



(12) 发明专利申请

(10) 申请公布号 CN 112181416 A

(43) 申请公布日 2021.01.05

(21) 申请号 202011084484.4

(22) 申请日 2020.10.12

(71) 申请人 上海赛可出行科技服务有限公司
地址 200131 上海市浦东新区自由贸易试
验区杨高北路2001号1幢4部位三层
333室

(72) 发明人 赵化 金小俊 刘亚刚

(51) Int.Cl.
G06F 8/38 (2018.01)
G06F 8/34 (2018.01)
G06F 8/36 (2018.01)
G06K 9/00 (2006.01)

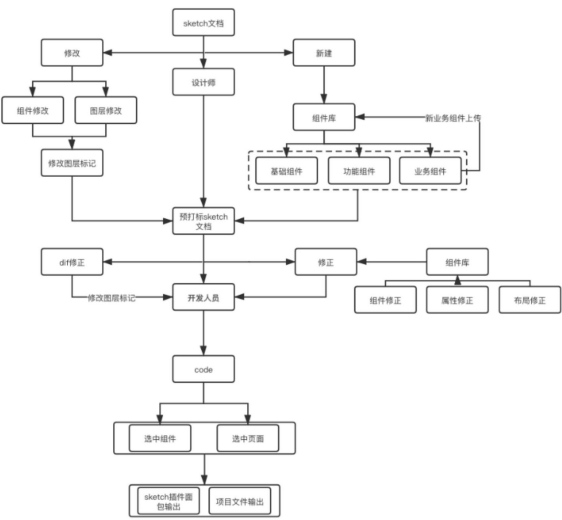
权利要求书3页 说明书8页 附图5页

(54) 发明名称

一种从视觉稿直接生成UI代码的方法及装置

(57) 摘要

本发明涉及移动应用开发领域,特别是涉及一种从视觉稿直接生成UI代码的方法及装置。该方法及装置包括下面六个部分:一、经过预打标的组件库,包括基础组件和业务组件;组件库亦同时为设计师提供快速构建页面的能力;二、实现不同主题色、字体的录入和导出;三、实现组件状态在SKETCH面板的可视化切换;四、对设计稿图层信息进行识别,包括组件的打标信息和组件库的属性和状态信息;五、布局识别和DSL树生成;六、DSL解析和实现UI代码在各平台的生成和资源导出。



1. 一种从视觉稿直接生成UI代码的方法,其特征在于,包括以下流程:

S1,输入文件类型为SKETCH文件设计文档;

S2,通过解析SKETCH的图层信息,可以得到设计文稿中的,图层结构、模板结构、图层信息,以此来划分最小的不可切割的块;

S3,图层结构获取划分完毕后,需要根据图层的数据描述信息,坐标信息,进行图层布局的合并、布局拆分及布局的预测,解析生成相关的布局属性,为了进行多平台的布局适配,采用Flex布局,在这一步,会对图层的不可分割的最小块进行横向排列和纵向排列的循环递归,直到所有的块都能放在对应产生的“格子”里,在元素相互叠加的状态下,产生绝对布局,将相互叠加的元素放在叠加后识别的“格子”里;

S4,在已经初步布局识别的格栅中,根据视图结构信息和图层属性,进行布局的预测的修正,根据图层的结构和属性,判断图层的伸缩性以此选取合适的布局容器;通过对设计文档中的视图结构信息和图层属性读取和类型匹配,映射为对应的移动端的UI元素,提炼视图结构信息和图层属性,由于模板信息中有OVERRIDE属性描述、基础组件有预打标信息描述,填充UI元素的属性结构,在图层的遍历过程中,还可以获取到图层的依赖数据,作为资源、素材的导出依赖;

S5,查找当前图层依赖关系,重复S1-S4,直到当前图层及依赖的sketch的外部引用文件图层信息全部识别完成;进入修正阶段,由操作人员控制,修正解析图层、解析布局、解析属性的偏差;除此之外,对于非组件元素、异常的视图元素进行指定规则的元素转化或者抛弃;

S6,上述步骤完成后,整个视图树基本形成,以下步骤是语义化定义页面、定义内容,在语义化阶段会继续对视图树进行修改,在语义化阶段,会对视图树中的节点进行命名的分析、节点与节点之间的关系分析,还有对多状态组件进行语义定义,此项为数据依赖关系,节点命名等提供支撑;

S7,分析节点信息、修正信息及语义化信息,根据组件与数据规则,生成图片数据映射信息、文本数据映射信息和状态映射信息,其中还分成,静态数据和动态数据,数据依赖语义化分析和修正信息;

S8,在上述内容准备就绪后,即可以开始进行DSL的推导,生产成为标准语义的视图树、样式树以及对应的业务接口描述;

S9,DSL编译器可以通过对组件跟视图布局的配置,载入对应不同的编译器,生成对应不同平台的代码;

S10,合并代码到项目中,需要项目有对应的业务层支撑框架。

2. 一种从视觉稿直接生成UI代码的装置,其特征在于,包括第一层、第二层、第三层和第四层及两个工具库,

第一层为识别层:

识别层内主要识别输入文件资源,并将文件资源内,识别图层结构、基础组件的识别和预测,通过算法,对布局进行基础的识别和预测,在组件库的支撑下,可以对已有的业务组件进行匹配使用,识别组件的属性,对热区资源进行识别,预测组件的内容数据为可变类型还是非可变类型,提取图层信息,使结构树内的节点信息语义化;

第二层为修正层:

修正层的工作是内容的修正,包含图层信息修正,组件识别修正,布局及布局属性修正,组件属性修正,语义化项修正,除此外,对交互、数据绑定等上层未带入数据,进行预测或者修正,修正主要通过操作人员在可视化工具上进行属性的提交及变更;

