生脚本,以及对该原生脚本采用异步模块加载机制AMD规范包装得到的用于构建用户界面的脚步库React组件。

[0118] 参数配置模块906用于获取对该用于构建用户界面的脚步库React组件配置参数。

[0119] 传递模块908用于将该配置参数传递给对应的用于构建用户界面的脚步库React组件,形成包含该参数的用于构建用户界面的脚步库React组件。

[0120] 展示模块910用于展示包含该配置参数的该用于构建用户界面的脚步库React组件。

[0121] 上述组件调用装置,根据业务需求调用对应的React组件,并获取React组件配置参数,将参数传递给React组件,形成包含该参数的React组件,并展示该React组件,调用方式简单,不需要开发者专门学习React即可使用React组件,降低了开发者的学习门槛,同时组件提供的丰富功能能满足开发组日常需求,提升了新产品售卖开发效率,缩短了产品上线周期。

[0122] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于一非易失性计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)等。

[0123] 以上所述实施例仅表达了本发明的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对本发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本发明构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。因此,本发明专利的保护范围应以所附权利要求为准。

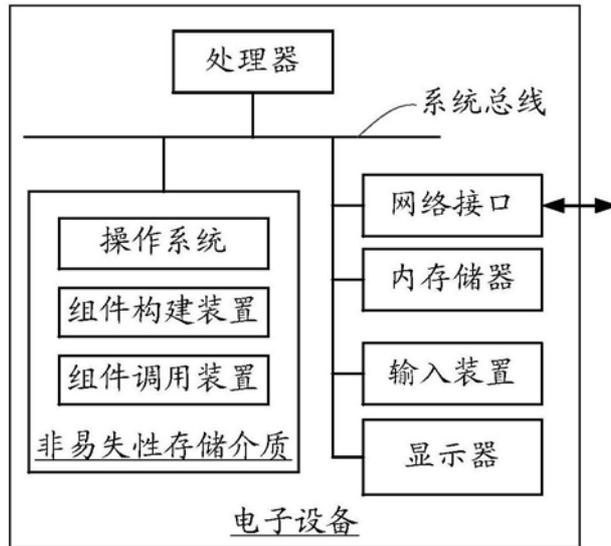


图1

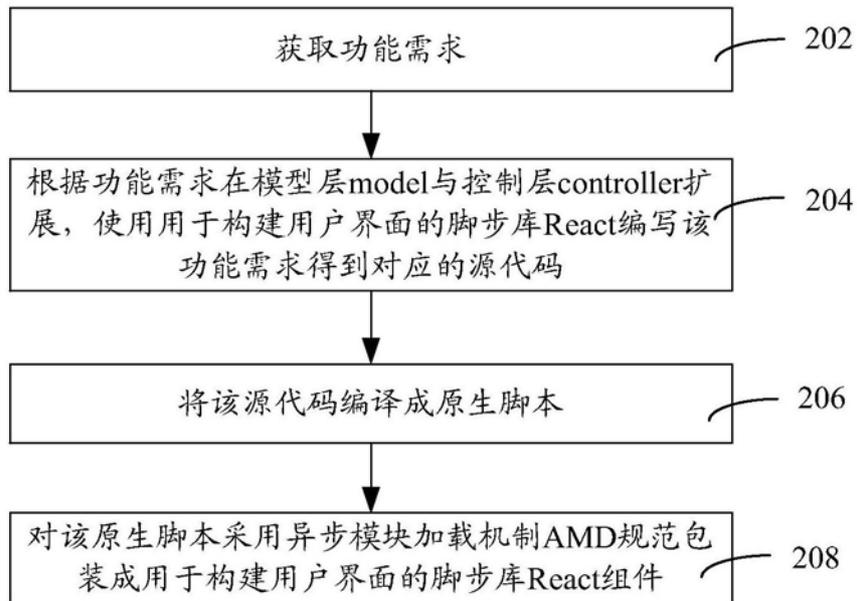


图2

广州	上海	上海金融	北京	香港	多伦多
----	----	------	----	----	-----

图3A

请选择地域 <span style="float: right;">▽</span>
广州
64位6.3
北美

图3B

1	^
	v

图3C

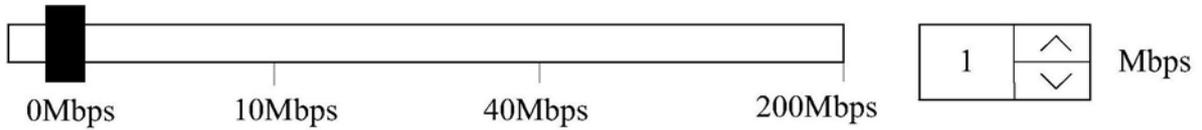


图3D

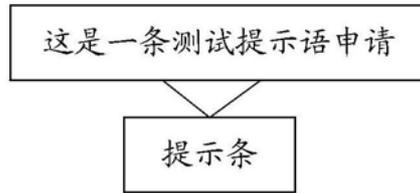


图3E

机型	CPU	内存	是否支持云硬盘	费用
标准型	1核	1G	是	45.00元/月起
标准型	1核	2G	是	85.00元/月起
标准型	1核	4G	是	150.00元/月起
标准型	2核	2G	否	150.00元/月起

