



(12)发明专利

(10)授权公告号 CN 108388622 B

(45)授权公告日 2020.02.28

(21)申请号 201810144185.1

G06F 16/25(2019.01)

(22)申请日 2018.02.12

(56)对比文件

(65)同一申请的已公布的文献号

CN 102622227 A, 2012.08.01, 全文.

申请公布号 CN 108388622 A

审查员 郑舒玲

(43)申请公布日 2018.08.10

(73)专利权人 平安科技(深圳)有限公司

地址 518000 广东省深圳市福田区八卦岭

工业区平安大厦六楼

(72)发明人 林水明

(74)专利代理机构 深圳市精英专利事务所

44242

代理人 林燕云

(51)Int.Cl.

G06F 16/21(2019.01)

G06F 16/242(2019.01)

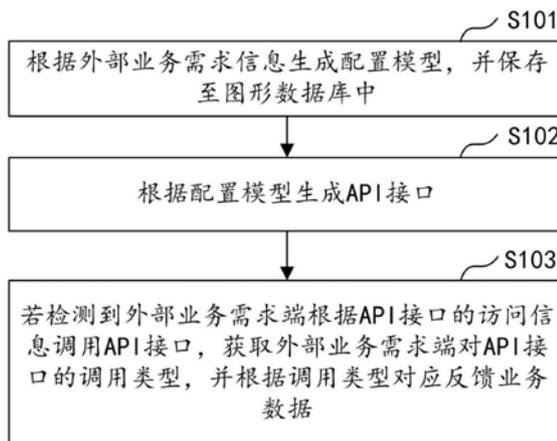
权利要求书2页 说明书11页 附图4页

(54)发明名称

API接口动态生成方法、装置、计算机设备及存储介质

(57)摘要

本申请实施例公开了一种API接口动态生成方法、装置、计算机设备及存储介质。该方法包括：根据外部业务需求信息生成配置模型，并保存至图形数据库中；根据配置模型生成API接口；若检测到外部业务需求端根据API接口的访问信息调用API接口，获取外部业务需求端对API接口的调用类型，并根据调用类型对应反馈业务数据。该方法无需编写大量的SQL语句生成API接口，只需快速的修改配置模型即可得到满足需求的API接口，节省了开发大量SQL语句所耗用的时间，也避免了接口变更导致应用起停的问题。



1. 一种API接口动态生成方法,其特征在于,包括:
  - 根据外部业务需求信息生成配置模型,并保存至图形数据库中;
  - 根据配置模型生成API接口;
  - 若检测到外部业务需求端根据API接口的访问信息调用API接口,获取外部业务需求端对API接口的调用类型,并根据调用类型对应反馈业务数据;
  - 解析外部业务需求信息中所包括的业务视图层节点的名称、与业务视图层节点相关联的实体的名称,实体之间的联系、及每一实体所对应关系型数据库中的业务数据;
  - 根据业务视图层节点的名称,对应生成配置模型中的业务视图层节点;
  - 根据与业务视图层节点相关联的实体的名称,对应生成配置模型中的实体;
  - 根据业务视图层节点和与业务视图层节点相关联的实体之间的联系对应生成第一关系连接线组,并通过第一关系连接线组将配置模型中的业务视图层节点与实体进行连接;
  - 根据实体之间的联系对应生成第二关系连接线组,并通过第二关系连接线组将配置模型中的实体与相关联实体进行连接。
2. 根据权利要求1所述的API接口动态生成方法,其特征在于,所述根据配置模型生成API接口,包括:
  - 根据配置模型中的业务视图层节点的名称创建API接口;
  - 将API接口与配置模型中的业务视图层节点进行映射。
3. 根据权利要求2所述的API接口动态生成方法,其特征在于,所述根据配置模型生成API接口之后,包括:
  - 将API接口的访问信息发送至外部业务需求端;其中访问信息包括请求格式、请求参数、请求地址。
4. 根据权利要求3所述的API接口动态生成方法,其特征在于,所述获取外部业务需求端对API接口的调用类型,并根据调用类型对应反馈业务数据,包括:
  - 根据外部业务需求端的访问信息中所包括的请求参数,获取外部业务需求端所访问API接口对应的业务视图层节点;
  - 根据业务视图层的表名、及对API接口的调用类型对应生成SQL脚本;其中所述调用类型包括增加数据请求、删除数据请求、编辑数据请求及查询数据请求;
  - 执行SQL脚本从关系型数据库中获取业务数据,并对应反馈至API接口。
5. 一种API接口动态生成装置,其特征在于,包括:
  - 配置模型生成单元,用于根据外部业务需求信息生成配置模型,并保存至图形数据库中;
  - API接口生成单元,用于根据配置模型生成API接口;
  - 业务数据反馈单元,用于若检测到外部业务需求端根据API接口的访问信息调用API接口,获取外部业务需求端对API接口的调用类型,并根据调用类型对应反馈业务数据;
  - 解析单元,用于解析外部业务需求信息中所包括的业务视图层节点的名称、与业务视图层节点相关联的实体的名称,实体之间的联系、及每一实体所对应关系型数据库中的业务数据;
  - 节点生成单元,用于根据业务视图层节点的名称,对应生成配置模型中的业务视图层节点;

实体生成单元,用于根据与业务视图层节点相关联的实体的名称,对应生成配置模型中的实体;

第一连线单元,用于根据业务视图层节点和与业务视图层节点相关联的实体之间的联系对应生成第一关系连接线组,并通过第一关系连接线组将配置模型中的业务视图层节点与实体进行连接;

第二连线单元,用于根据实体之间的联系对应生成第二关系连接线组,并通过第二关系连接线组将配置模型中的实体与相关联实体进行连接。

6. 根据权利要求5所述的API接口动态生成装置,其特征在于,所述API接口生成单元,包括:

创建单元,用于根据配置模型中的业务视图层节点的名称创建API接口;

映射单元,用于将API接口与配置模型中的业务视图层节点进行映射。

7. 一种计算机设备,包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1至4中任一项所述的API接口动态生成方法。