第三层为表达层,表达层的主要步骤为:

S1,根据以上信息生成DSL;

S2,根据DSL编译器生成对应代码,DSL规定怎样描述当前页面/组件的完整结构树、引用的样式还有业务相关的数据导出规则,交互规则,DSL编译器中以插件形式,实现固定DSL输入,在在插件内部处理,以输出对应语言环境下的代码;

第四层为接入层:

在组件代码转换后,需要接入层来实现高质量的UI代码与业务代码分离,且UI代码高复用高可用,故需要在实现组件UI的同时,完成UI跟业务的分离,所以需要在不同的编译器中,就相同的接口及视图数据模型,实现不同的代码生产规则,以及提供项目代码原生接入、混编接入、资源接入的快速集成方式;

第一个工具库为组件库:

组件库工具主要功能是预打标的基础组件、功能组件和已生产入库的业务组件及修正修正后的业务组件,为识别提供样本,以及在设计工具或者可视化工具中,对颗粒组件甚至整个页面元素进行复用;

第二个工具库为可视化工具库:

可视化工具提供代码生产预览、平台无关性的跨端UI样式预览,以及即时性的修正结果,在可视化中,还可以对历史组件构建版本进行控制及修正,可以补充交互及组件属性,以及项目嵌入接入方法的可视化选择。

3.根据权利要求2所述的一种从视觉稿直接生成UI代码的装置,其特征在于,在该装置中,主要参与角色有两个:设计师和开发人员,

设计师的主要操作为:使用预打标的组件库新建或修改SKETCH文档,以形成视觉设计稿;

开发人员的主要操作为:对设计师完成的视觉设计稿进行识别,实现项目下某一页面的整体代码导出或者页面下的某组件元素代码导出;

设计师的具体操作步骤为:

S1,使用组件库下的基础组件、功能组件、业务组件进行元器件的组合和属性的重写形成业务组件,组件库下的基础组件、功能组件是一组已经经过预先打标的sketch图层,用以标识图层组的对应各平台下的组件的类型及样式属性;

S2,在形成新的业务组件后,可将新的业务组件上传至组件库,以实现复用;

S3,在制作页面时,使用组件库中的基础组件、功能组件或者业务组件进行拼装组合,对组件的位置、数据进行修改,实现UI效果图的编辑;

S4,在修改SKETCH文档后,软件会自动对所有修改图层及影响图层添加修改标签,用以标记在项目迭代过程中的数据修改;

开发人员的具体操作步骤为:

S1,描述展示设计师对设计文档的上次修改信息;

S2,对设计师图层、组件的类型、属性和使用布局进行修正;

S3,代码的导出及修改标记的清除。

一种从视觉稿直接生成UI代码的方法及装置

技术领域

[0001] 本发明涉及移动应用开发领域,特别涉及一种从视觉稿直接生成UI代码的方法及装置。

背景技术

[0002] 在移动应用开发中,UI的构建、修改迭代占据了整个项目开发中的很大一部分时间。尤其是在编写高质量的UI代码情况下,需要保证UI代码和业务代码的分离与控制;在目前的开发方式下,由于开发人员的代码质量参差不齐,经常导致UI结构混乱,UI代码与业务重叠,多人开发的项目环境下,导致UI碎片化,UI库维护困难以及代码的冗余,直接或间接的导致代码运行效率、调试效率以及包体的过大等;另外在多平台下,多个开发人员不同的开发方式使每个平台的展示效果与设计图的还原度难以一致;在代码生成层面,由于孤立的设计稿难以提供足够的元素类别信息,布局信息,状态信息以及最终的代码生成逻辑信息。因此,业内实现方式主要包括两种:

- 1.在Sketch中预置组件的模板代码,即只提供静态代码输出:如Dapollo插件
- 2.借助人工智能技术,通过模型训练,识别设计稿中的元素类别并动态生成代码:如imgcook工具

目前现有的技术存在如下问题:

- 1.模板代码缺乏动态生产代码的能力,难以适应实际复杂场景高可用代码的需求。
- [0003] 2.人工智能技术识别的准确率尚不足以应用于实际生产,且训练模型对数据的数量和质量要求甚高。

发明内容

[0004] 本发明要解决的技术问题是克服现有技术的缺陷,提供一种从视觉稿直接生成UI代码的方法及装置。

[0005] 为了解决上述技术问题,本发明提供了如下的技术方案:

本发明提供一种从视觉稿直接生成UI代码的方法,包括以下流程:

S1,输入文件类型为SKETCH文件设计文档;

S2,通过解析SKETCH的图层信息,可以得到设计文稿中的,图层结构、模板结构、图层信息,以此来划分最小的不可切割的块;

S3,图层结构获取划分完毕后,需要根据图层的数据描述信息,坐标信息,进行图层布局的合并、布局拆分及布局的预测,解析生成相关的布局属性,为了进行多平台的布局适配,采用Flex布局,在这一步,会对图层的不可分割的最小块进行横向排列和纵向排列的循环递归,直到所有的块都能放在对应产生的“格子”里,在元素相互叠加的状态下,产生绝对布局,将相互叠加的元素放在叠加后识别的“格子”里;

S4,在已经初步布局识别的格栅中,根据视图结构信息和图层属性,进行布局的预测的修正,根据图层的结构和属性,判断图层的伸缩性以此选取合适的布局容器;通过对设计文

档中的视图结构信息和图层属性读取和类型匹配,映射为对应的移动端的UI元素,提炼视图结构信息和图层属性,由于模板信息中有OVERRIDE属性描述、基础组件有预打标信息描述,填充UI元素的属性结构,在图层的遍历过程中,还可以获取到图层的依赖数据,作为资源、素材的导出依赖;

