

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局



(10) 国际公布号

WO 2019/233293 A1

(43) 国际公布日
2019 年 12 月 12 日 (12.12.2019)

(51) 国际专利分类号:
G06F 8/20 (2018.01)

(21) 国际申请号: PCT/CN2019/088258

(22) 国际申请日: 2019 年 5 月 24 日 (24.05.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

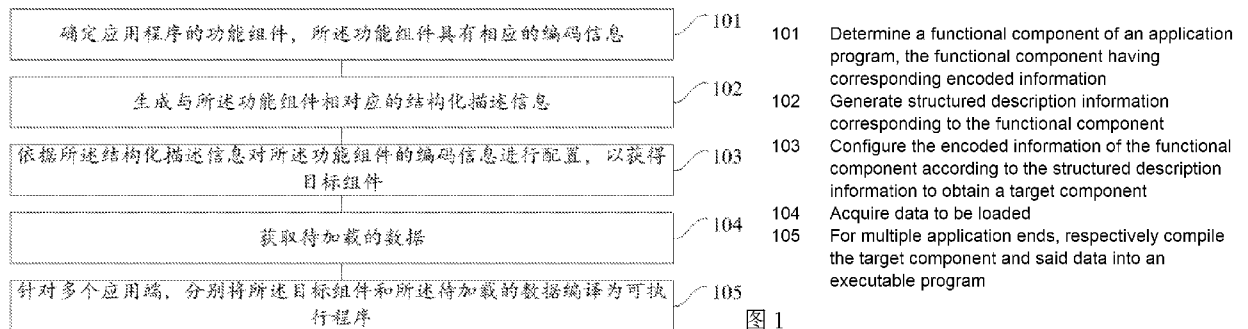
(30) 优先权:
201810580599.9 2018年6月7日 (07.06.2018) CN

(71) 申请人: 阿里巴巴集团控股有限公司 (ALIBABA GROUP HOLDING LIMITED) [—/CN]; 开曼群岛
大开曼资本大厦一座四层 847 号 邮箱,
Grand Cayman (KY)。

(72) 发明人: 李帆(LI, Fan); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 万宁邦(WAN, Ningbang); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 张果(ZHANG, Guo); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 刘彦玮(LIU, Yanwei); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 陈智浩(CHEN, Zhihao); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。 李春城(LI, Chuncheng);

(54) Title: METHOD AND APPARATUS FOR DEVELOPING APPLICATION PROGRAM

(54) 发明名称: 一种应用程序的开发方法和装置



(57) Abstract: A method and apparatus for developing an application program. The method comprises: determining a functional component of an application program, the functional component having corresponding encoded information (101); generating structured description information corresponding to the functional component (102); configuring the encoded information of the functional component according to the structured description information to obtain a target component (103); acquiring data to be loaded (104); and for multiple application ends, respectively compiling the target component and said data into an executable program (105). According to the method, the same set of code writing methods can be used on different application ends and the code is then built and translated into executable programs supported by different application ends without it being necessary to separately conduct development for different application ends, realizing the reuse of a set of service logic on multiple application ends, improving the development efficiency of an application program and reducing the development cost of the application program.

(57) 摘要: 一种应用程序的开发方法和装置, 所述方法包括: 确定应用程序的功能组件, 所述功能组件具有相应的编码信息 (101); 生成与所述功能组件相对应的结构化描述信息 (102); 依据所述结构化描述信息对所述功能组件的编码信息进行配置, 以获得目标组件 (103); 获取待加载的数据 (104); 针对多个应用端, 分别将所述目标组件和所述待加载的数据编译为可执行程序 (105)。该方法可以在不同的应用端使同一套代码编写方式, 然后通过构建翻译成不同的应用端支持的可执行程序, 无需针对不同的应用端分别进行开发, 实现了一套业务逻辑的在多个应用端的复用, 提高了应用程序的开发效率, 降低了应用程序的开发成本。

中国浙江省杭州市余杭区文一西路969号3号楼5楼
阿里巴巴集团法务部, Zhejiang 311121 (CN)。

(74) 代理人: 北京三友知识产权代理有限公司 (BEIJING SANYOU INTELLECTUAL PROPERTY AGENCY LTD.); 中国北京市金融街35号国际企业大厦A座16层, Beijing 100033 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

一种应用程序的开发方法和装置

本申请要求 2018 年 06 月 07 日递交的申请号为 201810580599.9、发明名称为“一种应用程序的开发方法和装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

技术领域

本申请涉及计算机技术领域，特别是涉及一种应用程序的开发方法和一种应用程序的开发装置。

10 背景技术

React 是一个用于构建用户界面的 JavaScript（一种直译式脚本语言）库，其设计思想极其独特，性能也十分出众，然而代码逻辑却非常简单。于是有越来越多的开发人员开始关注和使用 React，并通过 React 进行应用程序的开发。

15 在基于 React 进行的应用程序的开发过程中，开发人员首先需要将业务需求拆分为具体步骤，然后按照各个步骤的顺序依次去实现、集成，直到最终完成整个应用程序的开发。在此过程中，开发形式往往是以串行方式，而非协议配置的形式来进行的。

按照 React 此种开发方式，其最终产出的语法结构是类似 XML（Extensible Markup Language，可扩展标记语言）结构，通过该结构进行应用程序的搭建，如果仅仅是应用在一个平台上，不会有特别的问题出现。但是，针对具体编程实现来完成的开发任务，
20 很难做到将同一种业务模型和逻辑应用到其他不同的平台上，使得基于 React 开发完成的应用程序难以适配多个不同的平台。如果要将该应用程序应用在多个平台上，就需要针对各个平台分别进行开发，导致开发成本的急剧增加。