8. 一种存储介质,其特征在于,所述存储介质存储有计算机程序,所述计算机程序包括程序指令,所述程序指令当被处理器执行时使所述处理器执行如权利要求1-4任一项所述的API接口动态生成方法。

## API接口动态生成方法、装置、计算机设备及存储介质

### 技术领域

[0001] 本申请涉及数据库配置管理技术领域,尤其涉及一种API接口动态生成方法、装置、计算机设备及存储介质。

### 背景技术

[0002] CMDB(Configuration Management Database,即配置管理数据库)作为一个互联网企业的所有配置核心,对外的数据交互极多。为了保证数据的准确性以及安全性,系统的对外交互主要是经过对外的API接口来提供外部系统的数据交互。目前常用的API接口存在以下不足之处:

[0003] 1.大量SQL的编写;

[0004] 2.业务变更,接口也要对应修改,并需要起停应用发版。

### 发明内容

[0005] 本申请提供了一种API接口动态生成方法、装置、计算机设备及存储介质,旨在解决现有技术中CMDB系统的对外API接口生成需编写大量的SQL语句,且当业务变更时对外API接口也要修改,导致开发CMDB系统的对外API接口极为耗时的问题。

[0006] 第一方面,本申请提供了一种API接口动态生成方法,其包括:

[0007] 根据外部业务需求信息生成配置模型,并保存至图形数据库中;

[0008] 根据配置模型生成API接口;

[0009] 若检测到外部业务需求端根据API接口的访问信息调用API接口,获取外部业务需求端对API接口的调用类型,并根据调用类型对应反馈业务数据。

[0010] 第二方面,本申请提供了一种API接口动态生成装置,其包括:

[0011] 配置模型生成单元,用于根据外部业务需求信息生成配置模型,并保存至图形数据库中;

[0012] API接口生成单元,用于根据配置模型生成API接口;

[0013] 业务数据反馈单元,用于若检测到外部业务需求端根据API接口的访问信息调用API接口,获取外部业务需求端对API接口的调用类型,并根据调用类型对应反馈业务数据。

[0014] 第三方面,本申请又提供了一种计算机设备,包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现本申请提供的任一项所述的API接口动态生成方法。

[0015] 第四方面,本申请还提供了一种存储介质,其中所述存储介质存储有计算机程序,所述计算机程序包括程序指令,所述程序指令当被处理器执行时使所述处理器执行本申请提供的任一项所述的API接口动态生成方法。

[0016] 本申请提供一种API接口动态生成方法、装置、计算机设备及存储介质。该方法通过根据外部业务需求信息生成配置模型,并保存至图形数据库中;根据配置模型生成API接口;若检测到外部业务需求端根据API接口的访问信息调用API接口,获取外部业务需求端

对API接口的调用类型,并根据调用类型对应反馈业务数据。该方法无需编写大量的SQL语句生成API接口,只需快速的修改配置模型即可得到满足需求的API接口,节省了开发大量SQL语句所耗用的时间,也避免了接口变更导致应用起停的问题。

### 附图说明

[0017] 为了更清楚地说明本申请实施例技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0018] 图1a为本申请实施例中一具体图形数据库的三层结构示意图;

[0019] 图1b为本申请实施例提供的一种API接口动态生成方法的示意流程图;

[0020] 图2是本申请实施例提供的一种API接口动态生成方法的子流程示意图;

[0021] 图3是本申请实施例提供的一种API接口动态生成方法的另一子流程示意图;

[0022] 图4为本申请实施例提供的一种API接口动态生成方法的另一子流程示意流程图;

[0023] 图5为本申请实施例提供的一种API接口动态生成装置的示意性框图;

[0024] 图6为本申请实施例提供的一种API接口动态生成装置的子单元示意性框图;

[0025] 图7为本申请实施例提供的一种API接口动态生成装置的另一子单元示意性框图;

[0026] 图8为本申请实施例提供的一种API接口动态生成装置的另一子单元示意性框图;

[0027] 图9为本申请实施例提供的一种计算机设备的示意性框图。

### 具体实施方式

[0028] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0029] 应当理解,当在本说明书和所附权利要求书中使用时,术语“包括”和“包含”指示所描述特征、整体、步骤、操作、元素和/或组件的存在,但并不排除一个或多个其它特征、整体、步骤、操作、元素、组件和/或其集合的存在或添加。

[0030] 还应当理解,在此本申请说明书中所使用的术语仅仅是出于描述特定实施例的目的而并不意在限制本申请。如在本申请说明书和所附权利要求书中所使用的那样,除非上下文清楚地指明其它情况,否则单数形式的“一”、“一个”及“该”意在包括复数形式。

[0031] 还应当进一步理解,在本申请说明书和所附权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合,并且包括这些组合。

[0032] 为了方便理解,下面对本发明实施例中涉及的图形数据库,及CDMB应用系统及关系型数据库进行介绍。

[0033] 在图形数据库中,存储有多个内核配置(内核配置也可称为配置模型)。配置模型是一种标准的图形结构,具体图形结构参考图1a。如图1a所示,所述配置模型被划分为三层,具体如下:

[0034] 最上面一层是页面层(ViewGroup),该层是对页面基础信息的抽象,主要包含页面的基础信息,例如:页面名称,页面编码等信息。

[0035] 中间一层是业务视图层 (View), 该层是对业务数据的抽象, 每个View在页面当中可以理解成一个数据表格, 主要包含业务视图的名称、编码、View类型以及View组件类型等信息, 每个业务视图View都是由ER层的一个或多个Entity (实体) 组成。

[0036] 其中, View类型主要包含:

[0037] a. 拓扑图类型, 主要用于展示数据关联视图;

[0038] b. 业务数据类型, 主要用于展示页面表格数据。

[0039] View的组件类型主要包含:

[0040] a. 数据表格;

[0041] b. 其他数据展示组件, 例如树状图、柱状图等其他数据展示组件。

[0042] 最下面一层是ER层, 主要是对关系型数据库E-R模型的表示, 用于在对业务视图层View所需业务数据的管理提供基础, 在ER层中主要包含Entity/Attribute (其中, Entity表示实体, Attribute表示实体的属性) 以及Entity (实体) 与Entity (实体) 的关系, 以及Entity与其Attribute (属性) 的关系。