图3F

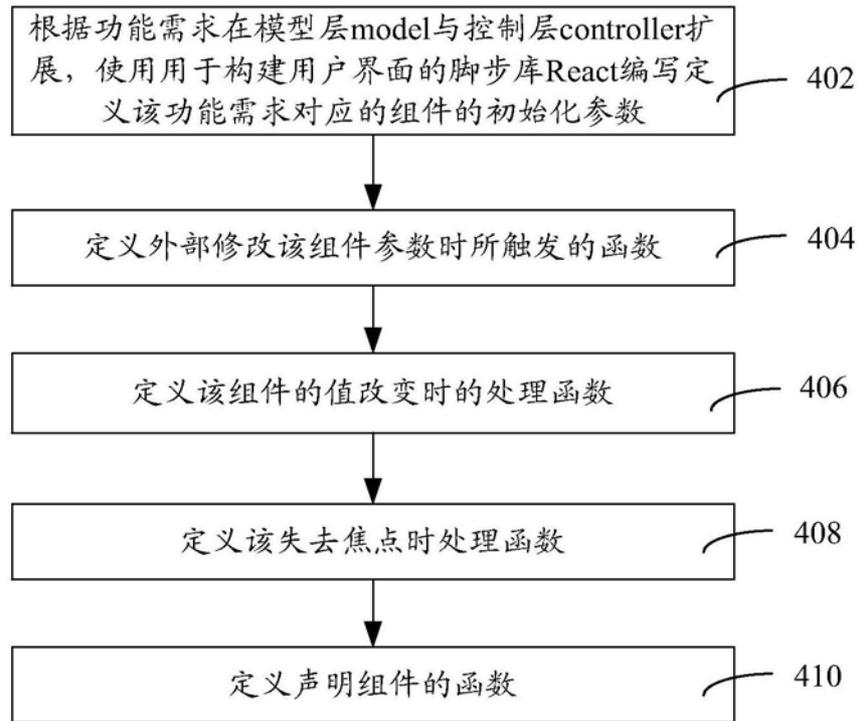


图4

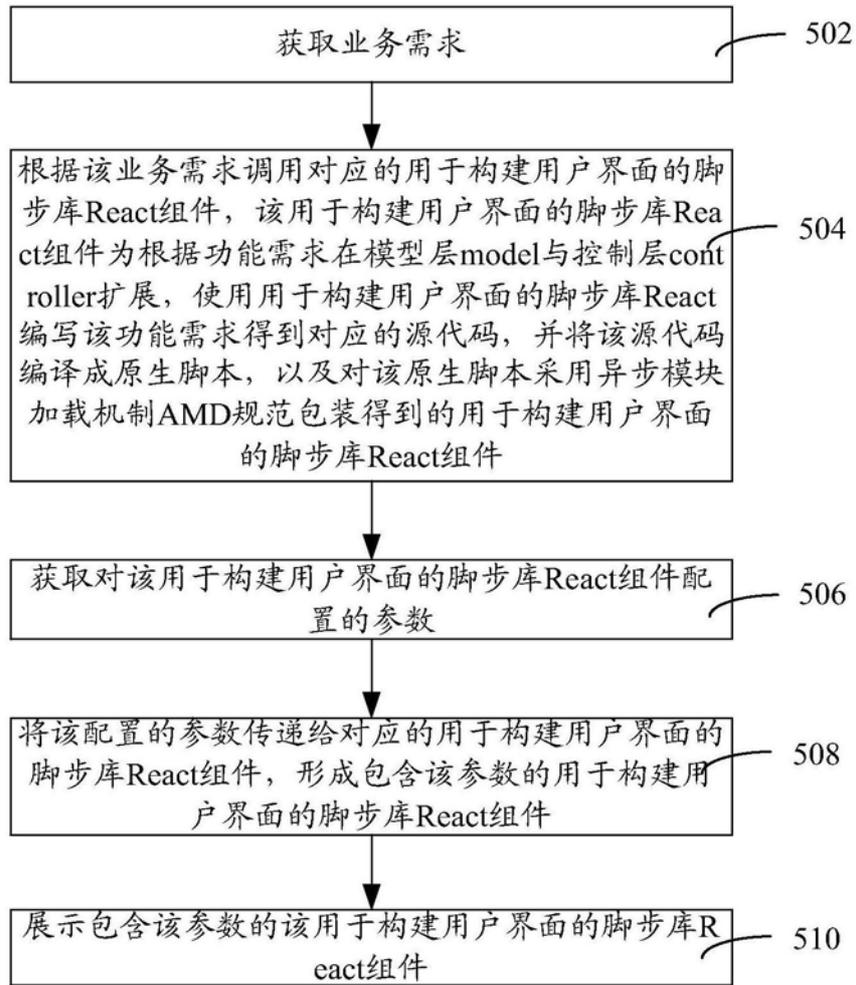