发明内容

25 鉴于上述问题，提出了本申请实施例以便提供一种克服上述问题或者至少部分地解决上述问题的一种应用程序的开发方法和相应的一种应用程序的开发装置。

为了解决上述问题，本申请公开了一种应用程序的开发方法，包括：

确定应用程序的功能组件，所述功能组件具有相应的编码信息；

生成与所述功能组件相对应的结构化描述信息；

30 依据所述结构化描述信息对所述功能组件的编码信息进行配置，以获得目标组件；

获取待加载的数据；

针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序。

可选地，所述生成与所述功能组件相对应的结构化描述信息的步骤包括：

确定各个功能组件的逻辑实现信息；

5 将所述逻辑实现信息转换为结构化描述信息。

可选地，所述结构化描述信息为 Json Schema 描述信息。

可选地，所述依据所述结构化描述信息对所述功能组件的编码信息进行配置，以获得目标组件的步骤包括：

加载所述功能组件的编码信息至解析引擎；

10 按照所述结构化描述信息，在所述解析引擎中对所述功能组件的编码信息进行渲染，以获得目标组件。

可选地，所述加载所述功能组件的编码信息至解析引擎的步骤包括：

调用预设的注册函数；

通过所述注册函数，将所述功能组件的编码信息注册至解析引擎。

15 可选地，所述获取待加载的数据的步骤包括：

确定各个功能组件的数据配置规则；

按照所述数据配置规则，获取相应的待加载数据。

可选地，所述针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序的步骤包括：

20 确定目标应用端，所述目标应用端具有对应的编译方式；

按照所述编译方式，将所述目标组件和所述待加载的数据编译为所述目标应用端的可执行程序。

可选地，还包括：

在所述目标应用端中运行所述可执行程序。

25 可选地，还包括：

对所述应用程序的功能组件进行更新或修改。

可选地，所述对所述应用程序的功能组件进行更新或修改的步骤包括：

确定待更新或修改的功能组件；

对所述待更新或修改的功能组件对应的结构化描述信息进行更新或修改。

30 为了解决上述问题，本申请公开了一种应用程序的开发方法，包括：

确定应用程序的功能组件，所述功能组件具有相应的编码信息和逻辑实现信息；
将所述功能组件的逻辑实现信息转换为 Json Schema 描述信息；
加载所述功能组件的编码信息至解析引擎；

依据所述 Json Schema 描述信息，在所述解析引擎中对所述功能组件的编码信息进行

5 行配置，以获得目标组件；

获取待加载的数据；

针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序；
分别在各个应用端中运行所述可执行程序。

为了解决上述问题，本申请公开了一种应用程序的开发装置，包括：

10 确定模块，用于确定应用程序的功能组件，所述功能组件具有相应的编码信息；

生成模块，用于生成与所述功能组件相对应的结构化描述信息；

配置模块，用于依据所述结构化描述信息对所述功能组件的编码信息进行配置，以
获得目标组件；

获取模块，用于获取待加载的数据；

15 编译模块，用于针对多个应用端，分别将所述目标组件和所述待加载的数据编译为
可执行程序。

为了解决上述问题，本申请公开了一种应用程序的开发装置，包括：

确定模块，用于确定应用程序的功能组件，所述功能组件具有相应的编码信息和逻辑实现信息；

20 转换模块，用于将所述功能组件的逻辑实现信息转换为 Json Schema 描述信息；

加载模块，用于加载所述功能组件的编码信息至解析引擎；

配置模块，用于依据所述 Json Schema 描述信息，在所述解析引擎中对所述功能组件的编码信息进行配置，以获得目标组件；

获取模块，用于获取待加载的数据；

25 编译模块，用于针对多个应用端，分别将所述目标组件和所述待加载的数据编译为
可执行程序；

运行模块，用于分别在各个应用端中运行所述可执行程序。

与背景技术相比，本申请实施例包括以下优点：

本申请实施例，通过确定应用程序的各个功能组件，并生成与各个功能组件相对应的
30 的结构化描述信息，从而可以依据上述结构化描述信息对功能组件的编码信息进行配置

获得目标组件，然后在获取待加载的数据后，可以针对多个不同的应用端，分别将上述目标组件和待加载的数据编译为能够在不同的应用端上运行的可执行程序。本实施例可以在不同的应用端使同一套代码编写方式，然后通过构建翻译成不同的应用端支持的可执行程序，无需针对不同的应用端分别进行开发，实现了一套业务逻辑的在多个应用端的复用，提高了应用程序的开发效率，降低了应用程序的开发成本。

附图说明

图 1 是本申请一个实施例的一种应用程序的开发方法的步骤流程示意图；

图 2 是本申请一个实施例的另一种应用程序的开发方法的步骤流程示意图；

10 图 3 是本申请一个实施例的又一种应用程序的开发方法的步骤流程示意图；

图 4 是本申请一个实施例的一种应用程序的开发方法的原理图；

图 5 是本申请一个实施例的一种应用程序的开发装置的示意性结构框图；

图 6 是本申请一个实施例的另一种应用程序的开发装置的示意性结构框图。

15 具体实施方式

为使本申请的上述目的、特征和优点能够更加明显易懂，下面结合附图和具体实施方式对本申请作进一步详细的说明。

参照图 1，示出了本申请一个实施例的一种应用程序的开发方法的步骤流程示意图，具体可以包括如下步骤：

20 步骤 101，确定应用程序的功能组件，所述功能组件具有相应的编码信息；

需要说明的是，上述应用程序可以是应用于不同的应用平台或应用端的程序或软件。例如，Web 端、Html5 端、Native 端等等，本实施例对此不作限定。

通常，各个应用程序都包括有相应的功能，各种功能都是通过具体的功能组件来实现的。因此，在进行应用程序的开发时，开发人员首先需要确定当前开发的应用程序应当具有哪些功能组件。

在本申请实施例中，各个功能组件具有的编码信息可以是由开发人员在确定出相应的功能组件后按照特定的开发方式进行编写得到的组件代码。在具体实现中，开发人员可以使用 React Component 编写应用程序各个功能组件的代码。

