



(12) 发明专利

(10) 授权公告号 CN 118519622 B

(45) 授权公告日 2024.10.11

(21) 申请号 202410980343.2

G06F 8/36 (2018.01)

(22) 申请日 2024.07.22

G06F 8/38 (2018.01)

G06F 8/71 (2018.01)

(65) 同一申请的已公布的文献号

申请公布号 CN 118519622 A

(43) 申请公布日 2024.08.20

(73) 专利权人 珠海金智维信息科技有限公司

地址 519000 广东省珠海市香洲区福田路

18号1栋1层103-043室

(72) 发明人 廖万里 金卓 魏小彬 林达南

郑文锋 陈鹏

(74) 专利代理机构 广州君策达知识产权代理事

务所(普通合伙) 44749

专利代理师 陈宏升

(51) Int. Cl.

G06F 8/20 (2018.01)

(56) 对比文件

CN 113094037 A, 2021.07.09

CN 116775013 A, 2023.09.19

审查员 舒婷

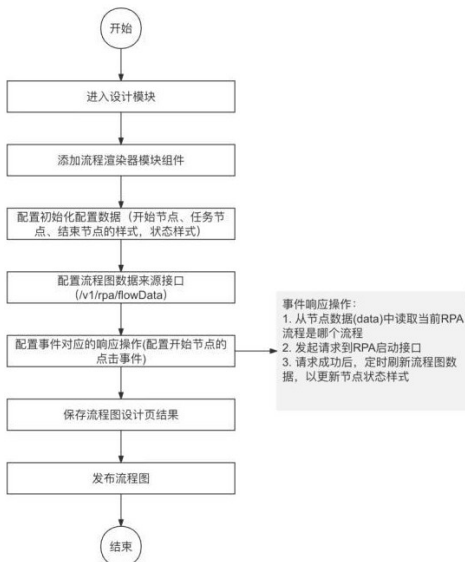
权利要求书2页 说明书10页 附图6页

(54) 发明名称

低代码化的流程图动态配置及自定义交互的系统

(57) 摘要

本发明公开了低代码化的流程图动态配置及自定义交互的系统,包括低代码页面渲染模块、数据结构定义模块、流程图渲染模块和低代码页面设计器模块;其中,低代码页面渲染模块,用于渲染低代码页面;数据结构定义模块,用于定义流程图所需的数据结构;流程图渲染模块,用于初始化流程的图配置以及渲染流程图的数据的可视化模块;低代码页面设计器模块,该模块是一个设计器,用户通过拖拽来编辑页面。本发明统一数据结构、实现模式,有助于降低实现多种类似流程图方案的开发成本;同时以事件化的方式结合低代码页面设计,能够针对不同节点实现不同的交互逻辑;另外以低代码的方式提供配置,开箱即用,减少重复部分的代码开发工作。



1. 低代码化的流程图动态配置及自定义交互的系统,其特征在于,包括低代码页面渲染模块、数据结构定义模块、流程图渲染模块和低代码页面设计器模块;其中,

低代码页面渲染模块,用于渲染低代码页面;

数据结构定义模块,用于定义流程图所需的数据结构;

所述数据结构定义模块,在实现一个RPA流程图的显示,以及在流程图上点击开始节点能启动RPA流程图的时候,按照以下方式操作:

1) 把流程图渲染模块的组件添加入页面中,这时页面的数据结构是页面里包含着一个流程图渲染器的iframe;

2) 配置初始化配置数据:

a. 需要初始化若干种数据类型的节点去与RPA的任务节点类型对应,有3种节点:开始节点、结束节点、任务节点;

b. 给开始节点、结束节点、任务节点分别配置绿、灰、蓝;

c. 定义节点状态列表,分别定义了将要做、正在做、已完成3种状态,并给这3种状态配置对应颜色,分别为蓝、红、绿;

3) 配置流程图数据来源接口:接口返回的数据需要是一个对象数据,对象中存在着2个数组,分别是节点数组和连线数组,且在节点数据中,需要有能与上面配置的状态名称对应的节点状态;

4) 配置启动事件:在低代码页面设计器模块中添加监听节点添加事件,判断如果这个节点是“开始节点”的话,就发起请求启动RPA流程,发起后每隔5秒刷新一次3中配置的“流程图数据来源接口”的数据来刷新流程图渲染模块,并判断如果结束节点为“已结束”状态则停止刷新;

5) “流程图数据来源接口”返回的数据中节点数组有3个数据,分别是开始节点、结束节点、任务节点,连线数组有2条数据,如果返回的节点数据中,从开始到结束的顺序的状态是:已完成、正在做、将要做的话,那么流程图渲染模块上的图将会按蓝、红、绿显示;如果顺序状态是:已完成、已完成、已完成的话,那显示颜色将绿、绿、绿;

流程图渲染模块,用于初始化流程的图配置以及渲染流程图的数据的可视化模块;

在初始化页面时,流程图渲染模块接收流程图数据和初始化参数,流程图渲染模块将它们渲染成流程图,在流程图中点击节点时,会触发节点点击事件,这时会通过窗口消息发送方法把事件响应的数据发送给低代码页面渲染模块,低代码页面渲染模块作为父页面来监听这个事件,当被触发时就会对应处理相关的业务;

低代码页面设计器模块,该模块是一个设计器,用户通过拖拽来编辑页面。

2. 根据权利要求1所述低代码化的流程图动态配置及自定义交互的系统,其特征在于,所述低代码页面渲染模块,工作过程如下:

步骤1、进入页面时,开始加载页面渲染器;

步骤2、加载完成页面渲染器后,通过接口从数据库中读取页面对应的JSON;

步骤3、将页面JSON传给页面渲染器,通过渲染器渲染页面效果;

步骤4、如果页面在设计过程中有绑定事件,渲染出来的页面会监听对应的事件,则会按设计绑定的事件处理函数进行事件处理;如果没有触发事件,则本次页面渲染完成。

3. 根据权利要求1所述低代码化的流程图动态配置及自定义交互的系统,其特征在于,

所述数据结构定义模块,用于定义流程图渲染所需的配置信息,整个配置信息为一个JSON对象,包含了:初始化配置数据对象和流程图实体数据两个主要的配置,所述初始化配置数据对象包含有布局配置、状态列表、节点样式配置。

4.根据权利要求1所述低代码化的流程图动态配置及自定义交互的系统,其特征在于,所述流程图渲染模块,在用户获取到流程图数据和初始化配置信息后,通过低代码页面渲染模块将其以iframe的方式嵌入流程图渲染模块的页面,把流程图数据和初始化配置信息通过窗口消息发送机制与iframe里的流程图渲染模块进行初始化动作、事件交互。