图5

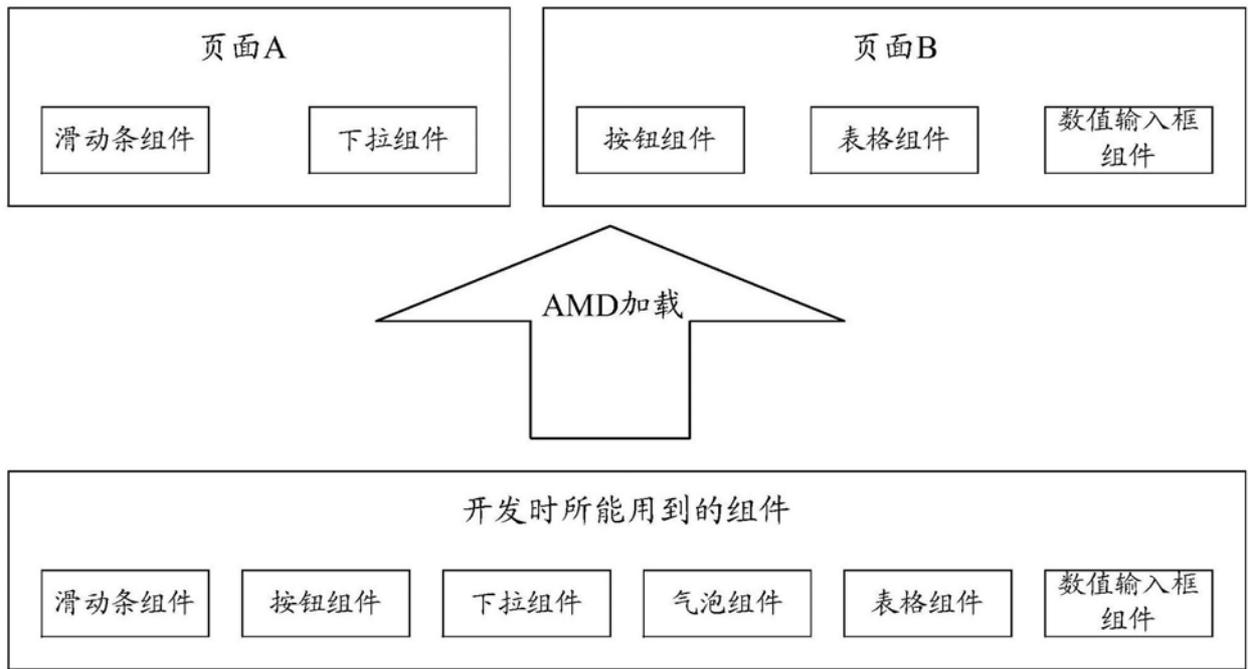


图6

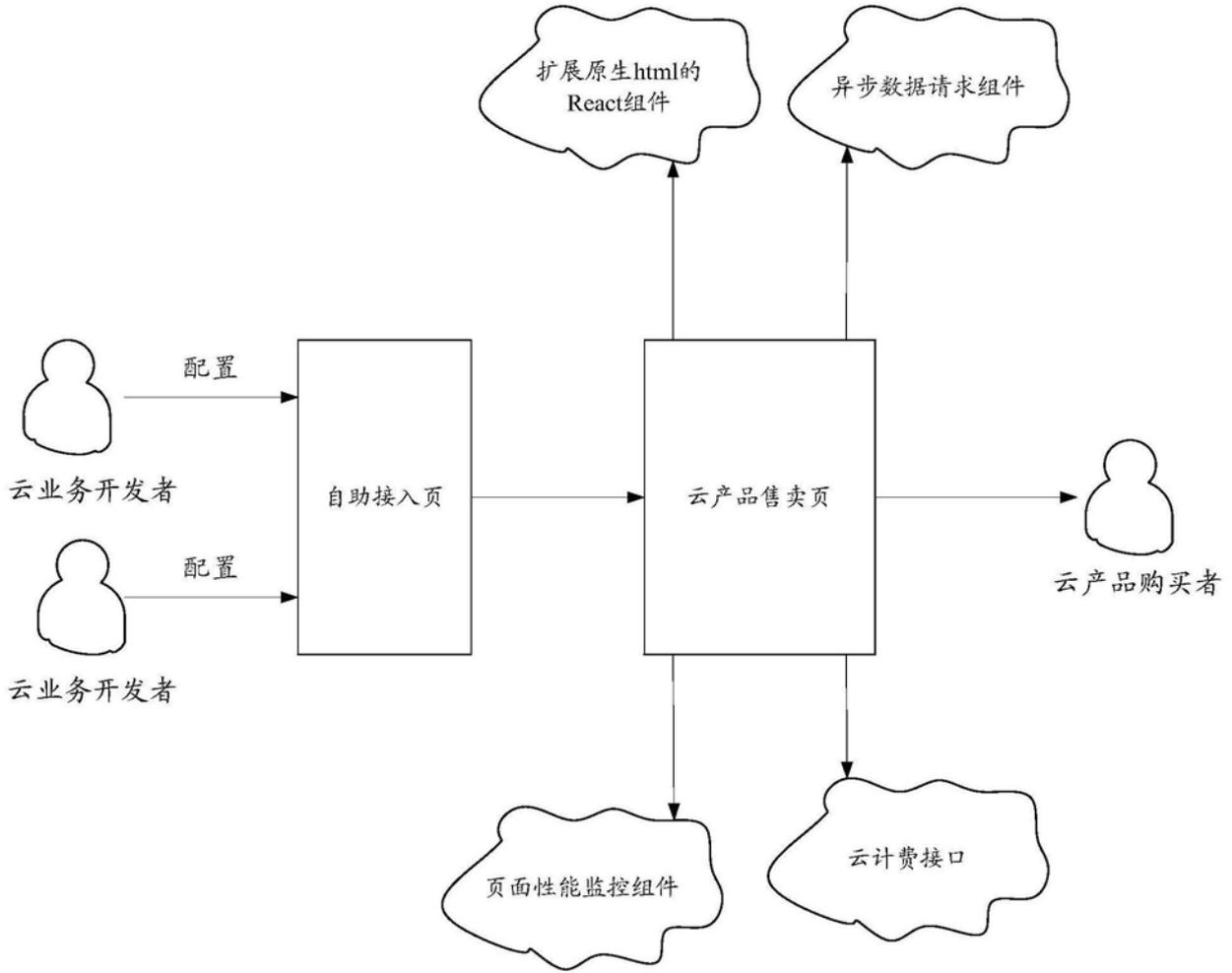