30 React Component 是一个抽象基类，通过实现一个它的子类，并且至少定义一个 render()方法即可使用。React Component 允许开发人员将 UI (User Interface, 用户界面)

拆分为独立的可重用的部分，并单独地去考虑每个部分。

步骤 102，生成与所述功能组件相对应的结构化描述信息；

在本申请实施例中，在开发人员进行应用程序的开发时，为了实现一次开发便能够最大限度地多个不同的应用端上应用，可以在确定出应用程序的各个功能组件后，以

5 一种结构化的描述信息去描述各个功能组件。

在本申请实施例中，结构化描述信息可以是一种结构化的协议。对于应用程序的功能组件，可以通过该协议去进行描述或表达。

在具体实现中，上述结构化描述信息可以是 Json Schema 描述信息。

熟悉 XML 的开发人员都知道，对 XML 文档的定义有一个 XML Schema（模式）。

10 同样，对于 Json 文件的定义，也应该有一个 Json Schema 来规范 Json 文件的内容。因此，Json Schema 是一种用以标注和验证 Json 文档的元数据的文档，可以类比于 XML Schema。相对于 Json Schema，一个 Json 文档就是 Json Schema 的一个 instance（实例）。

Json Schema 可以解决有关一致性验证的问题，例如：

- 1、值的数据类型是否正确：可以具体规定一个值是数字、字符串等类型；
- 15 2、是否包含所需的数据：可以规定哪些数据是需要的，哪些是不需要的；
- 3、值的形式是不是我需要的：可以指定范围、最小值和最大值。

如下信息，是 Json Schema 描述的一种示例：

```
{  
  "materials":{  
    "category":[  
      {  
        "categoryId":"a",  
        "name":"分类 1",  
        "content":[  
          "identify":"componentid_a","icon":"","desc":"这是一个测试图标 a"},  
          "identify":"componentid_b","icon":"","desc":"这是一个测试图标 b"}  
        ]  
      },  
      {
```

20

25

30

```
        "categoryId":"b",
        "name":"自定义分类",
        "content":[
            "identify":"componentid_c","icon":"","desc":"这是一个测试图标 a"}
    ]
}
],
},
10  "attributes":{
    "category":[
        {
            "attributeId":"attribute_a",
            "identify":"componentid_a",
15  "categoryType":"",
            "name":"属性分类 1",
            "content":[
                {"type":"select"}
            ]
20  },
        {
            "categoryType":"style",
            "name":"属性分类 2",
            "content":[
25  {"identify":"componentid_b","type":"string"},
                {"identify":"componentid_c","type":"boolean"}
            ]
        }
    ]
},
30  "components":[
    {
        "identify":"componentid_a",
```



```
"bundle": "https://gaic.alicdn.com/tms/jsscript_a"  
}  
]
```

5 }

步骤 103，依据所述结构化描述信息对所述功能组件的编码信息进行配置，以获得目标组件；

在本申请实施例中，通过结构化描述信息对功能组件的编码信息进行配置可以在解析引擎中实现。

10 以结构化描述信息为 Json Schema 描述信息为例，上述解析引擎即是可以对 Json Schema 协议进行解析的一种装置。通过解析引擎，可以将 Json 文档最终解析为应用功能的代码实现。

在具体实现中，可以首先提供一个注册函数，来把功能组件的代码信息注册到解析引擎中。

15 例如，engine.register({input: ComInput})，其中 input 为定义的功能组件的类型(type)，comInput 为具体的功能组件。

只有注册到解析引擎中的功能组件，才能在 Json Schema 中完成配置，并且最终通过解析引擎才能解析并渲染出该功能组件，从而得到需要的目标组件。

步骤 104，获取待加载的数据；

20 通常，一个功能组件仅有样式等信息是不够的，还需要有具体的行为或者数据来支持。例如，一个图表模块，需要去拉取在该图标中进行显示的数据；或者，该图表模块可能还有一些交互行为，如点击的时候需要响应等等。这就需要具体的数据去支持。

在本申请实施例中，在获取待加载的数据前，可以首先确定在 Json Schema 中配置数据的规则、相关的定制逻辑等等，然后按照上述规则或逻辑去拉取相应的数据。

25 步骤 105，针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序。

在本申请实施例中，在完成前述各个步骤后，为了实现将待开发的应用程序应用在不同的应用端中，可以利用解析引擎，将目标组件和待加载的数据解析为不同应用端的可执行程序，从而实现一套业务逻辑在多个应用端的复用。

30 在本申请实施例中，通过确定应用程序的各个功能组件，并生成与各个功能组件相

对应的结构化描述信息，从而可以依据上述结构化描述信息对功能组件的编码信息进行配置获得目标组件，然后在获取待加载的数据后，可以针对多个不同的应用端，分别将上述目标组件和待加载的数据编译为能够在不同的应用端上运行的可执行程序。本实施例可以在不同的应用端使同一套代码编写方式，然后通过构建翻译成不同的应用端支持的可执行程序，无需针对不同的应用端分别进行开发，实现了一套业务逻辑的在多个应用端的复用，提高了应用程序的开发效率，降低了应用程序的开发成本。

参照图 2，示出了本申请一个实施例的另一种应用程序的开发方法的步骤流程示意图，具体可以包括如下步骤：

步骤 201，确定应用程序的功能组件，所述功能组件具有相应的编码信息；

需要说明的是，上述应用程序可以是应用于多个应用端的程序或软件。例如，Web 端、Html5 端、Native 端等等。

在本申请实施例中，开发人员进行应用程序的开发前，可以首先确定该应用程序所具有的功能组件包括哪些。该应用程序具有的各种功能均可以通过各个功能组件来实现。

在本申请实施例中，应用程序各个功能组件的编码信息可以由开发人员使用 React Component 编写得到的功能组件的代码。

步骤 202，确定各个功能组件的逻辑实现信息；

在本申请实施例中，各个功能组件的逻辑实现信息可以是指该功能组件的具体功能实现，逻辑实现常常以软件编程的形式得以体现。

步骤 203，将所述逻辑实现信息转换为结构化描述信息；

在本申请实施例中，上述结构化描述信息可以是 Json Schema 描述信息，Json Schema 描述信息可以是一种 Json Schema 协议。

为了实现一次开发便能够最大限度地在多个不同的应用端上应用，可以在确定出应用程序的各个功能组件的逻辑实现后，以一种 Json Schema 协议去描述各个功能组件。