[0043] 其中, Entity主要包含表名、表别名等信息是否可见、是否可编辑等信息;

[0044] Attribute主要包含属性名称、编码、是否可见、是否可编辑等信息。

[0045] 在了解图形数据库的基本知识后, 下面对CMDB应用系统进行简要介绍。CMDB应用系统分为三层, 其中面向用户且能被用户直接看到的是页面, 页面的下一层是CMDB应用系统层, 再下一层是后台的图形数据库及关系型数据库 (图形数据库、及关系型数据库均与CMDB应用系统层通讯连接, 也就是CMDB应用系统层能从图形数据库、及关系型数据库获取数据)。

[0046] 其中, 图形数据库包括多个配置模型, 每一配置模型均由上述三层结构组成 (即页面层、业务视图层、和ER层), 每个配置模型中的业务视图层均包括至少一个业务视图层节点, 并在业务视图层进行显示。

[0047] 请参阅图1b, 图1b是本申请实施例提供的一种API接口动态生成方法的示意流程图。该方法应用于台式电脑、手提电脑、平板电脑等终端中。如图1b所示, 该方法包括步骤S101~S103。

[0048] S101、根据外部业务需求信息生成配置模型, 并保存至图形数据库中。

[0049] 在本实施例中, 在获取外部业务需求信息时, 是通过各种获取信息的方式均可行 (如接收邮件、QQ消息、微信、短信等)。例如, 外部业务需求端需要获取机房相关信息的数据, 则需要通过服务端 (服务端也即CMDB应用系统) 提供一个能查询机房信息的API接口, 此时由外部业务需求端向服务端所发送的外部业务需求信息包括机房名称、位置、机房里面有哪些主机等信息; 该外部业务需求信息通过邮件的方式来发送至CMDB应用系统。

[0050] 如图2所示, 该步骤S101包括以下子步骤:

[0051] S1011、解析外部业务需求信息中所包括的业务视图层节点的名称、与业务视图层节点相关联的实体的名称, 实体之间的联系、及每一实体所对应关系型数据库中的业务数据;

[0052] S1012、根据业务视图层节点的名称, 对应生成配置模型中的业务视图层节点;

[0053] S1013、根据与业务视图层节点相关联的实体的名称, 对应生成配置模型中的实体;

[0054] S1014、根据业务视图层节点和与业务视图层节点相关联的实体之间的联系对应生成第一关系连接线组,并通过第一关系连接线组将配置模型中的业务视图层节点与实体进行连接;

[0055] S1015、根据实体之间的联系对应生成第二关系连接线组,并通过第二关系连接线组将配置模型中的实体与相关联实体进行连接。

[0056] 即,首先解析外部业务需求信息中的以下关键信息:业务视图层节点的名称,与业务视图层相关联的实体的名称,实体之间的相互关系,每一实体所对应关系型数据库中的业务数据;

[0057] 然后,根据业务视图层节点的名称对应生成配置模型中的业务视图层节点,根据实体的名称生成配置模型中的实体,根据实体与业务视图层节点之间的关系在配置模型中生成对应的第一关系连接线组(即View层的节点与E-R层中所包括实体之间的关系连接线),根据实体与实体之间的关系在配置模型中生成对应第二关系连接线组(及E-R层实体相互之间的关系连接线);

[0058] 最后,将配置模型保存在图形数据库中。

[0059] 通过上述过程,实现了配置模型的自动生成,或是人工配置而成。所生成的配置模型被保存至图形数据库中,即可进一步对该配置模型映射处理,以开放对应的API接口供用户进行数据操作。

[0060] S102、根据配置模型生成API接口。

[0061] 在本实施例中,由于配置模型是存储在图形数据库中的图形模型,并能不直接对外开放而进行数据操作,故需将配置模型映射成为API接口,并向外部业务需求端发送API接口的访问信息,才能通过API接口访问与图形数据库所连接的关系型数据库。由于配置模型是根据用户的实际数据需求而配置,故能灵活调整以满足用户的数据需求。而且配置模型的配置操作简单,只需简单的添加业务视图层节点及其属性,ER层的实体,实体之间的联系,及实体与业务视图层节点之间的联系就能完成配置,无需编辑复杂的SQL语句。而且当外部业务需求信息发生变化时,重新快速建立配置模型即可重新生成满足要求的API接口,也避免了接口变更导致应用起停的问题。

[0062] 如图3所示,该步骤S102包括以下子步骤:

[0063] S1021、根据配置模型中的业务视图层节点的名称创建API接口;

[0064] S1022、将API接口与配置模型中的业务视图层节点进行映射。

[0065] 在一实施例中,所述步骤S102之后,还包括:将API接口的访问信息发送至外部业务需求端;其中访问信息包括请求格式、请求参数、请求地址。

[0066] 在本实施例中,先根据配置模型中的业务视图层节点的名称初始化创建API接口,并根据业务视图层节点的属性进一步完善API接口的访问信息,即API接口以业务视图层节点的名称来命名,并在API接口的属性栏中编辑包括请求格式、请求参数、请求地址的访问信息,之后将具有相同名称的API接口和业务视图层节点进行一一映射即可完成API接口的生成。完成API接口的生成后,将API接口的访问信息发送至外部业务需求端,外部业务需求端才能向服务端发出增删改查等业务请求,从而对应的对数据型数据库中的数据进行相应操作。

[0067] 例如,向外部业务需求端所发送的访问信息如下:

```

URL: http://localhost:8090/add/data
method:POST
{
[0068]     "apiName": "xx" ,
        "dataSet":{
            {
                "hostName": "xx" ,
                "hostIp": "xx"
            }, {
                "hostName": "xx" ,
[0069]     "hostIp": "xx"
            }
        }
    }

```

[0070] 其中URL表示API接口的请求地址,method:POST表示该API接口的数据请求方式为POST请求,dataSet表示该API接口的数据请求格式为{"hostName":"xx","hostIp":"xx"}, {"hostName":"xx","hostIp":"xx"},add/data表示对该API接口的数据调用类型为增加数据。

