



(12) 发明专利申请

(10) 申请公布号 CN 103473108 A

(43) 申请公布日 2013. 12. 25

(21) 申请号 201310349318. 6

(22) 申请日 2013. 08. 12

(71) 申请人 福建富士通信息软件有限公司

地址 350000 福建省福州市鼓楼区五凤街道
软件大道 89 号 12 号楼

(72) 发明人 陈华光 徐舒延 王涵 刘雅文

(74) 专利代理机构 福州市鼓楼区京华专利事务
所(普通合伙) 35212

代理人 宋连梅

(51) Int. Cl.

G06F 9/45(2006. 01)

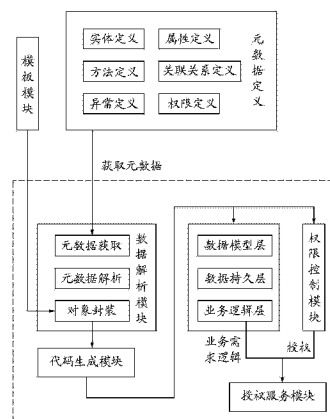
权利要求书2页 说明书5页 附图2页

(54) 发明名称

一种 Java 代码生成方法

(57) 摘要

一种 Java 代码生成方法,该方法需提供一模板模块、数据解析模块、代码生成模块和数据模块,代码生成模块中定义有源代码自动生成脚本;具体操作方法:提取元数据,将元数据进行定义形成元数据文件,并存储于数据模块中;数据解析模块提取元数据文件,解析出元数据,同时从模板模块获取代码模板,并把元数据转换为模板对象;代码模板中具有用于标识所述模板对象的命名参数;代码生成模块执行所述源代码自动生成脚本,使用所述模板对象对所述代码模板中的命名参数进行赋值,生成复数个源代码文件,并保存在数据模块中且对应于所述元数据文件所在位置处;本发明具有可维护性高、代码一致性好、可重用性强,提高了开发效率,降低了人力成本。



1. 一种 Java 代码生成方法,其特征在于:该方法需提供一模板模块、数据解析模块、代码生成模块和数据模块,所述代码生成模块中定义有一源代码自动生成脚本;所述 Java 代码生成方法的具体操作方法如下:

步骤 10:将元数据进行解释定义,形成一元数据文件,并存储于所述数据模块中;所述元数据用于描述数据结构、数据库中的要素及要素之间关联关系信息的数据;

步骤 20:所述数据解析模块提取所述元数据文件,并解析出元数据,同时从所述模板模块获取代码模板,并根据所述代码模板封装元数据,把元数据转换为模板对象,最后将所述代码模板和模板对象交给代码生成模块;所述代码模板中具有用于标识所述模板对象的命名参数;

步骤 30:代码生成模块执行所述源代码自动生成脚本,使用所述模板对象对所述代码模板中的命名参数进行赋值,生成复数个源代码文件,并保存在数据模块中且对应于所述元数据文件所在位置处,该元数据文件所在位置处为:步骤 10 中元数据文件存储在数据模块中的地址位置。

2. 如权利要求 1 所述的一种 Java 代码生成方法,其特征在于:所述步骤 10 中的元数据定义为 6 种类型,包括实体定义、属性定义、方法定义、关联关系定义、异常定义和权限定义;

所述实体定义用于指定代码的实体类,其对应着一数据库的数据库表;

所述属性定义用于指定实体类下的属性,其对应着一数据库的数据库表字段;

所述方法定义用于指定实体类下对一数据库进行操作的方法,其对应着基本数据操作;

所述关联关系定义用于指定各实体类之间的引用关系,其对应着一数据库的数据库表之间的关联关系;

所述异常定义用于生成一实体类的异常类,以精确定位异常;

所述权限定义用于定义功能模块,并为各功能模块分别加上权限限制。

3. 如权利要求 2 所述的一种 Java 代码生成方法,其特征在于:所述复数个源代码文件的逻辑架构分为数据模型层、数据持久层和业务逻辑层;所述数据模型层与一数据库对应,数据模型层中的每一实体类均对应于该数据库中的一数据库表;所述数据库持久层对应于所述数据库的基本数据操作;所述业务逻辑层为各功能模块提供业务逻辑;