步骤 204，加载所述功能组件的编码信息至解析引擎；

在本申请实施例中，通过结构化描述信息对功能组件的编码信息进行配置可以在解析引擎中实现。

例如，在结构化描述信息为 Json Schema 描述信息时，上述解析引擎即是可以对 Json Schema 协议进行解析的一种装置，如，React Engine（一种基于 React 的解析引擎）。通过 React Engine，可以将 Json 文档最终解析为应用功能的代码实现。

在具体实现中，可以首先调用预设的注册函数，通过该注册函数，可以将功能组件的编码信息注册至 React Engine 解析引擎。

步骤 205，按照所述结构化描述信息，在所述解析引擎中对所述功能组件的编码信息进行渲染，以获得目标组件；

5 在本申请实施例中，只有注册到 React Engine 中的功能组件，才能在 Json Schema 中完成配置，并且最终通过 React Engine 才能解析并渲染出该功能组件，从而得到需要的目标组件。

因此，在将功能组件的代码注册到 React Engine 后，可以按照 Json Schema 协议，在 React Engine 中对上述代码进行配置和渲染，得到所需的目标组件。

10 步骤 206，确定各个功能组件的数据配置规则；

通常，一个功能组件仅有样式等信息是不够的，还需要有具体的行为或者数据来支持。例如，一个图表模块，需要去拉取在该图标中进行显示的数据；或者，该图表模块可能还有一些交互行为，如点击的时候需要响应等等。这就需要具体的数据去支持。

15 在本申请实施例中，在获取相应的数据前，可以首先确定在 Json Schema 中配置数据的规则、相关的定制逻辑等等。

步骤 207，按照所述数据配置规则，获取相应的待加载数据；

在确定出在 Json Schema 中配置数据的规则后，便可以按照这个规则去拉取相应的数据。这些数据即是在后续的开发过程中，需要用到的待加载数据，通过将这些数据加载至目标组件中，可以使得形成的各个目标组件具备了具体的行为和数据支撑。

20 步骤 208，确定目标应用端，所述目标应用端具有对应的编译方式；

在本申请实施例中，目标应用端可以是指当前开发的应用程序所要运行的应用端。例如，目标应用端可以是 Web 端、Html5 端、Native 端，以及，其他类型的应用端。

通常，不同的应用端具有不同的软件开发编译方式。因此，在确定出目标应用端后，可以进一步确定与该目标应用端相匹配的编译方式是什么，从而可以按照对应的编译方式将前述步骤获得的目标组件和待加载的数据编译为适配于该目标应用端的可执行程序。

25 步骤 209，按照所述编译方式，将所述目标组件和所述待加载的数据编译为所述目标应用端的可执行程序；

在本申请实施例中，可以按照各个应用端对应的编译方式，通过构建方式，例如 babel 脚本，将一套代码编译成可以在不同应用端运行的可执行程序。

Babel 是一种用于编写下一代 JavaScript 的编译器，能够把用最新标准编写的 JavaScript 代码向下编译成可以在今天随处可用的版本。这一过程通常被叫做“源码到源码”编译，也被称为转换编译。

通过上述编译过程，使得程序能够匹配不同的应用端，实现一次开发便能够在多个应用端上的复用。

步骤 210，在所述目标应用端中运行所述可执行程序。

在本申请实施例中，在将代码编译为对应目标应用端上的可执行程序后，便可以在该目标应用端上运行该程序，从而完成在多个应用端上的应用程序开发。

需要说明的是，在完成对应用程序的开发后，还可以根据应用程序在后续的使用过程中的实际需要，对应用程序的功能组件进行更新或修改，动态地替换掉已经上线的功能组件的各个内容，例如，如更改图标、字段、颜色等等，实现对应用程序的内容的随时更新。

在本申请实施例中，由于各个应用端共用一份结构化描述信息，因此在需要对应用程序进行更新后修改时，可以首先确定待更新或修改的功能组件是哪一个，然后对上述待更新或修改的功能组件对应的结构化描述信息进行更新或修改，实现一次修改多端同步，对应用程序内容快速更新。

在本申请实施例中，通过将应用程序的设计抽象为 Json Schema 描述，这样 PC 端、Mobile 端等各个不同的应用端都可以使用同一份 Schema 描述来生成功能组件；同时，整个应用程序都是 Json 数据格式的，每一次操作都是一份新的 Json 文档，所有过程都是可逆的；业务逻辑通过相应的配置规则实现，不会侵入功能组件内部，降低了组件的使用难度，无需针对不同的应用端分别进行开发，实现了一套业务逻辑的在多个应用端的复用，提高了应用程序的开发效率，降低了应用程序的开发成本。

参照图 3，示出了本申请一个实施例的又一种应用程序的开发方法的步骤流程示意图，具体可以包括如下步骤：

步骤 301，确定应用程序的功能组件，所述功能组件具有相应的编码信息和逻辑实现信息；

步骤 302，将所述功能组件的逻辑实现信息转换为 Json Schema 描述信息；

步骤 303，加载所述功能组件的编码信息至解析引擎；

步骤 304，依据所述 Json Schema 描述信息，在所述解析引擎中对所述功能组件的编码信息进行配置，以获得目标组件；

步骤 305, 获取待加载的数据;

步骤 306, 针对多个应用端, 分别将所述目标组件和所述待加载的数据编译为可执行程序;