[0071] S103、若检测到外部业务需求端根据API接口的访问信息调用API接口,获取外部业务需求端对API接口的调用类型,并根据调用类型对应反馈业务数据。

[0072] 如图4所示,该步骤S103包括以下子步骤:

[0073] S1031、根据外部业务需求端的访问信息中所包括的请求参数,获取外部业务需求端所访问API接口对应的业务视图层节点;

[0074] S1032、根据业务视图层的表名、及对API接口的调用类型对应生成SQL脚本;其中所述调用类型包括增加数据请求、删除数据请求、编辑数据请求及查询数据请求;

[0075] S1033、执行SQL脚本从关系型数据库中获取业务数据,并对应反馈至API接口。

[0076] 在本实施例中,外部系统调用生成的API时,具有以下过程:

[0077] 先根据请求参数定位到业务视图层的View(即业务视图层节点);

[0078] 然后根据业务视图层的View生成对应的SQL脚本;

[0079] 最后在获取到SQL脚本后,在mybatis框架中执行,从关系型数据库中对得到结果集,并反馈操作。

[0080] 其中,业务调用类型具体包括增加数据请求、删除数据请求、编辑数据请求、查询数据请求;

[0081] 当判断调用类型为查询数据请求,则根据查询数据请求、及业务视图层的view及

查询的关键词生成第一SQL脚本(第一SQL脚本中包括select语句),执行第一SQL脚本在数据型数据库中查询得到对应的业务数据,并将业务数据通过业务视图层节点反馈至外部业务需求端;

[0082] 当判断调用类型为增加数据请求,则根据增加数据请求、及业务视图层的view及待增加的数据生成第二SQL脚本(第二SQL脚本中包括add语句),执行第二SQL脚本在数据型数据库中增加对应的业务数据,并将增加数据成功的消息反馈至外部业务需求端;

[0083] 当判断调用类型为删除数据请求,则根据删除数据请求、及业务视图层的view及待删除的数据生成第三SQL脚本(第三SQL脚本中包括delete语句),执行第三SQL脚本在数据型数据库中删除对应的业务数据,并将删除数据成功的消息反馈至外部业务需求端;

[0084] 当判断调用类型为编辑数据请求,则根据编辑数据请求、及业务视图层的view及待编辑的数据生成第四SQL脚本(第四SQL脚本中包括update语句),执行第四SQL脚本在数据型数据库中修改对应的业务数据,并将编辑数据成功的消息反馈至外部业务需求端。

[0085] 可见,该方法无需编写大量的SQL语句生成API接口,只需快速的修改配置模型即可得到满足需求的API接口,节省了开发大量SQL语句所耗用的时间,也避免了接口变更导致应用起停的问题。

[0086] 本申请实施例还提供一种API接口动态生成装置,该API接口动态生成装置用于执行前述任一项API接口动态生成方法。具体地,请参阅图5,图5是本申请实施例提供的一种API接口动态生成装置的示意性框图。API接口动态生成装置100可以安装于台式电脑、平板电脑、手提电脑、等终端中。

[0087] 如图5所示,API接口动态生成装置100包括配置模型生成单元101、API接口生成单元102、业务数据反馈单元103。

[0088] 配置模型生成单元101,用于根据外部业务需求信息生成配置模型,并保存至图形数据库中。

[0089] 在本实施例中,在获取外部业务需求信息时,是通过各种获取信息的方式均可行(如接收邮件、QQ消息、微信、短信等)。例如,外部业务需求端需要获取机房相关信息的数据,则需要通过服务端(服务端也即CMDB应用系统)提供一个能查询机房信息的API接口,此时由外部业务需求端向服务端所发送的外部业务需求信息包括机房名称、位置、机房里面有哪些主机等信息;该外部业务需求信息通过邮件的方式来发送至CMDB应用系统。

[0090] API接口生成单元102,用于根据配置模型生成API接口。

[0091] 在本实施例中,由于配置模型是存储在图形数据库中的图形模型,并能不直接对外开放而进行数据操作,故需将配置模型映射成为API接口,并向外部业务需求端发送API接口的访问信息,才能通过API接口访问与图形数据库所连接的关系型数据库。由于配置模型是根据用户的实际数据需求而配置,故能灵活调整以满足用户的数据需求。而且配置模型的配置操作简单,只需简单的添加业务视图层节点及其属性,ER层的实体,实体之间的联系,及实体与业务视图层节点之间的联系就能完成配置,无需编辑复杂的SQL语句。而且当外部业务需求信息发生变化时,重新快速建立配置模型即可重新生成满足要求的API接口,也避免了接口变更导致应用起停的问题。

[0092] 业务数据反馈单元103,用于若检测到外部业务需求端根据API接口的访问信息调用API接口,获取外部业务需求端对API接口的调用类型,并根据调用类型对应反馈业务数

据。

[0093] 如图6所示,所述配置模型生成单元101包括以下子单元:

[0094] 解析单元1011,用于解析外部业务需求信息中所包括的业务视图层节点的名称、与业务视图层节点相关联的实体的名称,实体之间的联系、及每一实体所对应关系型数据库中的业务数据;

[0095] 节点生成单元1012,用于根据业务视图层节点的名称,对应生成配置模型中的业务视图层节点;

[0096] 实体生成单元1013,用于根据与业务视图层节点相关联的实体的名称,对应生成配置模型中的实体;

[0097] 第一连线单元1014,用于根据业务视图层节点和与业务视图层节点相关联的实体之间的联系对应生成第一关系连接线组,并通过第一关系连接线组将配置模型中的业务视图层节点与实体进行连接;

[0098] 第二连线单元1015,用于根据实体之间的联系对应生成第二关系连接线组,并通过第二关系连接线组将配置模型中的实体与相关联实体进行连接。

[0099] 即,首先解析外部业务需求信息中的以下关键信息:业务视图层节点的名称,与业务视图层相关联的实体的名称,实体之间的相互关系,每一实体所对应关系型数据库中的业务数据;