5.根据权利要求1所述低代码化的流程图动态配置及自定义交互的系统,其特征在于,所述流程图渲染模块的监听机制与低代码页面设计器模块的事件进行深入绑定对接,其监听机制如下:事件定义和驱动的交互过程中,流程图渲染模块通过在内部封装定义了事件对外开放,所述事件包括节点点击、保存;当前这些事件被触发时,会通过窗口的消息发送方式通讯,给外面的低代码页面渲染模块发送数据进行交互,而外面的低代码页面渲染模块监听此事件后基于低代码页面设计器模块给这个页面预配置好的事件处理配置去执行对应的动作,所述动作包括请求接口、打开弹窗。

6.根据权利要求1所述低代码化的流程图动态配置及自定义交互的系统,其特征在于,所述低代码页面渲染模块与iframe内所述流程图渲染模块的交互过程如下:

- (1) 进入页面,从服务端获取流程图初始化信息;
- (2) 在得到初始化信息后,开始加载iframe,并通过窗口消息发送方法把初始化数据传给iframe内部;
- (3) 进入iframe内部后,启动流程图渲染器模块开始渲染;
- (4) 渲染完流程图后,将事件开放给iframe内部绑定,通过窗口消息发送方法把事件开放给父页面的低代码页面渲染模块;
- (5) 父页面的低代码页面渲染模块监听iframe内开放的事件,等待用户操作触发事件;
- (6) 用户操作后,触发了事件,立即将事件传递到iframe内部的流程图渲染模块;
- (7) 流程图渲染模块响应到事件操作后,把对应事件和数据向iframe发送;
- (8) iframe层再把事件和数据传给低代码页面渲染模块,由页面渲染分配事件对应的业务处理功能去响应给用户。

7.根据权利要求1所述低代码化的流程图动态配置及自定义交互的系统,其特征在于,所述低代码页面设计器模块的工作过程如下:

- S1、在页面信息表中创建一个新的页面,初始到低代码页面设计器模块中;
- S2、在低代码页面设计器模块中,从组件区拖一个iframe组件,放到效果区;
- S3、设置iframe的属性为流程图渲染模块所在的页面;
- S4、在iframe的事件面板添加对流程图开放出来的事件的监听,并选择处理方式;
- S5、设计完成后,调用页面保存接收,把当前页面保存到页面信息表。

低代码化的流程图动态配置及自定义交互的系统

技术领域

[0001] 本发明涉及可视化建模领域,特别涉及低代码化的流程图动态配置及自定义交互的系统。

背景技术

[0002] 现代企业日益注重数字化转型,许多信息系统都开始采用业务流程图进行可视化建模。流程图可以清晰展示业务流转步骤和决策节点,有效帮助管理和优化业务处理流程。

[0003] 虽然目前市场已有一些优秀的流程图渲染工作,但对于流程图的应用还处于不同的垂直领域,以至于不同的软件开发商对这个流程图定义的事件、节点扩展方式、结构设计、功能针对性都不一样,在实现不同的有关流程图的整合工作都不尽相同,就有可能引入到不同的流程图实现方式和代码,这里面可能有大部分是逻辑重复的,但由于来自不同的行业或技术领域,难以统一,所以只能分别开发实现,这样的开发模式往往比较浪费成本,且效率也会比较低。原因暴露出了以下几个问题亟待解决:

[0004] (1) 流程图定义方式各异,同一个企业内不同系统的流程难以集成和重用;

[0005] (2) 流程节点交互功能单一,不便于根据具体业务行业或技术领域需要进行个性化定制;

[0006] (3) 流程图的数据格式和结构没有标准,扩展性和兼容性较差;

[0007] (4) 对于流程修改,需要重新开发部署才能实现,效率很低。

[0008] 例如,“202111351513.3,一种基于Antv的自定义可视化拓扑图的实现方法,江苏省未来网络创新研究院”的技术方案是一个只针对拓扑图的实现方法,那当用户在引入了它,实现了一个针对拓扑图的实现页面后,如果此时用户又有新需求,需要实现一个工作流的场景,那该方法将可能是无法实现的,因为该方法只适用于拓扑图的布局模式、连线等和工作流的方式。

发明内容

[0009] 本发明的目的在于克服现有技术的缺点与不足,提供低代码化的流程图动态配置及自定义交互的系统。

[0010] 本发明的目的通过以下的技术方案实现:

[0011] 低代码化的流程图动态配置及自定义交互的系统,包括低代码页面渲染模块、数据结构定义模块、流程图渲染模块和低代码页面设计器模块;其中,

[0012] 低代码页面渲染模块,用于渲染低代码页面;

[0013] 数据结构定义模块,用于定义流程图所需的数据结构;

[0014] 流程图渲染模块,用于初始化流程的图配置以及渲染流程图的数据的可视化模块;

[0015] 低代码页面设计器模块,该模块是一个设计器,用户通过拖拽来编辑页面。

[0016] 所述低代码页面渲染模块,工作过程如下:

- [0017] 步骤1、进入页面时,开始加载页面渲染器;
- [0018] 步骤2、加载完成页面渲染器后,通过接口从数据库中读取页面对应的JSON;
- [0019] 步骤3、将页面JSON传给页面渲染器,通过渲染器渲染页面效果;
- [0020] 步骤4、如果页面在设计过程中有绑定事件,渲染出来的页面会监听对应的事件,则会按设计绑定的事件处理函数进行事件处理;如果没有触发事件,则本次页面渲染完成。
- [0021] 所述数据结构定义模块,用于定义流程图渲染所需的配置信息,整个配置信息为一个JSON对象,包含了:初始化配置数据对象和流程图实体数据两个主要的配置,所述初始化配置数据对象包含有布局配置、状态列表、节点样式配置。
- [0022] 所述数据结构定义模块,在实现一个RPA流程图的显示,以及在流程图上点击开始节点能启动RPA流程图的时候,按照以下方式操作:
- [0023] 1) 把流程图渲染模块的组件添加加入页面中,这时页面的数据结构是页面里包含着一个流程图渲染器的iframe;
- [0024] 2) 配置初始化配置数据:
- [0025] a. 需要初始化若干种数据类型的节点去与RPA的任务节点类型对应,假设有3种节点:开始节点、结束节点、任务节点;
- [0026] b. 给开始节点、结束节点、任务节点分别配置绿、灰、蓝;
- [0027] c. 定义节点状态列表,分别定义了将要做、正在做、已完成3种状态,并给这3种状态配置对应颜色,分别为蓝、红、绿;
- [0028] 3) 配置流程图数据来源接口:接口返回的数据需要是一个对象数据,对象中存在于2个数组,分别是节点数组和连线数组,且在节点数据中,需要有能与上面配置的状态名称对应的节点状态;
- [0029] 4) 配置启动事件:在低代码页面设计器模块中添加监听节点添加事件,判断如果这个节点是“开始节点”的话,就发起请求启动RPA流程,发起后每隔5秒刷新一次3中配置的“流程图数据来源接口”的数据来刷新流程图渲染模块,并判断如果结束节点为“已结束”状态则停止刷新;
- [0030] 5) “流程图数据来源接口”返回的数据中节点数组有3个数据,分别是开始节点、结束节点、任务节点,连线数组有2条数据,如果返回的节点数据中,从开始到结束的的顺序的状态是:已完成、正在做、将要做的,那么流程图渲染模块上的图将会按蓝、红、绿显示;如果顺序状态是:已完成、已完成、已完成的话,那显示颜色将绿、绿、绿。
- [0031] 所述流程图渲染模块,在用户获取到流程图数据和初始化配置信息后,通过低代码页面渲染模块将其以iframe的方式嵌入流程图渲染模块的页面,把流程图数据和初始化配置信息通过窗口消息发送机制与iframe里的流程图渲染模块进行初始化动作、事件交互。
- [0032] 在初始化页面时,流程图渲染模块接收流程图数据和初始化参数,流程图渲染模块将它们渲染成流程图,在流程图中点击节点时,会触发节点点击事件,这时会通过窗口消息发送方法把事件响应的数据发送给低代码页面渲染模块,低代码页面渲染模块作为父页面来监听这个事件,当被触发时就会对应处理相关的业务。
- [0033] 所述流程图渲染模块的监听机制与低代码页面设计器模块的事件进行深入绑定对接,其监听机制如下:事件定义和驱动的交互过程中,流程图渲染模块通过在内部封装定

义了事件对外开放,所述事件包括节点点击、保存;当前这些事件被触发时,会通过窗口的消息发送方式通讯,给外面的低代码页面渲染模块发送数据进行交互,而外面的低代码页面渲染模块监听此事件后基于低代码页面设计器模块给这个页面预配置好的事件处理配置去执行对应的动作,所述动作包括请求接口、打开弹窗。

[0034] 所述低代码页面渲染模块与iframe内所述流程图渲染模块的交互过程如下:

[0035] (1) 进入页面,从服务端获取流程图初始化信息;

[0036] (2) 在得到初始化信息后,开始加载iframe,并通过窗口消息发送方法把初始化数据传给iframe内部;

[0037] (3) 进入iframe内部后,启动流程图渲染器模块开始渲染;

[0038] (4) 渲染完流程图后,将事件开放给iframe内部绑定,通过窗口消息发送方法把事件开放给父页面的低代码页面渲染模块;

[0039] (5) 父页面的低代码页面渲染模块监听iframe内开放的事件,等待用户操作触发事件;

[0040] (6) 用户操作后,触发了事件,立即将事件传递到iframe内部的流程图渲染模块;

[0041] (7) 流程图渲染模块响应到事件操作后,把对应事件和数据向iframe发送;

[0042] (8) iframe层再把事件和数据传给低代码页面渲染模块,由页面渲染分配事件对应的业务处理功能去响应给用户。

[0043] 所述低代码页面设计器模块的工作过程如下:

[0044] S1、在页面信息表中创建一个新的页面,初始到低代码页面设计器模块中;

[0045] S2、在低代码页面设计器模块中,从组件区拖一个iframe组件,放到效果区;

[0046] S3、设置iframe的属性为流程图渲染模块所在的页面;

[0047] S4、在iframe的事件面板添加对流程图开放出来的事件的监听,并选择处理方式;

[0048] S5、设计完成后,可以调用页面保存接收,把当前页面保存到页面信息表。

[0049] 本发明与现有技术相比,具有如下优点和有益效果:

[0050] 1、本发明统一数据结构、实现模式,有助于降低实现多种类似流程图方案的开发成本。

[0051] 2、本发明以事件化的方式结合低代码页面设计,能够针对不同节点实现不同的交互逻辑。

[0052] 3、本发明以低代码的方式提供配置,开箱即用,减少重复部分的代码开发工作。

[0053] 4、本发明以数据化的方式存储流程图结构、配置、事件、页面配置,无须再次打包部署,即可实现不同业务定制功能。

附图说明

[0054] 图1为本发明所述低代码化的流程图动态配置及自定义交互的系统的设计态业务流程图。

[0055] 图2为本发明所述低代码化的流程图动态配置及自定义交互的系统的执行态业务流程图。

[0056] 图3为低代码页面渲染模块在运行态中的工作流程图。

[0057] 图4为数据结构定义模块的工作流程图。

- [0058] 图5为低代码页面渲染模块与iframe内所述流程图渲染模块的业务逻辑图。
- [0059] 图6为低代码页面设计器模块的页面设计效果图。
- [0060] 图7为页面信息表结构图。

具体实施方式

[0061] 下面结合实施例及附图对本发明作进一步详细的描述,但本发明的实施方式不限于此。

[0062] 如图1至6,低代码化的流程图动态配置及自定义交互的系统,是一种基于数据驱动和低代码技术,实现流程图动态配置和自定义交互的解决方案。它采用标准的JSON数据格式定义流程公共元素,如节点、连线及其属性,这使得流程图定义不依赖于特定系统,具有良好的扩展性和重复利用性。同时,方案通过低代码页面设计器支持快速构建各类流程图交互行为的绑定,如绑定节点事件打开自定义属性表单。此外,它采用模块化设计,将流程定义与渲染划分开来,通过事件驱动机制实现了模块间松耦合的交互。该方案整体采用数据驱动的思路,将流程相关元素如定义、表单等信息统一存储于数据库,从而实现了动态配置的功能

[0063] 在该方案中,是通过四个模块来互相配合实现的,它们分别是:

[0064] 模块一:低代码页面渲染模块,用于渲染低代码页面;

[0065] 模块二:数据结构定义模块,用于定义流程图所需的数据结构;

[0066] 模块三:流程图渲染模块,用于初始化流程的图配置以及渲染流程图的数据的可视化模块;

[0067] 模块四:低代码页面设计器模块,该模块是一个设计器,用户可以通过拖拽来编辑页面;