步骤 307, 分别在各个应用端中运行所述可执行程序。

5 在本申请实施例中, 开发人员在进行应用程序的开发时, 可以首先确定该应用程序所要运行的应用端包括哪些。

如图 4 所示, 是本申请一个实施例的一种应用程序的开发方法的原理图。在图 4 中, 当前开发的应用程序可以是需要运行于 Web 端、Html5 端, 以及, Native 端中。

10 然后, 可以进一步确定该应用程序所具有的功能组件包括哪些, 以及各个功能组件的逻辑实现信息。通过这些功能组件能够实现该应用程序的完整功能。

在确定出功能组件的逻辑实现信息后, 可以将上述逻辑实现信息抽象为 Json Schema 描述, 如图 4 中所示的 Component (组件)、Layout (布局)、Data (数据)、Rules (配置规则)。

15 进而, 将通过使用 React Component 编写得到的各个功能组件的代码加载至能够解析 Json Schema 的 React Engine 解析引擎中, 并在 React Engine 进行解析、配置和渲染, 得到目标组件。

20 在获取到待加载的数据后, 可以针对各个应用端, 将目标组件编译为对应应用端的可执行程序, 并在该应用端中进行运行。例如, 在图 4 中, 对应于 Web 端, 可以编译得到适配于该 Web 端的 Web Page (网页页面); 对应于 Html5 端, 可以编译得到适配于该 Html5 端的 Html5 APP (Html5 应用程序); 对应于 Native 端, 可以编译得到适配于该 Native 端的 Native APP (Native 应用程序)。

由于本实施例中各个步骤与前述实施例步骤 201-步骤 210 类似, 可以相互参阅, 本实施例对此描述得比较简单, 相关过程可以参见上述步骤 201-步骤 210 的描述, 本实施例对此不再赘述。

25 需要说明的是, 对于方法实施例, 为了简单描述, 故将其都表述为一系列的动作组合, 但是本领域技术人员应该知悉, 本申请实施例并不受所描述的动作顺序的限制, 因为依据本申请实施例, 某些步骤可以采用其他顺序或者同时进行。其次, 本领域技术人员也应该知悉, 说明书中所描述的实施例均属于优选实施例, 所涉及的动作并不一定是本申请实施例所必须的。

30 参照图 5, 示出了本申请一个实施例的一种应用程序的开发装置的示意性结构框图,

具体可以包括如下模块：

确定模块 501，用于确定应用程序的功能组件，所述功能组件具有相应的编码信息；

生成模块 502，用于生成与所述功能组件相对应的结构化描述信息；

配置模块 503，用于依据所述结构化描述信息对所述功能组件的编码信息进行配置，

5 以获得目标组件；

获取模块 504，用于获取待加载的数据；

编译模块 505，用于针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序。

在本申请实施例中，所述生成模块 502 具体可以包括如下子模块：

10 逻辑实现信息确定子模块，用于确定各个功能组件的逻辑实现信息；

逻辑实现信息转换子模块，用于将所述逻辑实现信息转换为结构化描述信息。

在本申请实施例中，所述结构化描述信息可以为 Json Schema 描述信息。

在本申请实施例中，所述配置模块 503 具体可以包括如下子模块：

编码信息加载子模块，用于加载所述功能组件的编码信息至解析引擎；

15 编码信息渲染子模块，用于按照所述结构化描述信息，在所述解析引擎中对所述功能组件的编码信息进行渲染，以获得目标组件。

在本申请实施例中，所述编码信息加载子模块具体可以包括如下单元：

注册函数调用单元，用于调用预设的注册函数；

20 编码信息注册单元，用于通过所述注册函数，将所述功能组件的编码信息注册至解析引擎。

在本申请实施例中，所述获取模块 504 具体可以包括如下子模块：

数据配置规则确定子模块，用于确定各个功能组件的数据配置规则；

待加载数据获取子模块，用于按照所述数据配置规则，获取相应的待加载数据。

在本申请实施例中，所述编译模块 505 具体可以包括如下子模块：

25 目标应用端确定子模块，用于确定目标应用端，所述目标应用端具有对应的编译方式；

可执行程序编译子模块，用于按照所述编译方式，将所述目标组件和所述待加载的数据编译为所述目标应用端的可执行程序。

在本申请实施例中，所述装置还可以包括如下模块：

30 运行模块，用于在所述目标应用端中运行所述可执行程序。

在本申请实施例中，所述装置还可以包括如下模块：

更新修改模块，用于对所述应用程序的功能组件进行更新或修改。

在本申请实施例中，所述更新修改模块具体可以包括如下子模块：

功能组件确定子模块，用于确定待更新或修改的功能组件；

- 5 结构化描述信息更新修改子模块，用于对所述待更新或修改的功能组件对应的结构化描述信息进行更新或修改。

参照图 6，示出了本申请一个实施例的另一种应用程序的开发装置的示意性结构框图，具体可以包括如下模块：

- 10 确定模块 601，用于确定应用程序的功能组件，所述功能组件具有相应的编码信息和逻辑实现信息；

转换模块 602，用于将所述功能组件的逻辑实现信息转换为 **Json Schema** 描述信息；

加载模块 603，用于加载所述功能组件的编码信息至解析引擎；

配置模块 604，用于依据所述 **Json Schema** 描述信息，在所述解析引擎中对所述功能组件的编码信息进行配置，以获得目标组件；

- 15 获取模块 605，用于获取待加载的数据；

编译模块 606，用于针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序；

运行模块 607，用于分别在各个应用端中运行所述可执行程序。

- 20 对于装置实施例而言，由于其与方法实施例基本相似，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

本说明书中的各个实施例均采用递进的方式描述，每个实施例重点说明的都是与其他实施例的不同之处，各个实施例之间相同相似的部分互相参见即可。

- 25 本领域内的技术人员应明白，本申请实施例的实施例可提供为方法、装置、或计算机程序产品。因此，本申请实施例可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请实施例可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、**CD-ROM**、光学存储器等)上实施的计算机程序产品的形式。