[0100] 然后,根据业务视图层节点的名称对应生成配置模型中的业务视图层节点,根据实体的名称生成配置模型中的实体,根据实体与业务视图层节点之间的关系在配置模型中生成对应的第一关系连接线组(即View层的节点与E-R层中所包括实体之间的关系连接线),根据实体与实体之间的关系在配置模型中生成对应第二关系连接线组(及E-R层实体相互之间的关系连接线);

[0101] 最后,将配置模型保存在图形数据库中。

[0102] 通过上述过程,实现了配置模型的自动生成,或是人工配置而成。所生成的配置模型被保存至图形数据库中,即可进一步对该配置模型映射处理,以开放对应的API接口供用户进行数据操作。

[0103] 如图7所示,所述API接口生成单元102包括以下子单元:

[0104] 创建单元1021,用于根据配置模型中的业务视图层节点的名称创建API接口;

[0105] 映射单元1022,用于将API接口与配置模型中的业务视图层节点进行映射。

[0106] 其他发明实施例中,所述API接口生成单元102还用于:将API接口的访问信息发送至外部业务需求端;其中访问信息包括请求格式、请求参数、请求地址。

[0107] 在本实施例中,先根据配置模型中的业务视图层节点的名称初始化创建API接口,并根据业务视图层节点的属性进一步完善API接口的访问信息,外部业务需求端才能向服务端发出增删改查等业务请求,从而对应的对数据型数据库中的数据进行相应操作。

[0108] 其他发明实施例中,如图8所示,所述业务数据反馈单元103包括以下子单元:

[0109] 请求参数获取单元1031,用于根据外部业务需求端的访问信息中所包括的请求参数,获取外部业务需求端所访问API接口对应的业务视图层节点;

[0110] 脚本生成单元1032,用于根据业务视图层的表名、及对API接口的调用类型对应生成SQL脚本;其中所述调用类型包括增加数据请求、删除数据请求、编辑数据请求及查询数

据请求；

[0111] 数据获取单元1033,用于执行SQL脚本从关系型数据库中获取业务数据,并对反馈至API接口。

[0112] 在本实施例中,外部系统调用生成的API时,具有以下过程:

[0113] 先根据请求参数定位到业务视图层的View(即业务视图层节点);

[0114] 然后根据业务视图层的View生成对应的SQL脚本;

[0115] 最后在获取到SQL脚本后,在mybatis框架中执行,从关系型数据库中对应得到结果集,并反馈操作。

[0116] 其中,业务调用类型具体包括增加数据请求、删除数据请求、编辑数据请求、查询数据请求;

[0117] 当判断调用类型为查询数据请求,则根据查询数据请求、及业务视图层的view及查询的关键词生成第一SQL脚本(第一SQL脚本中包括select语句),执行第一SQL脚本在数据型数据库中查询得到对应的业务数据,并将业务数据通过业务视图层节点反馈至外部业务需求端;

[0118] 当判断调用类型为增加数据请求,则根据增加数据请求、及业务视图层的view及待增加的数据生成第二SQL脚本(第二SQL脚本中包括add语句),执行第二SQL脚本在数据型数据库中增加对应的业务数据,并将增加数据成功的消息反馈至外部业务需求端;

[0119] 当判断调用类型为删除数据请求,则根据删除数据请求、及业务视图层的view及待删除的数据生成第三SQL脚本(第三SQL脚本中包括delete语句),执行第三SQL脚本在数据型数据库中删除对应的业务数据,并将删除数据成功的消息反馈至外部业务需求端;

[0120] 当判断调用类型为编辑数据请求,则根据编辑数据请求、及业务视图层的view及待编辑的数据生成第四SQL脚本(第四SQL脚本中包括update语句),执行第四SQL脚本在数据型数据库中修改对应的业务数据,并将编辑数据成功的消息反馈至外部业务需求端。

[0121] 可见,该装置无需编写大量的SQL语句生成API接口,只需快速的修改配置模型即可得到满足需求的API接口,节省了开发大量SQL语句所耗用的时间,也避免了接口变更导致应用起停的问题。

[0122] 上述API接口动态生成装置可以实现为一种计算机程序的形式,该计算机程序可以在如图9所示的计算机设备上运行。

[0123] 请参阅图9,图9是本申请实施例提供的一种计算机设备的示意性框图。该计算机设备500设备可以是终端。该终端可以是平板电脑、笔记本电脑、台式电脑、个人数字助理等电子设备。

[0124] 参阅图9,该计算机设备500包括通过系统总线501连接的处理器502、存储器和网络接口505,其中,存储器可以包括非易失性存储介质503和内存储器504。

[0125] 该非易失性存储介质503可存储操作系统5031和计算机程序5032。该计算机程序5032包括程序指令,该程序指令被执行时,可使得处理器502执行一种API接口动态生成方法。

[0126] 该处理器502用于提供计算和控制能力,支撑整个计算机设备500的运行。

[0127] 该内存储器504为非易失性存储介质503中的计算机程序5032的运行提供环境,该计算机程序5032被处理器502执行时,可使得处理器502执行一种API接口动态生成方法。

[0128] 该网络接口505用于进行网络通信,如发送分配的任务等。本领域技术人员可以理解,图9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计算机设备500的限定,具体的计算机设备500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。

[0129] 其中,所述处理器502用于运行存储在存储器中的计算机程序5032,以实现如下功能:根据外部业务需求信息生成配置模型,并保存至图形数据库中;根据配置模型生成API接口;若检测到外部业务需求端根据API接口的访问信息调用API接口,获取外部业务需求端对API接口的调用类型,并根据调用类型对应反馈业务数据。

[0130] 在一实施例中,处理器502还执行如下操作:解析外部业务需求信息中所包括的业务视图层节点的名称、与业务视图层节点相关联的实体的名称,实体之间的联系、及每一实体所对应关系型数据库中的业务数据;根据业务视图层节点的名称,对应生成配置模型中的业务视图层节点;根据与业务视图层节点相关联的实体的名称,对应生成配置模型中的实体;根据业务视图层节点和与业务视图层节点相关联的实体之间的联系对应生成第一关系连接线组,并通过第一关系连接线组将配置模型中的业务视图层节点与实体进行连接;根据实体之间的联系对应生成第二关系连接线组,并通过第二关系连接线组将配置模型中的实体与相关联实体进行连接。