S5,查找当前图层依赖关系,重复S1-S4,直到当前图层及依赖的sketch的外部引用文件图层信息全部识别完成;进入修正阶段,由操作人员控制,修正解析图层、解析布局、解析属性的偏差;除此之外,对于非组件元素、异常的视图元素进行指定规则的元素转化或者抛弃;

S6,上述步骤完成后,整个视图树基本形成,以下步骤是语义化定义页面、定义内容,在语义化阶段会继续对视图树进行修改,在语义化阶段,会对视图树中的节点进行命名的分析、节点与节点之间的关系分析,还有对多状态组件进行语义定义,此项为数据依赖关系,节点命名等提供支撑;

S7,分析节点信息、修正信息及语义化信息,根据组件与数据规则,生成图片数据映射信息、文本数据映射信息和状态映射信息,其中还分成,静态数据和动态数据,数据依赖语义化分析和修正信息;

S8,在上述内容准备就绪后,即可以开始进行DSL的推导,生产成为标准语义的视图树、样式树以及对应的业务接口描述;

S9,DSL编译器可以通过对组件跟视图布局的配置,载入对应不同的编译器,生成对应不同平台的代码;

S10,合并代码到项目中,需要项目有对应的业务层支撑框架。

[0006] 本发明提供一种从视觉稿直接生成UI代码的装置,其特征在于,包括第一层、第二层、第三层和第四层及两个工具库,

第一层为识别层:

识别层内主要识别输入文件资源,并将文件资源内,识别图层结构、基础组件的识别和预测,通过算法,对布局进行基础的识别和预测,在组件库的支撑下,可以对已有的业务组件进行匹配使用,识别组件的属性,对热区资源进行识别,预测组件的内容数据为可变类型还是非可变类型,提取图层信息,使结构树内的节点信息语义化;

第二层为修正层:

修正层的工作是内容的修正,包含图层信息修正,组件识别修正,布局及布局属性修正,组件属性修正,语义化项修正,除此外,对交互、数据绑定等上层未带入数据,进行预测或者修正,修正主要通过操作人员在可视化工具上进行属性的提交及变更;

第三层为表达层,表达层的主要步骤为:

S1,根据以上信息生成DSL;

S2,根据DSL编译器生成对应代码,DSL规定怎样描述当前页面/组件的完整结构树、引用的样式还有业务相关的数据导出规则,交互规则,DSL编译器中以插件形式,实现固定DSL输入,在在插件内部处理,以输出对应语言环境下的代码;

第四层为接入层:

在组件代码转换后,需要接入层来实现高质量的UI代码与业务代码分离,且UI代码高复用高可用,故需要在实现组件UI的同时,完成UI跟业务的分离,所以需要在不同的编译器

中,就相同的接口及视图数据模型,实现不同的代码生产规则,以及提供项目代码原生接入、混编接入、资源接入的快速集成方式;

第一个工具库为组件库:

组件库工具主要功能是预打标的基础组件、功能组件和已生产入库的业务组件及修正修正后的业务组件,为识别提供样本,以及在设计工具或者可视化工具中,对颗粒组件甚至整个页面元素进行复用;

第二个工具库为可视化工具库:

可视化工具提供代码生产预览、平台无关性的跨端UI样式预览,以及即时性的修正结果,在可视化中,还可以对历史组件构建版本进行控制及修正,可以补充交互及组件属性,以及项目嵌入接入方法的可视化选择。

[0007] 作为本发明的一种优选技术方案,在该装置中,主要参与角色有两个:设计师和开发人员,

设计师的主要操作为:使用预打标的组件库新建或修改SKETCH文档,以形成视觉设计稿;

开发人员的主要操作为:对设计师完成的视觉设计稿进行识别,实现项目下某一页面的整体代码导出或者页面下的某组件元素代码导出;

设计师的具体操作步骤为:

S1,使用组件库下的基础组件、功能组件、业务组件进行元器件的组合和属性的重写形成业务组件,组件库下的基础组件、功能组件是一组已经经过预先打标的sketch图层,用以标识图层组的对应各平台下的组件的类型及样式属性;

S2,在形成新的业务组件后,可将新的业务组件上传至组件库,以实现复用;

S3,在制作页面时,使用组件库中的基础组件、功能组件或者业务组件进行拼装组合,对组件的位置、数据进行修改,实现UI效果图的编辑;

S4,在修改SKETCH文档后,软件会自动对所有修改图层及影响图层添加修改标签,用以标记在项目迭代过程中的数据修改;

开发人员的具体操作步骤为:

S1,描述展示设计师对设计文档的上次修改信息;

S2,对设计师图层、组件的类型、属性和使用布局进行修正;

S3,代码的导出及修改标记的清除。

[0008] 与现有技术相比,本发明的有益效果如下:

1.使用预打标的基础组件库,使设计稿不再只是孤立图层元素信息。打标信息包括组件类别和组件各属性信息,作为后续根据设计稿生产代码的信息支撑;

2.基础组件库内的组件可用于组合生成业务组件,设计稿中任意复杂的业务元素都可由有限种类的基础组件组成,由此业务组件亦包含了打标信息,且使整个业务组件的生产过程灵活多变;

3.基础组件除了能够提供生产代码的打标信息,还能用于辅助设计师构建页面,设计师只需拖拽组件库中的组件并通过属性面板调整所需的组件样式即可快速的构建设计稿;

4.代码生产过程包括布局识别和预测,根据元素信息,布局信息,生成DSL结构树,通过提供布局识别能力,使代码生成能够使用复杂业务场景下的非格式化UI;