- 30 在一个典型的配置中，所述计算机设备包括一个或多个处理器 (**CPU**)、输入/输出接口、网络接口和内存。内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器(**RAM**) 和/或非易失性内存等形式，如只读存储器(**ROM**)或闪存(**flash RAM**)。内存

是计算机可读介质的示例。计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括非持续性的电脑可读媒体(transitory media)，如调制的数据信号和载波。

本申请实施例是参照根据本申请实施例的方法、终端设备(系统)、和计算机程序产品的流程图和 / 或方框图来描述的。应理解可由计算机程序指令实现流程图和 / 或方框图中的每一流程和 / 或方框、以及流程图和 / 或方框图中的流程和 / 或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理终端设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理终端设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理终端设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能。

这些计算机程序指令也可装载到计算机或其他可编程数据处理终端设备上，使得在计算机或其他可编程终端设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程终端设备上执行的指令提供用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的步骤。

尽管已描述了本申请实施例的优选实施例，但本领域内的技术人员一旦得知了基本创造性概念，则可对这些实施例做出另外的变更和修改。所以，所附权利要求意欲解释为包括优选实施例以及落入本申请实施例范围的所有变更和修改。

最后，还需要说明的是，在本文中，诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来，而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且，术语“包括”、“包含”或者其任何

其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者终端设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者终端设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、物品或者终端设备中还存在另外的相同要素。

以上对本申请所提供的一种应用程序的开发方法和一种应用程序的开发装置，进行了详细介绍，本文中应用了具体个例对本申请的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本申请的方法及其核心思想；同时，对于本领域的一般技术人员，依据本申请的思想，在具体实施方式及应用范围上均会有改变之处，综上所述，本说明书内容不应理解为对本申请的限制。

权 利 要 求 书

1.一种应用程序的开发方法，其特征在于，包括：

确定应用程序的功能组件，所述功能组件具有相应的编码信息；

生成与所述功能组件相对应的结构化描述信息；

5 依据所述结构化描述信息对所述功能组件的编码信息进行配置，以获得目标组件；
获取待加载的数据；

针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序。

2.根据权利要求1所述的方法，其特征在于，所述生成与所述功能组件相对应的结构化描述信息的步骤包括：

10 确定各个功能组件的逻辑实现信息；

将所述逻辑实现信息转换为结构化描述信息。

3.根据权利要求2所述的方法，其特征在于，所述结构化描述信息为 Json Schema 描述信息。

4.根据权利要求1所述的方法，其特征在于，所述依据所述结构化描述信息对所述
15 功能组件的编码信息进行配置，以获得目标组件的步骤包括：

加载所述功能组件的编码信息至解析引擎；

按照所述结构化描述信息，在所述解析引擎中对所述功能组件的编码信息进行渲染，以获得目标组件。

5.根据权利要求4所述的方法，其特征在于，所述加载所述功能组件的编码信息至
20 解析引擎的步骤包括：

调用预设的注册函数；

通过所述注册函数，将所述功能组件的编码信息注册至解析引擎。

6.根据权利要求1所述的方法，其特征在于，所述获取待加载的数据的步骤包括：

确定各个功能组件的数据配置规则；

25 按照所述数据配置规则，获取相应的待加载数据。

7.根据权利要求1所述的方法，其特征在于，所述针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序的步骤包括：