[0068] 通过以上四个模块,我们分别完成了设计态和执行态的相应功能来实现本发明中的核心功能。

[0069] 在设计态中,我们通过低代码页面设计器模块配置页面布局、引用组件、数据配置(按数据结构标准定义配置)、接口配置、事件操作配置等操作完成一个流程图页面的设计,并将它发布成低代码页面,使其能在代码页面渲染模块中渲染使用,它的业务流程如图1。

[0070] 在执行态中,我们通过低代码页面渲染模块在加载已发布的流程图页面的过程中,分别读取配置、流程图数据、事件等,并监听相关事件,在它上面实现流程图的渲染、操作等功能,它的业务流程如图2。

[0071] 在本发明中,我们使用了四个模块了配合实现整体功能,而在这四个模块中,我们定义了统一的数据建模模块和事件驱动机制也是关键所在,其中我们通过统一的数据结构定义,实现了流程定义与运营的解耦,支撑流程动态化能力:

[0072] 1、通过在数据结构定义模块(模块二)使用JSON数据格式定义流程公共元素,如节点、连线等结构元素。

[0073] 2、在数据结构定义模块(模块二)将流程定义抽象为配置信息和实体数据两部分,分别被用于存放与调用。

[0074] (1)配置信息定义了样式、状态列表、布局方法的相关配置信息。

[0075] (2)数据模块定义了节点和连线的数据模型。

[0076] 3、通过低代码页面设计器模块(模块四),设计存储一个基于低代码页面渲染器(模块一)可执行的页面,以一个iframe组件的方式嵌入流程图渲染器模块(模块三),并配置事件处理(详见下文“页面渲染器与iframe内流程图渲染器交互过程”)。

[0077] 4、访问页面时,低代码页面渲染器(模块一)读取低代码页面设计器模块所设计存储的页面把页面渲染出来,获取交互行为、事件配置,对页面中配置的事件进行监听

[0078] 5、流程图渲染模块基于读取配置和实体数据进行动态渲染。

[0079] 6、在此之中将所有流程相关定义统一作为数据进行管理调用。

[0080] 7、支持流程修改自动生效,无需重启,实现热更新能力。

[0081] 上述所提到的交互方式,是通过事件定义和驱动,实现了模块间解耦而松耦合的交互通信机制,各模块交互方式:

[0082] 1、需要在流程图渲染模块中实现并对外开放标准事件,事件列表见下文“流程图渲染模块封装的事件”表中。

[0083] 2、低代码渲染器模块通过获取低代码设计器为不同事件绑定的事件或交互动作去处理逻辑。

[0084] 3、两个模块通过自由绑定式事件驱动进行松耦合交互。

[0085] 每个模块的关键逻辑描述如下:

[0086] 一、模块一,低代码页面渲染模块:该模块利用现在的成熟低代码页面渲染器来实现,它是以JSONSchema标准的JSON(JavaScript Object Notation, JS对象简谱,是一种轻量级的数据交换格式)数据方式来描述页面元素、布局、事件处理等信息,共有两种使用状态:设计态、运行态,在设计态中,可以使用页面设计器(editor)通过拖拽的方式布局页面元素,通过属性面板绑定属性值、事件处理函数,在editor中的所有操作最终都会转换为一个JSONSchema标准的JSON,我们可以将这些统一的数据结构进行存储,这个编辑、存储过程将在模块四中介绍;在运行态中,该渲染器会把设计态产生的JSON转换渲染成标准Web HTML页面,渲染出来的,如图3,实现步骤如下:

[0087] 步骤1:当我们进入页面时,开始加载页面渲染器;

[0088] 步骤2:加载完成页面渲染器后,通过接口从数据库中读取页面对应的JSON;

[0089] 步骤3:将页面JSON传给页面渲染器,通过渲染器渲染页面效果;

[0090] 步骤4:如果页面在设计过程中有绑定事件,渲染出来的页面会监听对应的事件,则会按设计绑定的事件处理函数进行事件处理;如果没有触发事件,则本次页面渲染完成。

[0091] 二、模块二,数据结构定义模块:该模块主要是定义流程图渲染所需的配置信息,整个配置信息为一个JSON对象,包含了:初始化配置数据对象和流程图实体数据两个主要的配置,初始化配置数据对象包含有布局配置、状态列表、节点样式配置。

[0092] 通过这个定义这样的一套数据结构,我们实现可以灵活的扩展节点的样式(图标、颜色、样式)以及自定义用户自己的数据属性可以在节点数据对象中任意扩展。

[0093] 比如:当用户在系统中需要实现一个拓扑图,以及一个工作流的流程图时,如果使用传统的开发方式,这时可能将会分别引入两个不同的对应专业的图工具,分别在不同的页面开发对应的功能去实现,这时由于两个不同的专业的图工具是由不同的人开发的,那数据结构、名称等很可能会有差异,那么开发者就可以对它们进行分别存储才可以,这样如果后面有需求调整,那维护成本可能会比较高。但如果使用本发明,由于本发明是在数据结

构定义模块中定义开放了布局、样式等配置,用户可以在设计阶段就能自由选择当前想要实现的流程图是什么样的布局形式和样式等,只要用户实现的接口数据结构是符合本发明的数据结构要求就可以根据配置渲染,所以完全可以统一进行管理。

[0094] 另外,再比如用户如果想要实现的是一个每个节点的属性是自定义的,且属性编辑页可自行调整的需求,那么使用传统开发方式,哪怕属性上他们自己去设计成动态数据表的方式,但属性编辑页这种一般都需要提前开发,给用户自行调整的空间不大。而由于是使用低代码的渲染器来结合的,也就是说我们完全可以充分发挥低代码设计的优势,当用户需要自己设计属性页时,我们可以开放出设计态给用户去设计这个属性窗口,再将它以属性页的方式与节点绑定,然后配置节点的触发事件来打开这个页面进行渲染,从而达到把选择权交给用户的目的。