5.生成的DSL结构树,通过各个平台的编译器解析成对应平台的UI实现代码,从而使代码生成能够支持不同平台代码的同时导出;

6.将UI设计稿的构建和最终代码的生产整个链路打通,在提高设计师工作效率的同时提供了智能代码生产的能力。

附图说明

[0009] 附图用来提供对本发明的进一步理解,并且构成说明书的一部分,与本发明的实施例一起用于解释本发明,并不构成对本发明的限制。在附图中:

图1为总体流程示意图;

图2为角色处理流程图;

图3为数据业务流程图;

图4为模块分层结构图;

图5为项目工作流程图。

具体实施方式

[0010] 以下结合附图对本发明的优选实施例进行说明,应当理解,此处所描述的优选实施例仅用于说明和解释本发明,并不用于限定本发明。

[0011] 实施例1

如图1-5所示,本发明提供一种从视觉稿直接生成UI代码的方法,包括以下流程:

S1,输入文件类型为SKETCH文件设计文档(视图结构),视觉稿的创建和编辑软件为SKETCH;

S2,通过解析SKETCH的图层信息,可以得到设计文稿中的,图层结构、模板结构、图层信息,以此来划分最小的不可切割的块;

S3,图层结构获取划分完毕后,需要根据图层的数据描述信息,坐标信息,进行图层布局的合并、布局拆分及布局的预测,解析生成相关的布局属性,为了进行多平台的布局适配,采用Flex布局,在这一步,会对图层的不可分割的最小块进行横向排列和纵向排列的循环递归,直到所有的块都能放在对应产生的“格子”里,在元素相互叠加的状态下,产生绝对布局,将相互叠加的元素放在叠加后识别的“格子”里;

S4,在已经初步布局识别的格栅中,根据视图结构信息和图层属性进行布局的预测的修正,根据图层的结构和属性判断图层的伸缩性以此选取合适的布局容器。通过对设计文档中的视图结构信息和图层属性读取和类型匹配,映射为对应的移动端的UI元素。提炼视图结构信息和图层属性,由于模板信息中有OVERRIDE属性描述、基础组件有预打标信息描述,用以填充UI元素的各种属性。在图层的遍历过程中,还可以获取到图层的依赖数据,作为资源、素材的导出依赖。

[0012] 名词解释:override属性-sketch软件提供的symbol元素的属性。可以在创建symbolInstance时,使用override属性对SymbolMaster元素上的预设值进行修改。override属性包括三类,组件中内嵌的子symbol标识,文本信息设置和图片信息设置;

S5,查找当前图层依赖关系,重复S1-S4,直到当前图层及依赖的sketch的外部引用文件图层信息全部识别完成;进入修正阶段,由操作人员控制,修正解析图层、解析布局、解析

属性的偏差;除此之外,对于非组件元素、异常的视图元素进行指定规则的元素转化或者抛弃;

S6,上述步骤完成后,整个视图树基本形成,以下步骤是语义化定义页面、定义内容,在语义化阶段会继续对视图树进行修改。在语义化阶段,会对视图树中的节点进行命名的分析、节点与节点之间的关系分析。关系分析包含了根据节点属性类型、节点上下问及语义化的分析,对节点结构进行复用类型合并。比如在列表容器中的多个cell类型的合并。还有对多状态组件进行语义定义,此项为数据依赖关系,节点命名等提供支撑;

S7,分析节点信息、修正信息及语义化信息,根据组件与数据规则,生成图片数据映射信息、文本数据映射信息和状态映射信息,其中还分成,静态数据和动态数据,数据依赖语义化分析和修正信息;

S8,在上述内容准备就绪后,即可以开始进行DSL的推导,生产成为标准语义的视图树、样式树以及对应的业务接口描述;

S9,DSL编译器可以通过对组件跟视图布局的配置,载入对应不同的编译器,生成对应不同平台的代码;

S10,合并代码到项目中,需要项目有对应的业务层支撑框架,

本发明提供一种从视觉稿直接生成UI代码的装置,包括:第一层、第二层、第三层和第四层及两个工具库,

第一层为识别层:

识别层内主要识别输入的sketch文件,并在sketch文件中,识别图层结构、基础组件的识别和预测,通过算法对布局进行基础的识别和预测,在组件库的支撑下,对已有的业务组件进行匹配,识别组件的各种属性,例如组件类型、字体、颜色、位置等。对热区资源进行识别,预测组件的内容数据为可变类型还是非可变类型,提取图层信息,使结构树内的节点信息语义化;

第二层为修正层:

修正层的工作是内容的修正,包含图层信息修正,组件识别修正,布局及布局属性修正、组件属性修正、语义化项修正,除此外,对交互、数据绑定等上层未带入数据,进行预测或者修正。修正后,修正数据上传至组件库中,以此记录修正后的组件数据,完善组件模型,以及保存必要的业务组件块,用以复用。修正主要通过操作人员在可视化工具上进行属性的提交及变更;

第三层为表达层,表达层的主要步骤为:

S1,根据以上信息生成DSL;

S2,根据DSL编译器生成对应代码,DSL规定怎样描述当前页面/组件的完整结构树、引用的样式还有业务相关的数据导出规则,交互规则,DSL编译器中以插件形式,实现固定DSL输入,在在插件内部处理,以输出对应语言环境下的代码;

第四层为接入层:

在组件代码转换后,需要接入层来实现高质量的UI代码与业务代码分离,且UI代码高复用高可用,故需要在实现组件UI的同时,完成UI跟业务的分离,所以需要在不同的编译器中,就相同的接口及视图数据模型,实现不同的代码生产规则,以及提供项目代码原生接入、混编接入、资源接入的快速集成方式;