确定目标应用端，所述目标应用端具有对应的编译方式；

30 按照所述编译方式，将所述目标组件和所述待加载的数据编译为所述目标应用端的
可执行程序。

8.根据权利要求 7 所述的方法，其特征在于，还包括：

在所述目标应用端中运行所述可执行程序。

9.根据权利要求 1 所述的方法，其特征在于，还包括：

对所述应用程序的功能组件进行更新或修改。

5 10.根据权利要求 9 所述的方法，其特征在于，所述对所述应用程序的功能组件进行更新或修改的步骤包括：

确定待更新或修改的功能组件；

对所述待更新或修改的功能组件对应的结构化描述信息进行更新或修改。

10 11.一种应用程序的开发方法，其特征在于，包括：

确定应用程序的功能组件，所述功能组件具有相应的编码信息和逻辑实现信息；

将所述功能组件的逻辑实现信息转换为 Json Schema 描述信息；

加载所述功能组件的编码信息至解析引擎；

15 依据所述 Json Schema 描述信息，在所述解析引擎中对所述功能组件的编码信息进行配置，以获得目标组件；

获取待加载的数据；

针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序；

分别在各个应用端中运行所述可执行程序。

20 12.一种应用程序的开发装置，其特征在于，包括：

确定模块，用于确定应用程序的功能组件，所述功能组件具有相应的编码信息；

生成模块，用于生成与所述功能组件相对应的结构化描述信息；

配置模块，用于依据所述结构化描述信息对所述功能组件的编码信息进行配置，以获得目标组件；

25 获取模块，用于获取待加载的数据；

编译模块，用于针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序。

13.一种应用程序的开发装置，其特征在于，包括：

30 确定模块，用于确定应用程序的功能组件，所述功能组件具有相应的编码信息和逻辑实现信息；

辑实现信息；

转换模块，用于将所述功能组件的逻辑实现信息转换为 Json Schema 描述信息；

加载模块，用于加载所述功能组件的编码信息至解析引擎；

5 配置模块，用于依据所述 Json Schema 描述信息，在所述解析引擎中对所述功能组件的编码信息进行配置，以获得目标组件；

获取模块，用于获取待加载的数据；

编译模块，用于针对多个应用端，分别将所述目标组件和所述待加载的数据编译为可执行程序；

运行模块，用于分别在各个应用端中运行所述可执行程序。

10

15

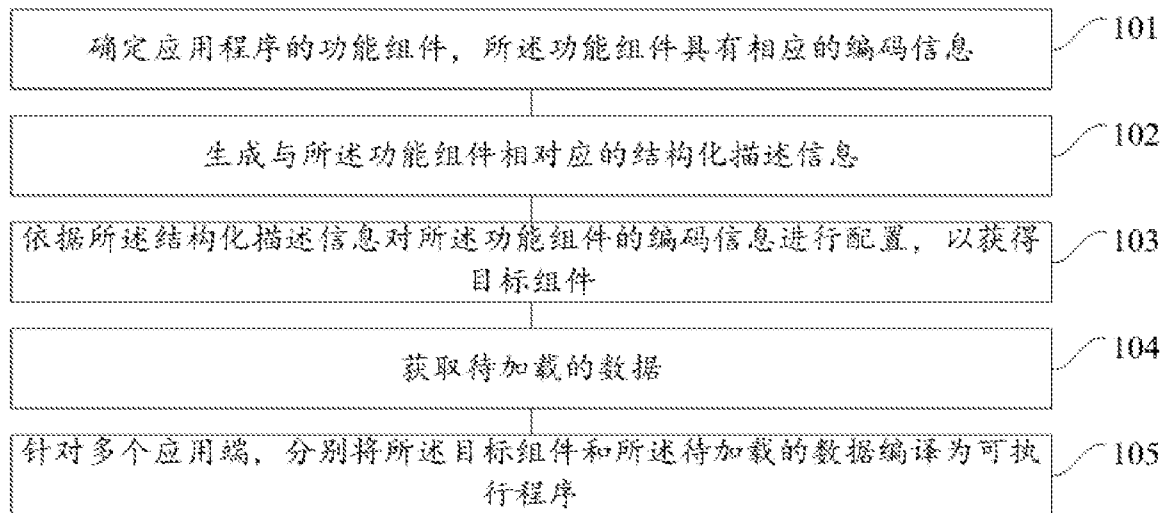


图 1



图 2

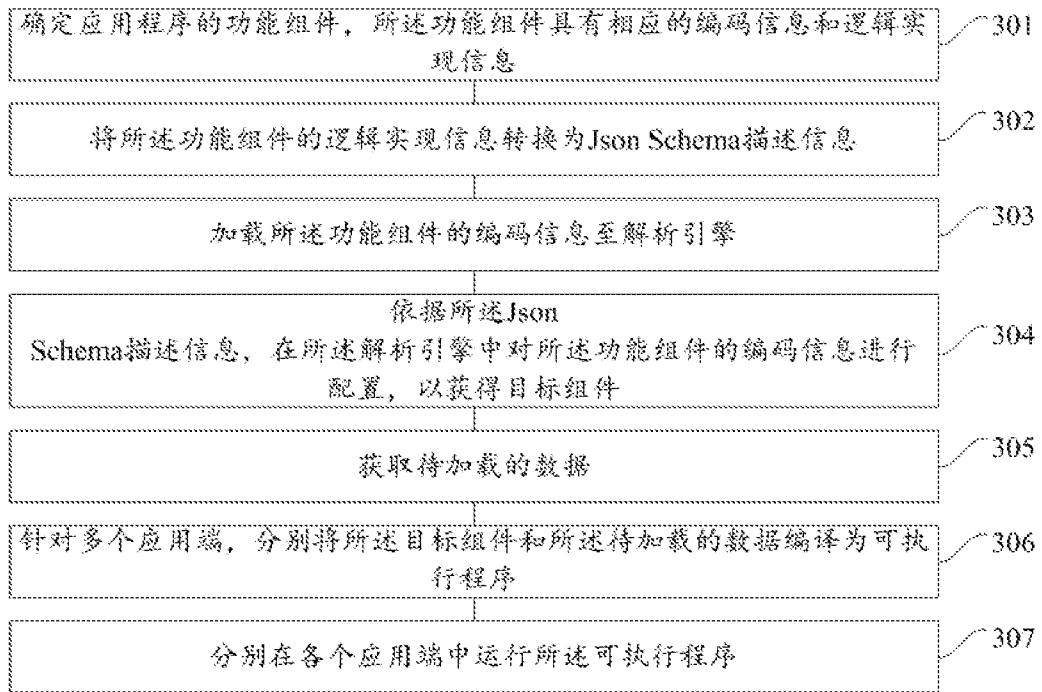


图 3

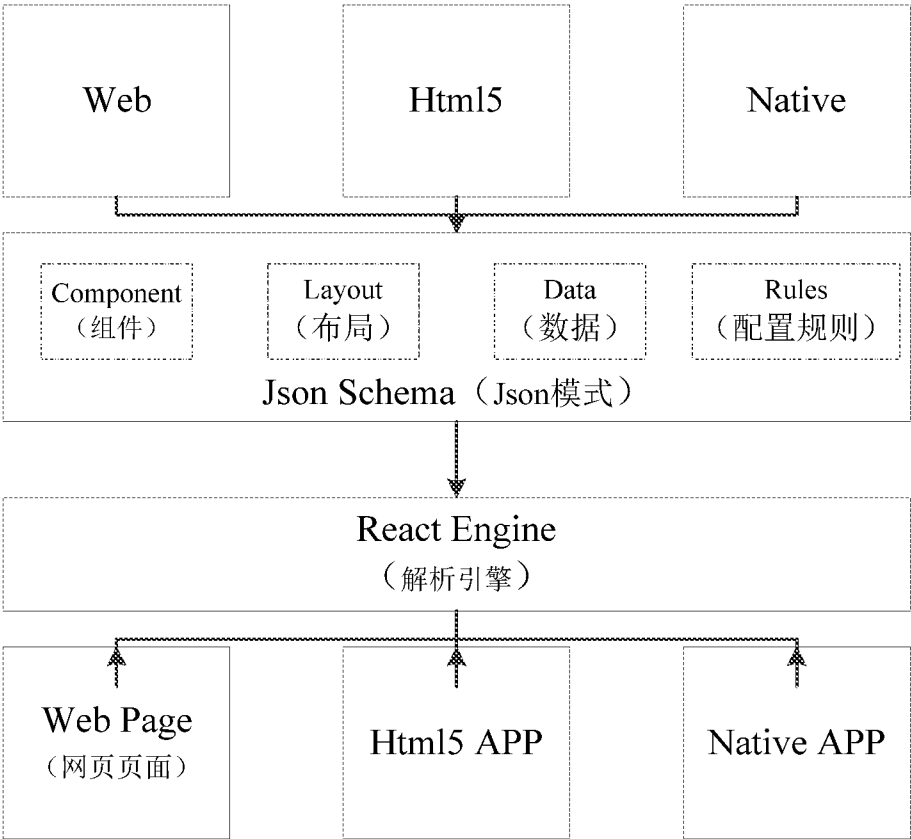


图 4



图 5

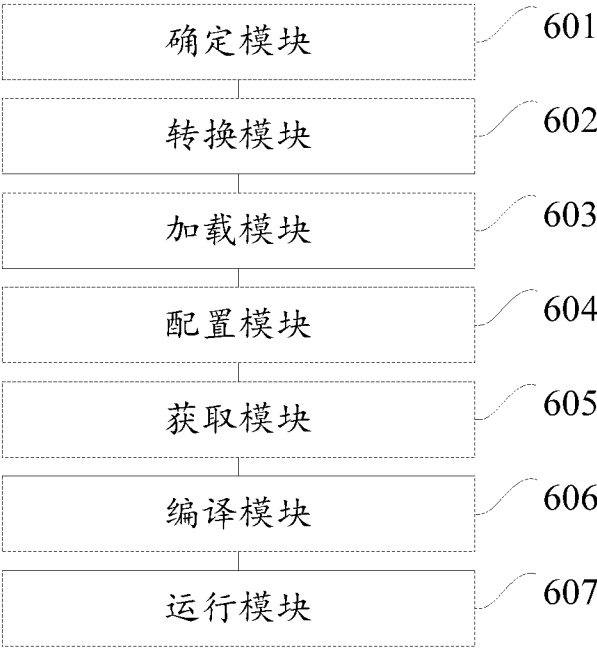


图 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/088258

A. CLASSIFICATION OF SUBJECT MATTER

G06F 8/20(2018.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, WPI, EPODOC, IEEE: 应用, 软件, 开发, 组件, 功能, 编码, 代码, 描述, json, schema, 配置, 多个, 一套, 平台, 一次, 编译, application, software, development, component, function, code, description, configuration, multiple, platform, compile

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 106951231 A (WEIMENG CHUANGKE NETWORK SCIENCE & TECHNOLOGY (CHINA) CO., LTD.) 14 July 2017 (2017-07-14) description, paragraphs [0032]-[0080]	1-13
A	CN 106775725 A (NETEASE HANGZHOU NETWORK CO., LTD.) 31 May 2017 (2017-05-31) entire document	1-13
A	CN 107908432 A (SHENZHEN QUANYAN SPORTS OPERATIONS MANAGEMENT CO., LTD.) 13 April 2018 (2018-04-13) entire document	1-13