[0095] 数据结构如表1、表2、表3。

[0096] 表1

简称	名称	父级简称	类型	描述
config	初始化配置	无	Object	初始化流程图的根对象数据,后面的配置都是它的子或子孙数据
nodeConfig	节点样式配置	config	Array	预定义将在流程图中显示的节点类型、样式、图标,同时也是左侧工具库内的元素,用于被使用端的用户拖出添加到流程图中
id	节点主键	nodeConfig	String	是节点的唯一标识,在定义阶段可以为空,设计时会自动生成,以便在运行时中使用放到 data 属性中使用
type	节点类型	nodeConfig	String	同一类型节点标识,当流程图需要重复渲染多个节点时,这个属性用来识别哪个节点是同类,以便处理相同逻辑
data	节点数据	nodeConfig	Object	存在节点业务数据,数据属性从定义的属性中继承再重新赋值,定义阶段允许为空,如果不为空则代码是默认值,且设计者可以在这里添加自己业务的自定义数据属性。
style	节点样式	nodeConfig	Object	节点样式,可以按 css 样式标准定义样式,会在渲染时按定义时的类型分别渲染样式
shape	节点	nodeConfig	String	功能节点类型,是一个可从代码端扩展的类型,通过它可以定义一些有特殊功能的默认节点,来被用户的自定义节点去继承
label	节点名称	nodeConfig	String	节点名称
icon	节点图标	nodeConfig	String	节点图标,使用符合 iconfont 标准的 class 定义方式来渲染图标
children	子节点	nodeConfig	Array	数据结构和当前父对象相似,同样定义 id, data, style, type, type, label, icon, children 等属性

[0098] 表2

[0099]

简称	名称	父级简称	类型	描述
statusList	节点状态预定义列表	config	Object	节点状态预定义列表，预定义好不同状态的样式，当需要显示状态时，会通过节点数据中的状态标识到这个列表中寻找对应样式显示
flowData	流程图数据	config	Object	流程图数据：用于在运行态按 initData 的预定义节点样式配置来渲染流程，本对象主要存 nodes, edges 两个属性，分别用于存在节点数据和连线关系数据
nodes	流程图节点集合	flowData	Array	节点数据，允许为空数组(代表流程图是空的)，数组内存储 0-N 个节点数据对象，用于渲染流程图后的节点显示和后续的交互数据使用
id	节点 ID	nodes	String	节点 ID
data	节点数据对象	nodes	Object	节点数据对象，存默认数据或自定义数据
id	节点 ID	data	String	节点 ID 同上
label	节点名称	data	String	节点名称
style	节点样式	data	Object	在内部填写对应的 CSS 样式
icon	节点图标	data	String	要显示的图标
flowStatus	节点状态标识	data	String	到对应节点状态列表中寻找对应样式显示的状态标识

[0100] 表3

[0101]

简称	名称	父级简称	类型	描述
position	节点位置	nodes	Object	节点位置对象，里面存储 x,y 轴坐标信息
x	x 轴坐标	position	Number	节点的 x 轴坐标
y	y 轴坐标	position	Number	节点的 y 轴坐标
edges	连线数据集合	flowData	Array	连线数据集合
id	连线 ID	edges	String	连线 ID
data	连线数据对象	edges	Object	连线数据对象，存默认数据或自定义数据
id	连线 ID	data	String	连线 ID 同上
label	连线上的名称	data	String	连线上的名称
source	连线的出发节点 ID	edges	String	连线两端的开始位置对应的节点，值为对应的节点 ID
target	连线的到达节点 ID	edges	String	连线两端的结束位置对应的节点，值为对应的节点 ID

[0102] 由表1、表2、表3可见，我们的数据核心主要在于初始化配置和流程数据上，初始化配置重点在于可以预先定义节点的样式、配置，流程数据上采用和行业内一样的数据结构，只是在那基础上添加了自定义的data属性和样式属性等，既保证了大家的开发习惯，又扩展了一些活动配置的功能。

[0103] 示例:比如我们想实现一个RPA流程图的显示,以及在流程图上点击开始节点能启动RPA流程图,如图4,那么我们在设计时应该如下方式操作:

[0104] 1) 进入设计模块时后:我们把流程渲染器组件添加入页面中,这时页面的数据结构是页面里包含着一个流程图渲染器的iframe

[0105] 2) 配置初始化配置数据:

[0106] a.需要初始化几种数据类型的节点去与RPA的任务节点类型对应,这里假设有3种分别是开始、结束、任务节点;

[0107] b.给以上3种节点都配置一个不同的颜色分别是绿、灰、蓝;

[0108] c.定义节点状态列表,分别定义了将要做、正在做、已完成3种状态,并给这3种状态配置对应颜色分别为蓝、红、绿;

[0109] 3) 配置流程图数据来源接口:接口返回的数据需要是一个对象数据,对象中存在着2个数组,分别是节点数组和连线数组,且在节点数据中,需要有能与上面配置的状态名称对应的节点状态;

[0110] 4) 配置启动事件:在页面设计器中添加监听节点添加事件,判断如果这个节点是“开始节点”的话,就发起请求启动RPA流程,发起后每隔5秒刷新一次3中配置的“流程图数据来源接口”的数据来刷新流程图渲染器,并判断如果结束节点为“已结束”状态则停止刷新;

[0111] 5) “流程图数据来源接口”返回的数据中节点数组有3个数据,分别是开始、结束、任务节点,连线数组有2条数据,如果返回的节点数据中,从开始到结束的顺序的状态是:已完成、正在做、将要做的話,那么流程图渲染器上的图将会按蓝、红、绿显示;如果顺序状态是:已完成、已完成、已完成的话,那显示颜色将绿、绿、绿。

[0112] 三、模块三,流程图渲染模块:当用户获取到流程图数据和初始化配置信息后,通过页面渲染器将以iframe(HTML浮动框架)的方式嵌入流程图渲染模块的页面,把流程图数据和初始化配置信息通过窗口消息发送机制与iframe里的流程图渲染模块进行初始化动作、事件交互。例如:初始化页面时,会把流程图数据和初始化参数传给流程图渲染器,流程把它们渲染成流程图,但在流程图中点击节点时,会触发节点点击事件,这时会通过窗口消息发送方法把事件响应的数据发送给页面渲染器这个父页面,父页面监听这个事件,当被触发时就会对应处理相关的业务。

[0113] 在本模块中,我们通过流程图渲染器开放了流程图中的常用交互事如点击、选择等事件,与传统实现不同的是,这些事件的监听处理机制上,我们将它和低代码页面设计器的事件进行深入绑定对接,从而使我们可以直接在低代码页面中直接接入它的事件,并对它进行处理,从而使用开发者在做交互时,可以完全使用低代码方式去接入,不需要编写任务代码,只需要确保数据结构是按我们定义的标准来定义即可。

[0114] 1) 监听机制

[0115] 事件定义和驱动的交互过程中,流程图渲染模块通过在内部封装定义了节点点击、保存等事件对外开放,当前这些事件被触发时,会通过窗口的消息发送方式通讯,给外面的页面渲染器模块发送数据进行交互,而外面的页面渲染模块监听此事件后基于设计器模块给这个页面预配置好的事件处理配置去执行对应的动作(如请求接口、打开弹窗等。