第一个工具库为组件库：

组件库工具主要功能是预打标的基础组件、功能组件和已生产入库的业务组件及修正修正后的业务组件，为识别提供样本，以及在设计工具或者可视化工具中，对颗粒组件甚至整个页面元素进行复用；

第二个工具库为可视化工具库：

可视化工具提供代码生产预览、平台无关性的跨端UI样式预览，以及即时性的修正结果。在可视化中，还可以对历史组件构建版本进行控制及修正，可以补充交互及组件属性，以及项目嵌入接入方法的可视化选择。

[0013] 在该装置中，主要参与角色有两个：设计师和开发人员，

设计师的主要操作为：使用预打标的组件库新建或修改SKETCH文档，以形成视觉设计稿；

开发人员的主要操作为：对设计师完成的视觉设计稿进行识别，实现项目下某一页面的整体代码导出或者页面下的某组件元素代码导出；

设计师的具体操作步骤为：

S1，使用组件库下的基础组件、功能组件、业务组件进行元器件的组合和属性的重写形成业务组件。组件库下的基础组件、功能组件是一组已经经过预先打标的sketch图层，用以标识图层组的对应各平台下的组件的类型及样式属性；

S2，在形成新的业务组件后，可将新的业务组件上传至组件库，以实现复用；

S3，在制作页面时，使用组件库中的基础组件、功能组件或者业务组件进行拼装组合，对组件的位置、数据进行修改，实现UI效果图的编辑；

S4，在修改SKETCH文档后，软件会自动对所有修改图层及影响图层添加修改标签，用以标记在项目迭代过程中的数据修改；

开发人员的具体操作步骤为：

S1，描述展示设计师对设计文档的上次修改信息；

S2，对设计师图层、组件的类型、属性和使用布局进行修正；

S3，代码的导出及修改标记的清除。

[0014] 如何识别图层信息：

Sketch软件为sketch插件开发者提供了一套基于CocoaScript语言开发框架。通过js，可以获取到sketch软件上面所有对应的文档信息，图层信息等。也可以通过sketch的API，或者MacOS API对sketch软件上展示的任何UI元素进行获取跟修改。

[0015] 如何识别组件类型：

Sketch软件文档保存是以json数据结构保存的一颗以page为根的页面描述树。任一元素对应文档中都有对应的Json格式的数据节点。我们使用插件的形式，在组件库中对一组元素进行预先打标，这样在使用组件库进行组件的生产时，图层组就已经有了含义，标明了这个图形对应的基础组件元素类型是什么，及属性是什么；

名词解释：打标-打标的意思是将类型、属性等信息，写入到元素对应Json结构下的自定义节点中。

[0016] 如何兼容未打标的组件：

兼容了未打标的组件，即直接使用sketch软件提供的创建图层及图层组合情况下的组

件,这种情况下,视为这个元素是静态的,即没有状态,没有事件,对应的在识别和代码生产的过程中,对类型进行了映射。

[0017] 目前现有的技术存在如下问题:

1.模板代码缺乏动态生产代码的能力,难以适应实际复杂场景高可用代码的需求。

[0018] 2.人工智能技术识别的准确率尚不足以应用于实际生产,且训练模型对数据的数量和质量要求甚高。

[0019] 为解决上述问题,本发明提出了一种基于SKETCH组件预打标的支持多平台动态UI代码生成的方法和装置,其创新点为:

1.使用预打标的基础组件库,使设计稿不再只是孤立图层元素信息。打标信息包括组件类别和组件各属性信息,作为后续根据设计稿生产代码的信息支撑;

2.基础组件库内的组件可用于组合生成业务组件,设计稿中任意复杂的业务元素都可由有限种类的基础组件组成,由此业务组件亦包含了打标信息,且使整个业务组件的生产过程灵活多变;

3.基础组件除了能够提供生产代码的打标信息,还能用于辅助设计师构建页面,设计师只需拖拽组件库中的组件并通过属性面板调整所需的组件样式即可快速的构建设计稿;

4.代码生产过程包括布局识别和预测,根据元素信息,布局信息,生成DSL结构树。通过提供布局识别能力,使代码生成能够使用复杂业务场景下的非格式化UI;

5.生成的DSL结构树,通过各个平台的编译器解析成对应平台的UI实现代码,从而使代码生成能够支持不同平台代码的同时导出;

6.将UI设计稿的构建和最终代码的生产整个链路打通,在提高设计师工作效率的同时提供了智能代码生产的能力。

[0020] 对于市面上常见的设计师的常用辅助工具,此发明重点关注在于,在辅助设计师快速构建页面的同时,生产高质量的可用代码。

[0021] 一.对于常见工具不同,本发明将预打标与设计师使用的提高效率和沉淀标准的组件库合并在一起,以此在方便设计师的同时,可以进行真实的组件元素的映射。

[0022] 二.由于在真实的使用情况中,可能会产生复杂的组件组合情况,此发明允许设计师自行通过基础组件进行拼装,形成各场景适用的业务组件。

[0023] 三.为了兼容使用场景,将原sketch内的图层元素,作为静态元素识别。在基础组件中,提供了动态内容导出和布局的适配。

[0024] 四.使用Flex布局进行多平台的适配。根据坐标信息,对所有元素进行格栅化,形成布局的预测,并通过内部元素的可伸缩性,进行布局属性调整,比如由于组件内部动态高,所以外部布局会更改为动态高度,并且获取和计算现有情况下,上下内边距的距离,修改外部布局容器的上下内边距,以适配动态高度。