[0131] 在一实施例中,处理器502还执行如下操作:根据配置模型中的业务视图层节点的名称创建API接口;将API接口与配置模型中的业务视图层节点进行映射。

[0132] 在一实施例中,处理器502还执行如下操作:将API接口的访问信息发送至外部业务需求端;其中访问信息包括请求格式、请求参数、请求地址。

[0133] 在一实施例中,处理器502还执行如下操作:根据外部业务需求端的访问信息中所包括的请求参数,获取外部业务需求端所访问API接口对应的业务视图层节点;根据业务视图层的表名、及对API接口的调用类型对应生成SQL脚本;其中所述调用类型包括增加数据请求、删除数据请求、编辑数据请求及查询数据请求;执行SQL脚本从关系型数据库中获取业务数据,并对应反馈至API接口。

[0134] 本领域技术人员可以理解,图9中示出的计算机设备的实施例并不构成对计算机设备具体构成的限定,在其他实施例中,计算机设备可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。例如,在一些实施例中,计算机设备可以仅包括存储器及处理器,在这样的实施例中,存储器及处理器的结构及功能与图9所示实施例一致,在此不再赘述。

[0135] 应当理解,在本申请实施例中,处理器502可以是中央处理单元(Central Processing Unit,CPU),该处理器502还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable GateArray,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。其中,通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0136] 在本申请的另一实施例中提供一种存储介质。该存储介质可以为非易失性的计算机可读存储介质。该存储介质存储有计算机程序,其中计算机程序包括程序指令。该程序指令被处理器执行时实现:根据外部业务需求信息生成配置模型,并保存至图形数据库中;根

据配置模型生成API接口;若检测到外部业务需求端根据API接口的访问信息调用API接口,获取外部业务需求端对API接口的调用类型,并根据调用类型对应反馈业务数据。

[0137] 其中,所述根据业务视图层节点对应生成第一SQL脚本中,根据业务视图层节点的表名对应生成包括select语句的第一SQL脚本。

[0138] 在一实施例中,该程序指令被处理器执行时实现:解析外部业务需求信息中所包括的业务视图层节点的名称、与业务视图层节点相关联的实体的名称,实体之间的联系、及每一实体所对应关系型数据库中的业务数据;根据业务视图层节点的名称,对应生成配置模型中的业务视图层节点;根据与业务视图层节点相关联的实体的名称,对应生成配置模型中的实体;根据业务视图层节点和与业务视图层节点相关联的实体之间的联系对应生成第一关系连接线组,并通过第一关系连接线组将配置模型中的业务视图层节点与实体进行连接;根据实体之间的联系对应生成第二关系连接线组,并通过第二关系连接线组将配置模型中的实体与相关联实体进行连接。

[0139] 在一实施例中,该程序指令被处理器执行时实现:根据配置模型中的业务视图层节点的名称创建API接口;将API接口与配置模型中的业务视图层节点进行映射。

[0140] 在一实施例中,该程序指令被处理器执行时实现:将API接口的访问信息发送至外部业务需求端;其中访问信息包括请求格式、请求参数、请求地址。

[0141] 在一实施例中,该程序指令被处理器执行时实现:根据外部业务需求端的访问信息中所包括的请求参数,获取外部业务需求端所访问API接口对应的业务视图层节点;根据业务视图层的表名、及对API接口的调用类型对应生成SQL脚本;其中所述调用类型包括增加数据请求、删除数据请求、编辑数据请求及查询数据请求;执行SQL脚本从关系型数据库中获取业务数据,并对应反馈至API接口。

[0142] 所述存储介质可以是前述设备的内部存储单元,例如设备的硬盘或内存。所述存储介质也可以是所述设备的外部存储设备,例如所述设备上配备的插接式硬盘,智能存储卡(Smart Media Card, SMC),安全数字(Secure Digital, SD)卡,闪存卡(Flash Card)等。进一步地,所述存储介质还可以既包括所述设备的内部存储单元也包括外部存储设备。

[0143] 所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,上述描述的设备、装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0144] 在本申请所提供的几个实施例中,应该理解到,所揭露的设备、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,所述单元的划分,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,也可以将具有相同功能的单元集成成一个单元,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另外,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口、装置或单元的间接耦合或通信连接,也可以是电的,机械的或其它

的形式连接。

[0145] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元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单元来实现本发明实施例方案的目的。

[0146] 另外,在本发明各个实施例中的各功能单元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,也可以是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,也可以采用软件功能单元的形式实现。

[0147] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分,或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品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,服务器,或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0148] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到各种等效的修改或替换,这些修改或替换都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求的保护范围为准。