[0116] 2) 页面渲染器与iframe内流程图渲染器交互过程,如图5,展示了低代码页面渲染

模块与iframe内所述流程图渲染模块之间如何渲染及提供、监听、触发、处理对应事件的交互过程。

[0117] 逻辑说明:

[0118] 步骤1:进入页面,从服务端获取流程图初始化信息;

[0119] 步骤2:在得到初始化信息后,开始加载iframe,并通过窗口消息发送方法把初始化数据传给iframe内部;

[0120] 步骤3:进入iframe内部后,启动流程图渲染器开始渲染;

[0121] 步骤4:渲染完流程图后,将事件开放给iframe内部绑定,通过窗口消息发送方法把事件开放给父页面的页面渲染器;

[0122] 步骤5:页面渲染器监听iframe内开放的事件,等待用户操作触发事件;

[0123] 步骤6:用户操作后,触发了事件,立即将事件传递到iframe内部的流程图渲染器;

[0124] 步骤7:流程图渲染器响应到事件操作后,把对应事件和数据向iframe发送;

[0125] 步骤8:iframe层再把事件和数据传给页面渲染器,由页面渲染分配事件对应的业务处理功能去响应给用户。

[0126] 3) 流程图渲染模块封装的事件如表4。

[0127] 表4

事件名称	描述
saveGraph	流程图数据更新事件
saveSelection	流程图中发生选择事件 (选择节点或连线)
nodeClick	节点鼠标左键单击事件
nodeDbClick	节点鼠标左键双击事件
edgeClick	连线鼠标左键单击事件
edgeDbClick	连线鼠标左键双击事件
blankClick	画布空白区域鼠标左键单击事件
blackDbClick	画布空白区域鼠标左键双击事件

[0128] 四、模块四,低代码页面设计器模块,我们可以在上面设计页面布局、表单、接口对接和事件对接效果如图6,并且为确保低代码页面是可以实时修改、读取、渲染的,需要将它们持久化存储到服务端,存储表结构(关键字段)如图7。

[0130] 1) 页面设计关键逻辑说明:

[0131] 步骤1:在页面信息表中创建一个新的页面,初始到页面设计器中(效果为空白);

[0132] 步骤2:在设计器中,从组件区拖一个iframe组件,放到效果区;

[0133] 步骤3:设置iframe的属性为流程图渲染器所在的页面;

[0134] 步骤4:在iframe的事件面板添加对流程图开放出来的事件的监听,并选择处理方式,处理方式有如下:

[0135] 跳转链接;

- [0136] 刷新页面；
- [0137] 回退页面；
- [0138] 打开/关闭弹窗(如果是弹窗,则可以对弹窗内容进行2次设计,编辑方式和本页面设计器的方式一样,可以嵌套设计)；
- [0139] 弹出消息提醒；
- [0140] 发送HTTP请求；
- [0141] 下载文件；
- [0142] 设置某个其他组件的可见性、可用性；
- [0143] 向某个组件传参、变量赋值；
- [0144] 提交、清空、重围、校验表单；
- [0145] 灵活扩展JavaScript脚本处理；
- [0146] 步骤5:设计完成后,可以调用页面保存接收,把当前页面保存到dev_page(页面信息表)。
- [0147] 本发明的关键技术点如下：
- [0148] 1、基于JSON的数据模型,定义标准的流程元素体系。这实现了流程的可配置化和复用。
- [0149] 2、使用事件驱动机制实现跨模块的流程交互。流程图渲染模块与低代码页面通过事件进行松耦合的交互,使得两个模块可以独立迭代升级。
- [0150] 3、基于低代码页面设计器进行流程交互绑定。设计器支持为流程图事件绑定不同处理逻辑,实现了流程交互行为的快速配置。
- [0151] 4、将流程相关元素采用数据驱动方式存储。将流程图定义、页面定义等信息存储在数据库中,实现了动态配置的能力。
- [0152] 上述实施例为本发明较佳的实施方式,但本发明的实施方式并不受上述实施例的限制,其他的任何未背离本发明的精神实质与原理下所作的改变、修饰、替代、组合、简化,均应为等效的置换方式,都包含在本发明的保护范围之内。