[0025] 名词解释:元素的可伸缩性-元素的大小是否随着内容改变,即是否动态宽、动态高。

[0026] 五.由于一个文档的操作方有两个人,所以在文档的流转过程中,需要标识出迭代过程的修改内容。所以在设计师操作修改图层信息后,会对对应的图层进行内容标记。在生产代码后,一个周期结束,清除标记。

[0027] 六.UI的视图识别后,会生产出一个DSL规范的结构树。这个树结构,完整的描述了

这个树的元素结构、样式结构、属性导出结构,还包含了资源依赖和图层依赖的信息。而这些信息,可以进行任何语言的翻译,称之为DSL编译。

[0028] 与现有技术相比,本发明的有益效果如下:

1.使用预打标的基础组件库,使设计稿不再只是孤立图层元素信息。打标信息包括组件类别和组件各属性信息,作为后续根据设计稿生产代码的信息支撑;

2.基础组件库内的组件可用于组合生成业务组件,设计稿中任意复杂的业务元素都可由有限种类的基础组件组成,由此业务组件亦包含了打标信息,且使整个业务组件的生产过程灵活多变;

3.基础组件除了能够提供生产代码的打标信息,还能用于辅助设计师构建页面,设计师只需拖拽组件库中的组件并通过属性面板调整所需的组件样式即可快速的构建设计稿;

4.代码生产过程包括布局识别和预测,根据元素信息,布局信息,生成DSL结构树,通过提供布局识别能力,使代码生成能够使用复杂业务场景下的非格式化UI;

5.生成的DSL结构树,通过各个平台的编译器解析成对应平台的UI实现代码,从而使代码生成能够支持不同平台代码的同时导出;

6.将UI设计稿的构建和最终代码的生产整个链路打通,在提高设计师工作效率的同时提供了智能代码生产的能力。

[0029] 最后应说明的是:以上所述仅为本发明的优选实施例而已,并不用于限制本发明,尽管参照前述实施例对本发明进行了详细的说明,对于本领域的技术人员来说,其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