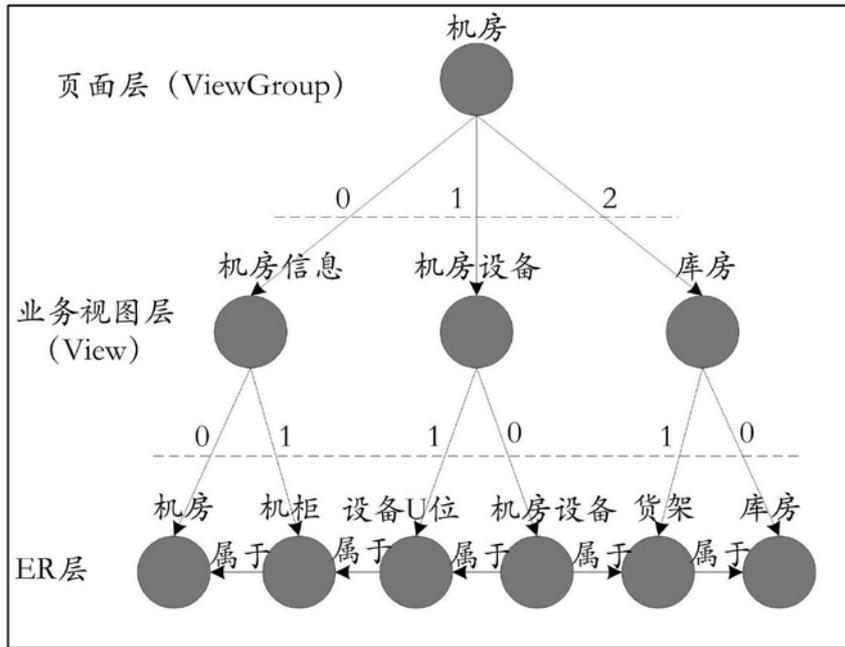


图1a

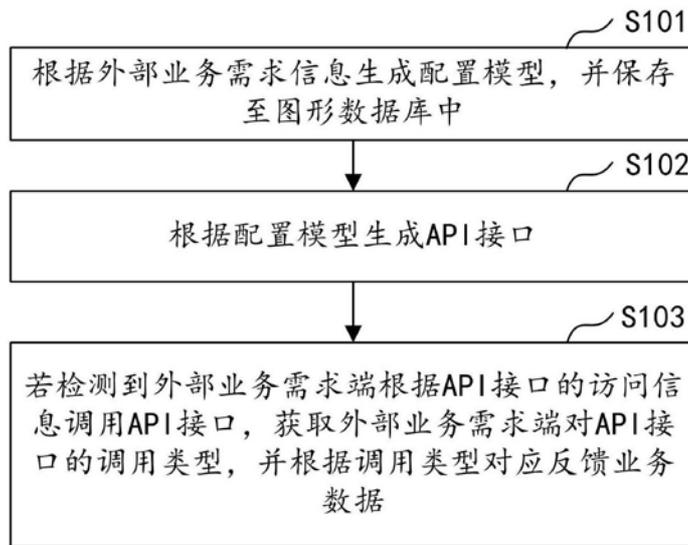


图1b

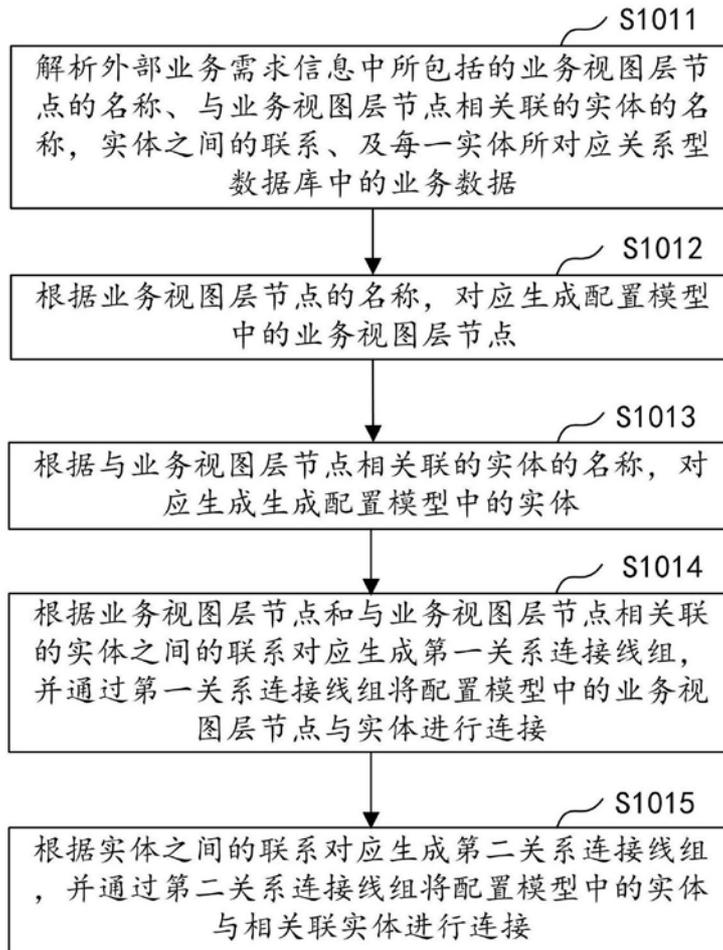


图2

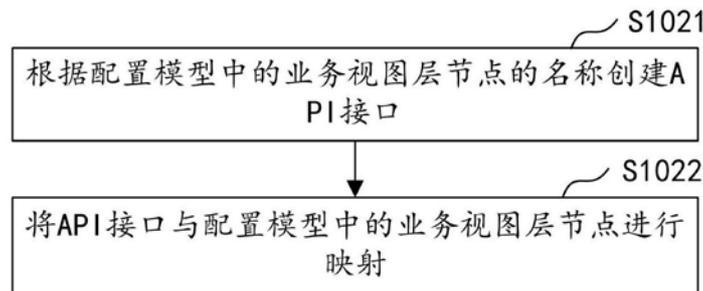


图3

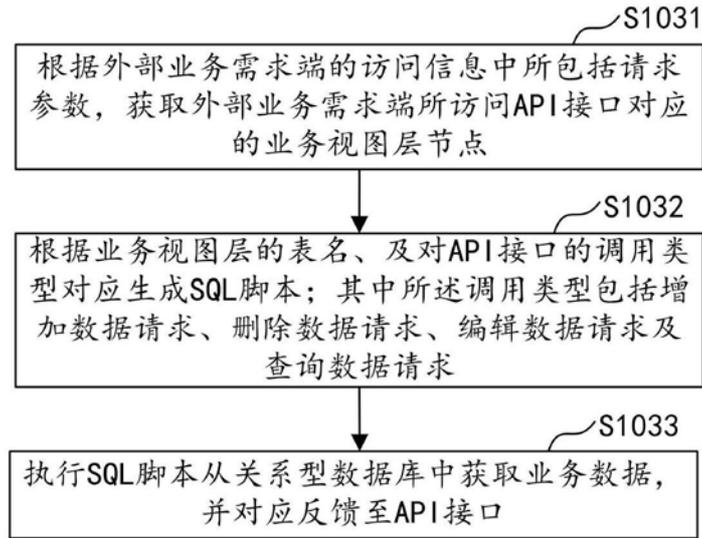