A	US 2017161121 A1 (STRONGLOOP, INC.) 08 June 2017 (2017-06-08) entire document	1-13

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

02 August 2019

Date of mailing of the international search report

26 August 2019

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2019/088258

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	106951231	A	14 July 2017	None			
CN	106775725	A	31 May 2017	None			
CN	107908432	A	13 April 2018	None			
US	2017161121	A1	08 June 2017	US	2017192803	A1	06 July 2017
				US	2016077902	A1	17 March 2016
				US	2018107501	A1	19 April 2018
				US	2016077901	A1	17 March 2016
				US	2016077853	A1	17 March 2016
				US	2016080493	A1	17 March 2016
				US	2016077897	A1	17 March 2016

国际检索报告

国际申请号

PCT/CN2019/088258

A. 主题的分类

G06F 8/20(2018.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNPAT, CNKI, WPI, EPODOC, IEEE: 应用, 软件, 开发, 组件, 功能, 编码, 代码, 描述, json, schema, 配置, 多个, 一套, 平台, 一次, 编译, application, software, development, component, function, code, description, configuration, multiple, platform, compile

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
A	CN 106951231 A (微梦创科网络科技中国有限公司) 2017年 7月 14日 (2017 - 07 - 14) 说明书第[0032]-[0080]段	1-13
A	CN 106775725 A (网易杭州网络有限公司) 2017年 5月 31日 (2017 - 05 - 31) 全文	1-13
A	CN 107908432 A (深圳泉眼体育运营管理有限公司) 2018年 4月 13日 (2018 - 04 - 13) 全文	1-13
A	US 2017161121 A1 (STRONGL00P, INC) 2017年 6月 8日 (2017 - 06 - 08) 全文	1-13

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2019年 8月 2日

国际检索报告邮寄日期

2019年 8月 26日

ISA/CN的名称和邮寄地址

中国国家知识产权局 (ISA/CN)
中国北京市海淀区蓟门桥西土城路6号 100088

受权官员

刘剑

传真号 (86-10)62019451

电话号码 86-(10)-53961304

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2019/088258

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	106951231	A	2017年 7月 14日	无			
CN	106775725	A	2017年 5月 31日	无			
CN	107908432	A	2018年 4月 13日	无			
US	2017161121	A1	2017年 6月 8日	US	2017192803	A1	2017年 7月 6日
				US	2016077902	A1	2016年 3月 17日
				US	2018107501	A1	2018年 4月 19日
				US	2016077901	A1	2016年 3月 17日
				US	2016077853	A1	2016年 3月 17日
				US	2016080493	A1	2016年 3月 17日
				US	2016077897	A1	2016年 3月 17日