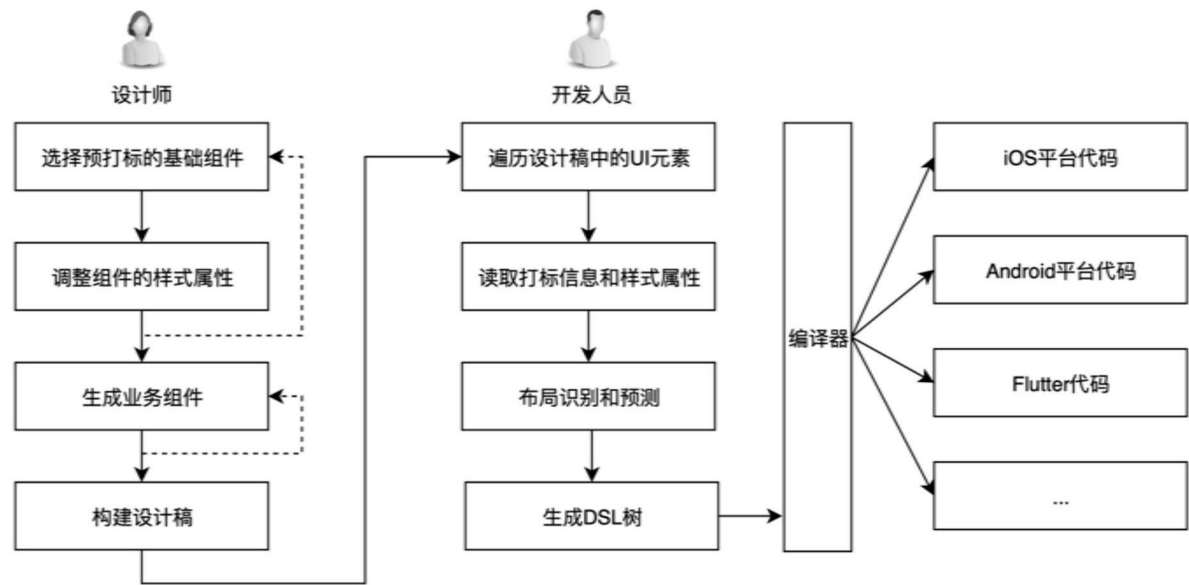


图1

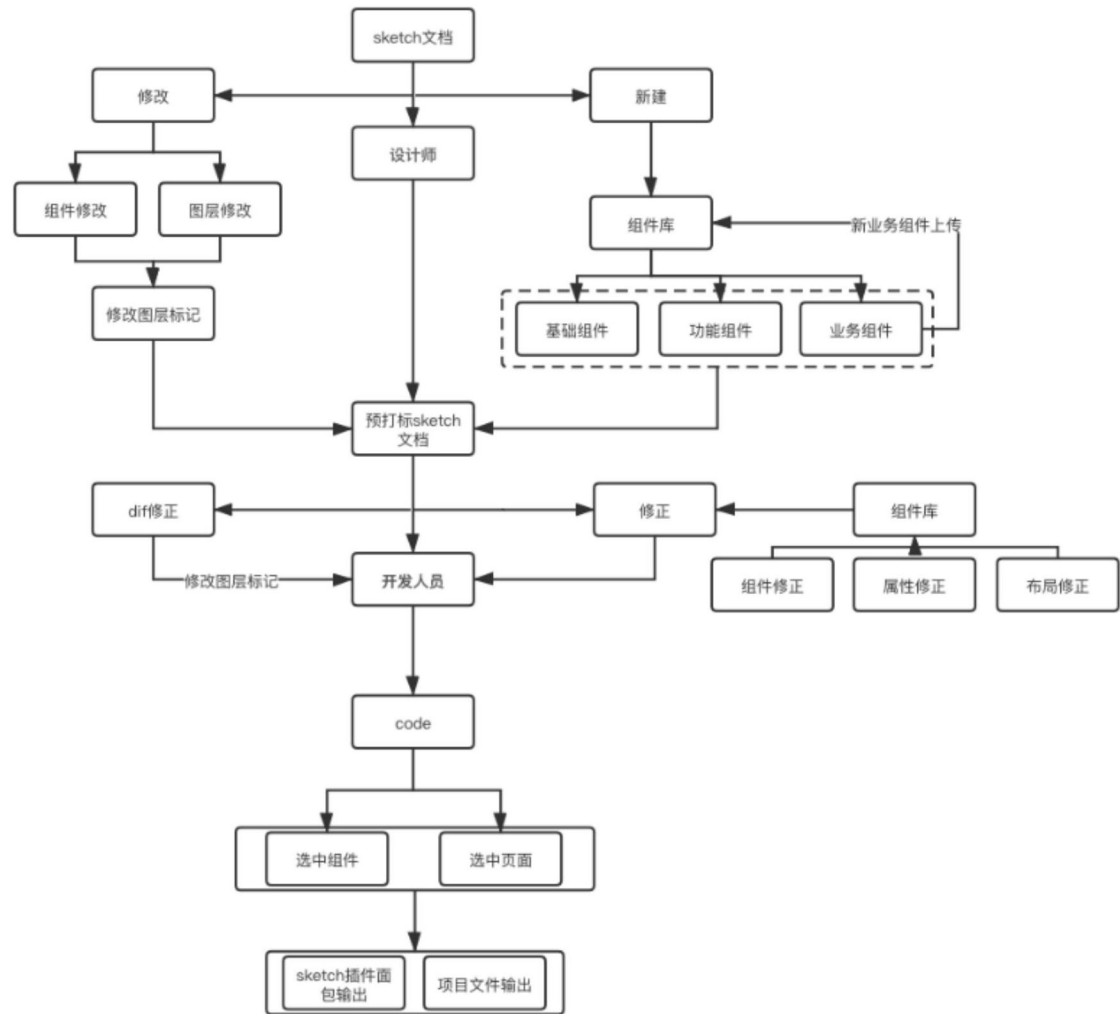


图2

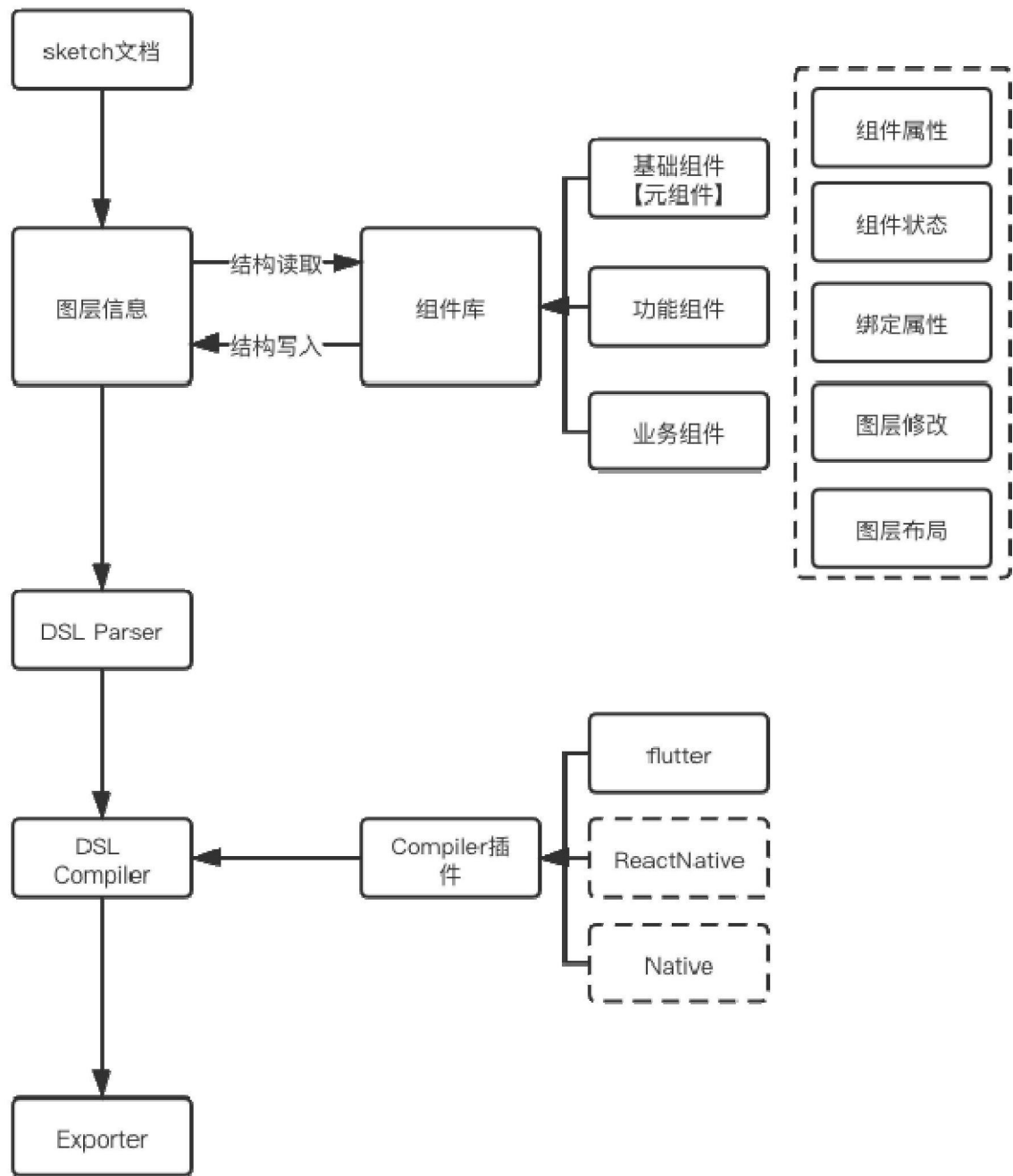