图7

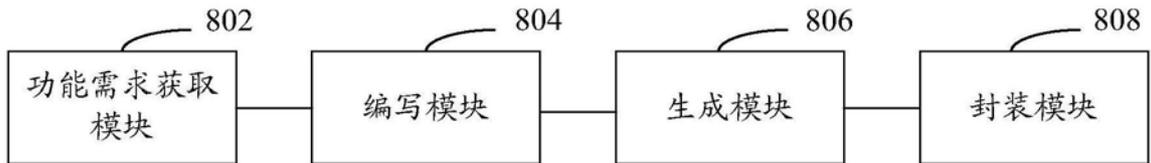


图8

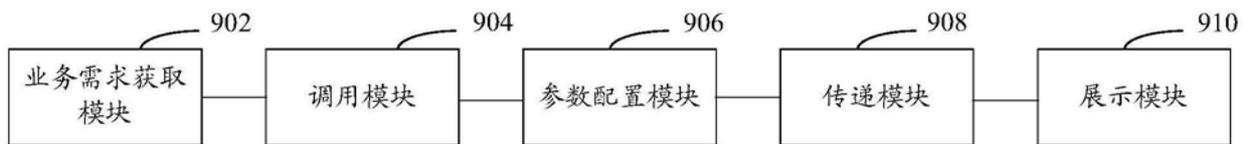


图9