图 1

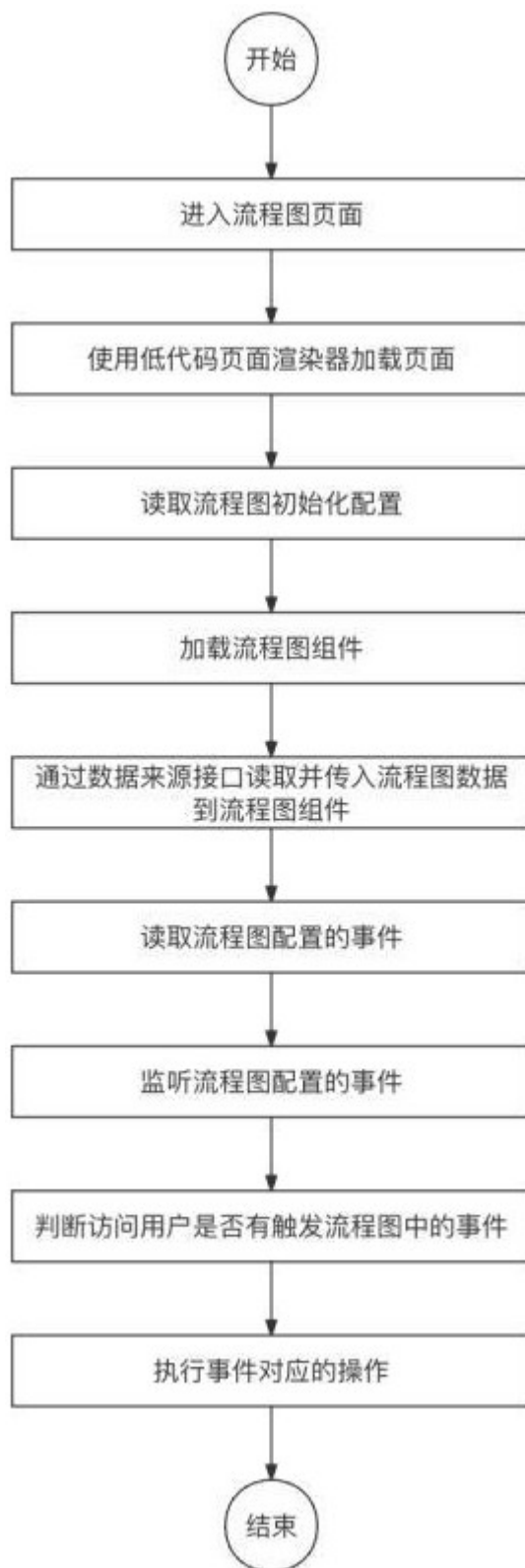


图 2

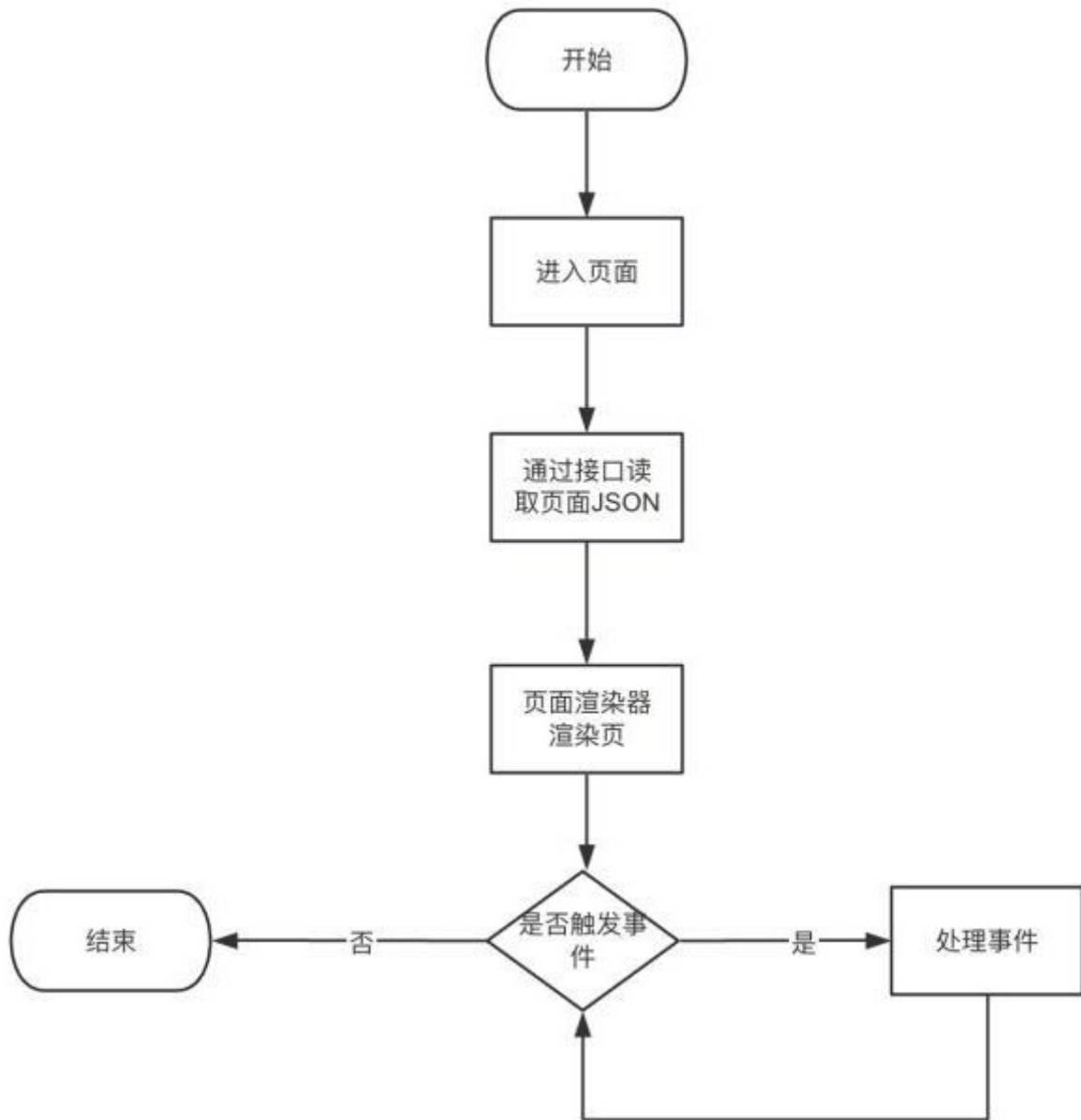


图 3

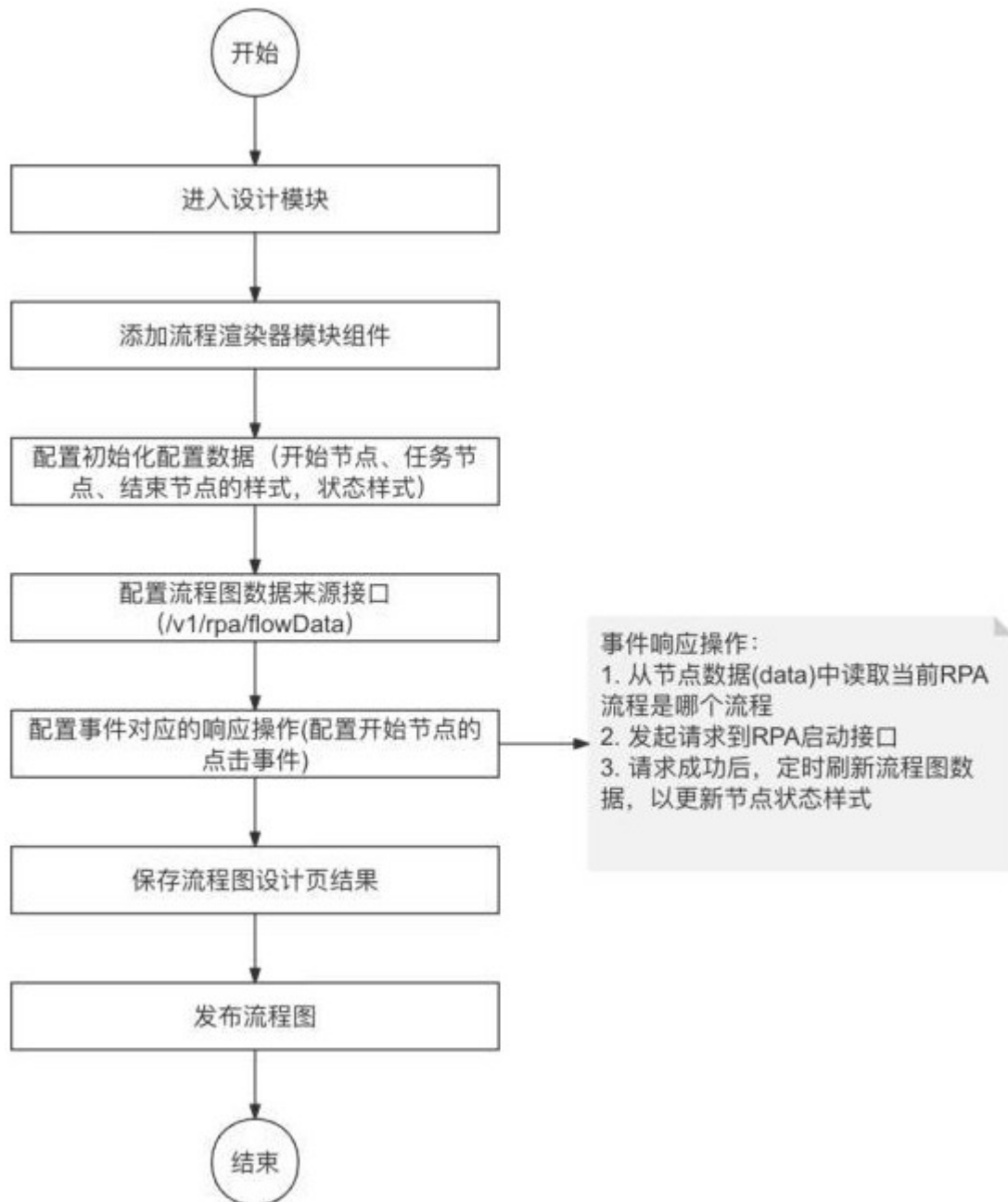


图 4