图3

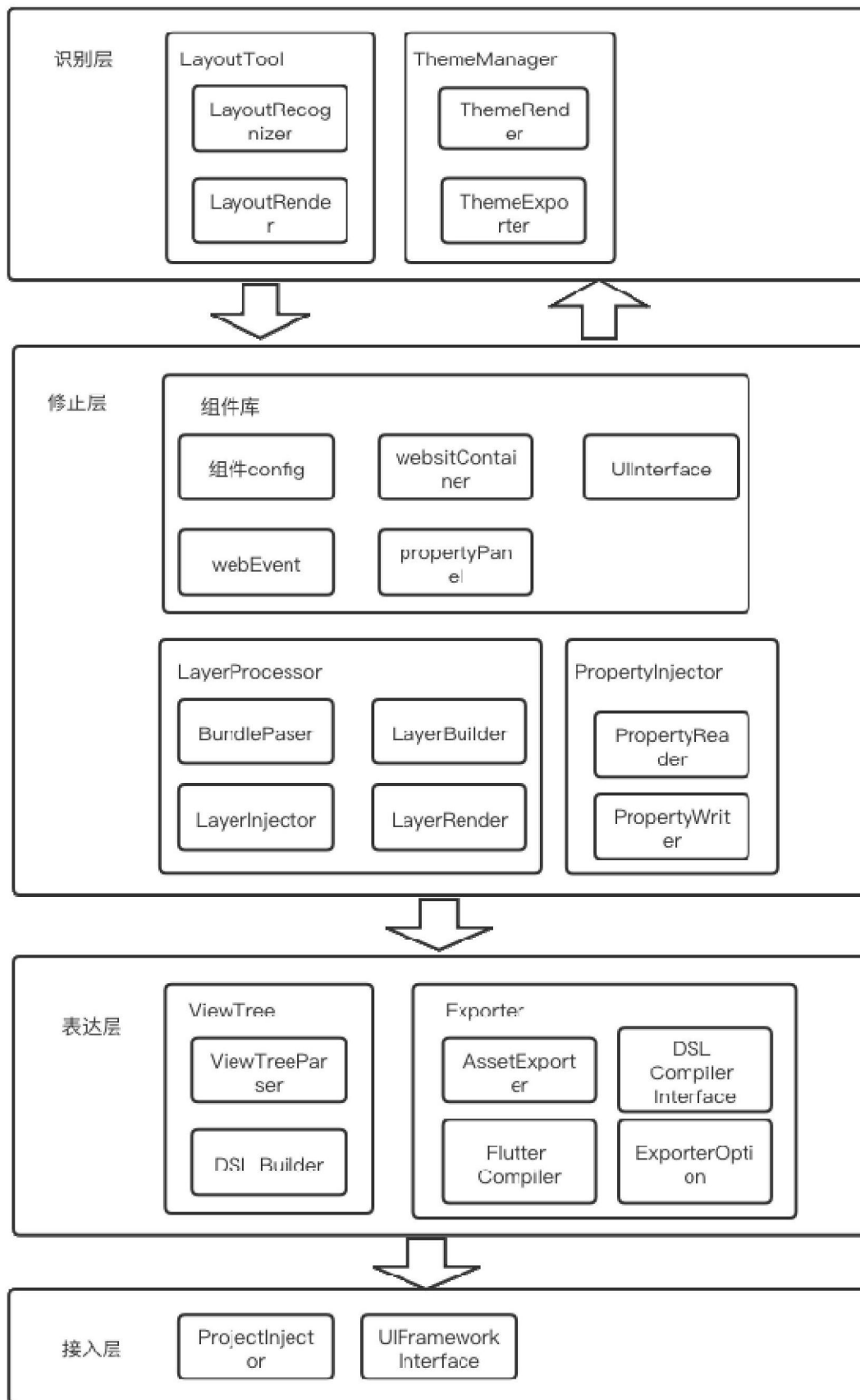


图4

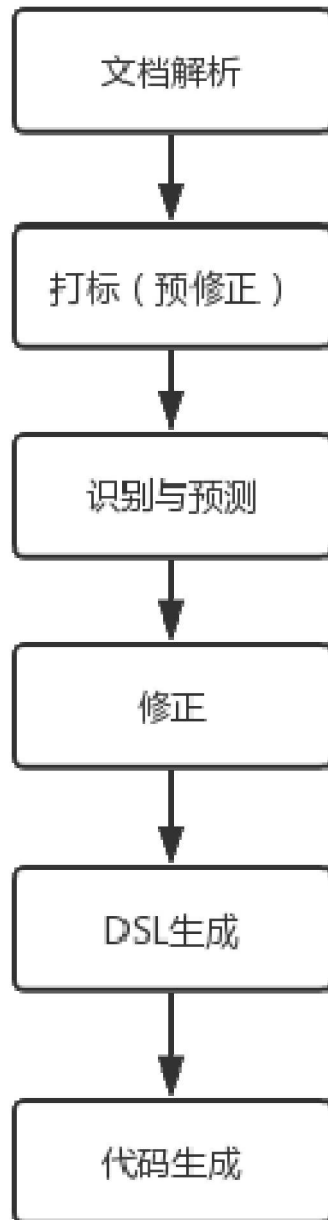


图5