所述数据持久层调用数据模型层操作数据库,所述业务逻辑层调用数据持久层,把各种业务逻辑处理完的数据写入数据库相关联的数据库表中,同时处理写入过程中上报的异常,精确定位到具体实体类的何种异常。

4. 如权利要求 2 所述的一种 Java 代码生成方法,其特征在于:所述步骤 30 中,代码生成模块执行所述源代码自动生成脚本后,代码生成模块还生成一权限控制模块和一授权服务模块;所述权限控制模块用于对各功能模块设置访问权限,所述授权服务模块用于对用户访问数据模块的访问请求进行访问权限检测;

所述 Java 代码生成方法还包括以下内容:

步骤 40:代码生成模块再次执行所述源代码自动生成脚本,然后检测所述功能模块和业务逻辑层是否存在变更,若是,则执行步骤 50;若否,则结束流程;

步骤 50:结合所述元数据的权限定义中为各功能模块所设置的权限,权限控制模块为

各功能模块分别加上访问权限控制,生成具有权限控制的本地服务代码。

5. 如权利要求 1 所述的一种 Java 代码生成方法,其特征在于:所述代码生成模块中还定义有一数据库初始化生成脚本;

步骤 20 和步骤 30 之间还包括步骤 21:代码生成模块执行数据库初始化生成脚本,并采用所述模板对象对所述代码模板中的命名参数进行赋值,生成数据库建表脚本,并保存于数据模块中。

6. 如权利要求 1 所述的一种 Java 代码生成方法,其特征在于:所述元数据文件为 XML 文档格式文件。

一种 Java 代码生成方法

技术领域

[0001] 本发明具体涉及一种 Java 代码生成方法。

背景技术

[0002] 现有代码生成方法一般分为两大类：被动代码生成方法和主动代码生成方法。在被动代码生成方法中，一般采用可视化向导的方式配置生成部分代码，然后软件开发人员对代码进行必要的编辑修改，这种生成过程只进行一次，不对已修改的代码进行维护；目前大部分的集成开发环境(Integrated Development Environment, IDE)或者编辑器就包含有这种代码生成方法。但是这种代码生成方法的局限性也是显而易见的，IDE 或者编辑器只负责生成一次，剩下的工作量都是软件开发人员的，重新生成将会覆盖已修改的代码，而且也只适用于部分场景，可配置性不足。

[0003] 在主动代码生成方法中，软件开发人员同样可对生成的代码进行编辑修改，而改动后的代码可以作为参数导入生成器，生成器保留改动后的代码，并影响之后的代码生成，同时可配置性以及适用范围都比被动代码生成方法更好，目前编译器类型的生成器属于主动代码生成方法。然而，现有的主动代码生成方法是直接通过数据库表进行的，由数据库表开始解析，必须转化出对应某一种数据库的建表脚本创建数据库，这要求代码生成方法要适配于多种主流数据库。另外，虽然可对单个新添加的数据库表单独生成代码，但是生成的代码全部混合放置于各个层次的包中，不能按照数据相关性分包分类生成，无法简便地在数据库模型阶段引入自定义标识、检索、异常等逻辑。

发明内容

[0004] 本发明所要解决的技术问题在于提供一种 Java 代码生成方法，具有可维护性高、代码一致性好、可重用性强，提高了开发效率，降低了人力成本。

[0005] 本发明是通过以下技术方案解决上述技术问题的：一种 Java 代码生成方法，该方法需提供一模板模块、数据解析模块、代码生成模块和数据模块，所述代码生成模块中定义有一源代码自动生成脚本；所述 Java 代码生成方法的具体操作方法如下：

[0006] 步骤 10：将元数据进行解释定义，形成一元数据文件，并存储于所述数据模块中；所述元数据用于描述数据结构、数据库中的要素及要素之间关联关系信息的数据；