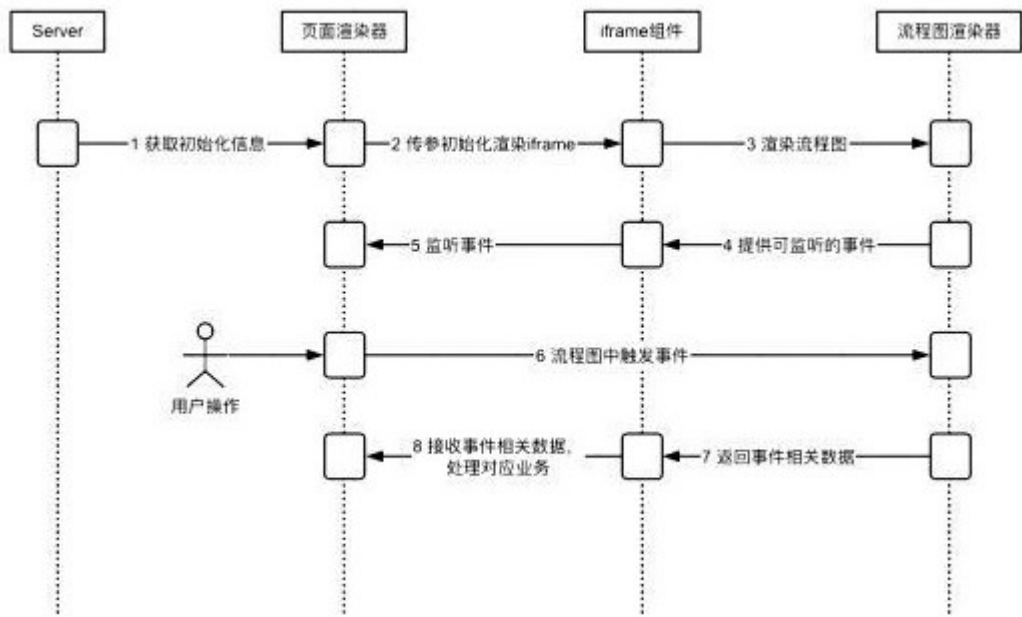


图 5

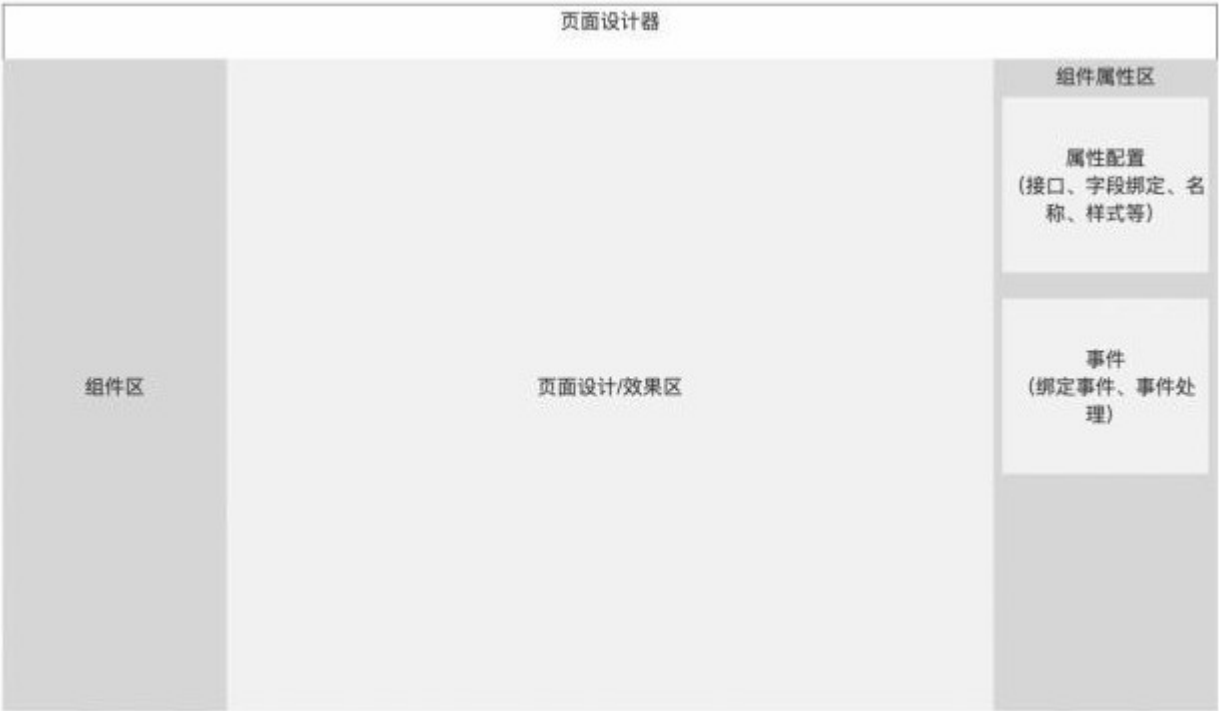


图 6

dev_page
- id varchar(32) - page_json longtext - path varchar(255) ...

图 7