图4

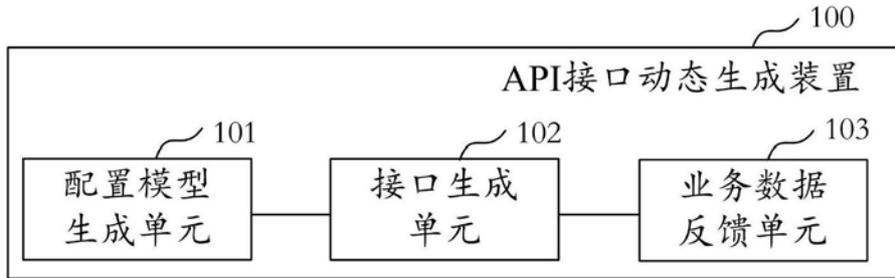


图5

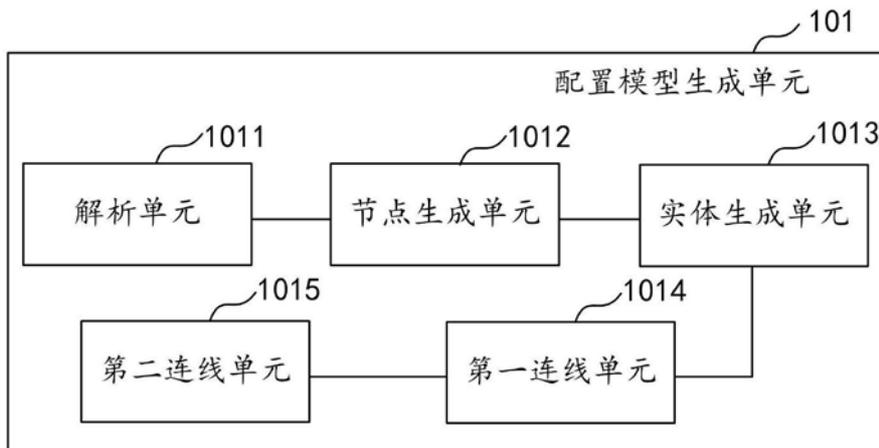


图6

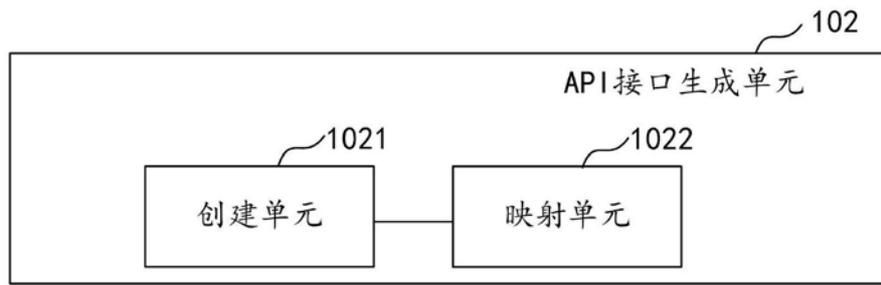


图7

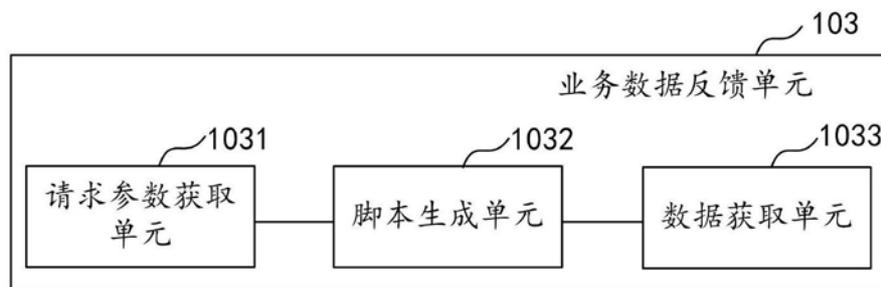


图8

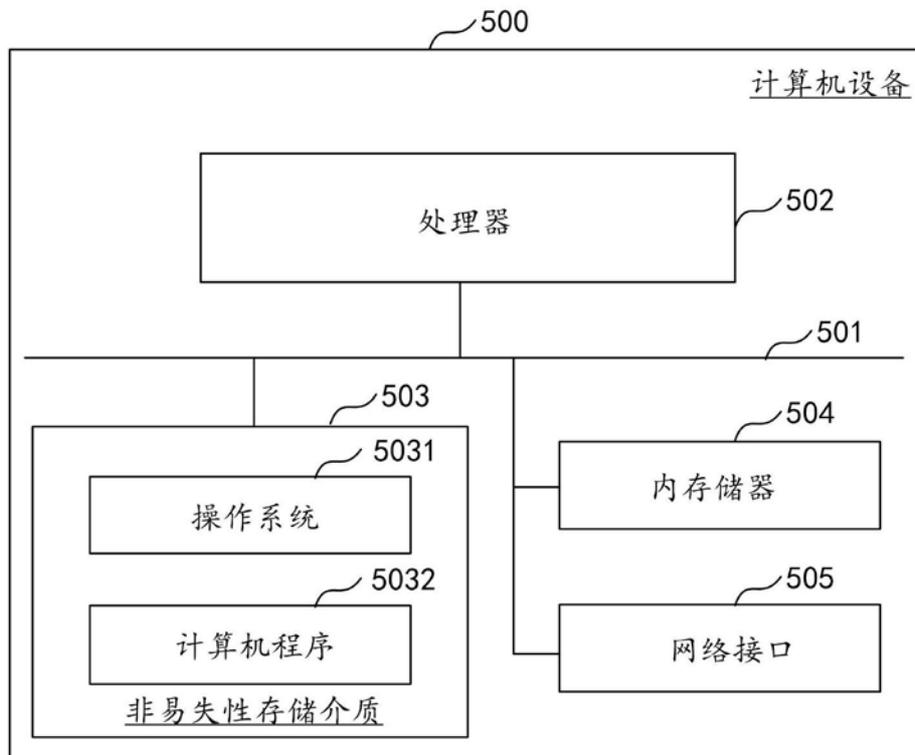


图9