[0007] 步骤 20：所述数据解析模块提取所述元数据文件，并解析出元数据，同时从所述模板模块获取代码模板，并根据所述代码模板封装元数据，把元数据转换为模板对象，最后将所述代码模板和模板对象交给代码生成模块；所述代码模板中具有用于标识所述模板对象的命名参数；

[0008] 步骤 30：代码生成模块执行所述源代码自动生成脚本，使用所述模板对象对所述代码模板中的命名参数进行赋值，生成复数个源代码文件，并保存在数据模块中且对应于所述元数据文件所在位置处，该元数据文件所在位置处为：步骤 10 中元数据文件存储在数据模块中的地址位置。

[0009] 进一步地,所述步骤 10 中的元数据定义为 6 种类型,包括实体定义、属性定义、方法定义、关联关系定义、异常定义和权限定义;

[0010] 所述实体定义用于指定代码的实体类,其对应着一数据库的数据库表;

[0011] 所述属性定义用于指定实体类下的属性,其对应着一数据库的数据库表字段;

[0012] 所述方法定义用于指定实体类下对一数据库进行操作的方法,其对应着基本数据操作;

[0013] 所述关联关系定义用于指定各实体类之间的引用关系,其对应着一数据库的数据库表之间的关联关系;

[0014] 所述异常定义用于生成一实体类的异常类,以精确定位异常;

[0015] 所述权限定义用于定义功能模块,并为各功能模块分别加上权限限制。

[0016] 进一步地,所述复数个源代码文件的逻辑架构分为数据模型层、数据持久层和业务逻辑层;所述数据模型层与一数据库对应,数据模型层中的每一实体类均对应于该数据库中的一数据库表;所述数据库持久层对应于所述数据库的基本数据操作;所述业务逻辑层为各功能模块提供业务逻辑;

[0017] 所述数据持久层调用数据模型层操作数据库,所述业务逻辑层调用数据持久层,把各种业务逻辑处理完的数据写入数据库相关联的数据库表中,同时处理写入过程中上报的异常,精确定位到具体实体类的何种异常。

[0018] 进一步地,所述步骤 30 中,代码生成模块执行所述源代码自动生成脚本后,代码生成模块还生成一权限控制模块和一授权服务模块;所述权限控制模块用于对各功能模块设置访问权限,所述授权服务模块用于对用户访问数据模块的访问请求进行访问权限检测;

[0019] 所述 Java 代码生成方法还包括以下内容:

[0020] 步骤 40:代码生成模块再次执行所述源代码自动生成脚本,然后检测所述功能模块和业务逻辑层是否存在变更,若是,则执行步骤 50;若否,则结束流程;

[0021] 步骤 50:结合所述元数据的权限定义中为各功能模块所设置的权限,权限控制模块为各功能模块分别加上访问权限控制,生成具有权限控制的本地服务代码。

[0022] 进一步地,所述代码生成模块中还定义有一数据库初始化生成脚本;

[0023] 步骤 20 和步骤 30 之间还包括步骤 21:代码生成模块执行数据库初始化生成脚本,并采用所述模板对象对所述代码模板中的命名参数进行赋值,生成数据库建表脚本,并保存于数据模块中。

[0024] 进一步地,所述元数据文件为 XML 文档格式文件。

[0025] 本发明的有益效果在于:相对于现有的直接通过数据库表进行 Java 代码生成的方法,本发明使用了元数据文件即 XML 文本作为元数据的载体,不过度依赖于具体的数据库,本发明按照所述元数据文件单独进行代码生成,所述元数据文件可只对应一张数据库表,也可对应多个耦合度高的库表,且生成的源代码文件按所述元数据文件分包,再分层分类,结构更加简明清晰;具有可维护性高、代码一致性好、可重用性强等优势,本发明通过在多个实际项目中的实践,自动生成的代码占了整个系统源代码的大部分,显著提高了开发效率,减少了开发人员的数量,降低了人力成本。

附图说明

[0026] 下面参照附图结合实施例对本发明作进一步的描述。

[0027] 图 1 为本发明一种 Java 代码生成方法的原理框图。

[0028] 图 2 为本发明中数据模块的原理框图。

具体实施方式

[0029] 请参阅图 1, 一种 Java 代码生成方法, 该方法需提供一模板模块、数据解析模块、代码生成模块和数据模块, 所述代码生成模块中定义有一源代码自动生成脚本和一数据库初始化生成脚本;

[0030] 所述 Java 代码生成方法的具体操作方法如下:

[0031] 步骤 10: 将元数据进行定义形成一 XML 文档格式的元数据文件, 并存储于所述数据模块中, 即存储在 src 目录里, src 目录在软件开发里是默认存放下文所述源代码文件的地方; 所述元数据文件为 XML 文档格式文件, 以方便人们进行编写查看。所述元数据用于描述数据结构、数据库中的要素及要素之间关联关系信息的数据, 元数据是用于描述数据及其环境的数据, 本发明通过解释元数据以达到控制代码行为。

[0032] 所述元数据定义为 6 种类型, 包括实体定义、属性定义、方法定义、关联关系定义、异常定义和权限定义;

[0033] 所述实体定义用于指定代码的实体类, 其对应着一数据库的数据库表, 一个元数据文件里可以有多个实体定义;

[0034] 所述属性定义用于指定实体类下的属性, 其对应着一数据库的数据库表字段;

[0035] 所述方法定义用于指定实体类下对一数据库进行操作即增删改查的方法, 其对应着基本数据操作;

[0036] 所述关联关系定义用于指定各实体类之间的引用关系, 其对应着一数据库的数据库表之间的关联关系;

[0037] 所述异常定义用于生成一实体类的异常类, 方便精确定位异常;

[0038] 所述权限定义用于定义功能模块, 并为各功能模块分别加上权限限制。

[0039] 步骤 20: 所述数据解析模块提取所述元数据文件, 并解析出元数据, 同时从所述模板模块获取代码模板, 并根据所述代码模板封装元数据, 把元数据转换为模板对象, 最后将所述代码模板和模板对象交给代码生成模块; 所述代码模板中具有用于标识所述模板对象的命名参数;

[0040] 所述模板对象是元数据的封装格式, 每一命名参数均是代码模板上的一个标记, 命名参数是做代码模板时定义的, 只是个标记, 标记这里要填入哪种模板对象, 而元数据经封装成模板对象后最终要替换代码模板中的命名参数。

[0041] 步骤 21: 代码生成模块执行数据库初始化生成脚本, 并采用所述模板对象对所述代码模板中的命名参数进行赋值, 生成数据库建表脚本, 并保存于数据模块中, 即存储于 sql 目录里; 即数据库初始化生成脚本会统一检测所有元数据的变更, 就是检查所有元数据添加、修改如新增一个元数据文件, 修改文件中的一个或几个参数等等, 都要反馈到生成的数据库建表脚本中, 再生成数据库建表脚本; 且所述数据库建表脚本存放在下文所述源代码文件的 src 目录中。

[0042] 本发明的数据库初始化生成脚本支持所有主流数据库,将统一生成所有主流数据库的建表脚本,方便选用数据库建库建表,因为执行所述数据库建表脚本就可以自动生成数据库表,不需手动编写建表脚本建数据库,数据库建表脚本建的是数据库和数据库表,而下文所述数据模型层就是与数据库对应,对数据模型层的操作将直接反映到使用数据库建表脚本创建的数据库中,每一张数据库表对应于数据模型层中的一个实体类,对实体类中数据的变更最终都将写入到数据库表中。

[0043] 步骤 30 :代码生成模块执行所述源代码自动生成脚本,代码生成模块使用所述模板对象对所述代码模板中的命名参数进行赋值,生成复数个源代码文件,保存在数据模块中并对应于所述元数据文件所在位置处,该元数据文件所在位置处为 :步骤 10 中元数据文件存储在数据模块中的地址位置 ;同时,代码生成模块执行所述源代码自动生成脚本后,代码生成模块还生成一权限控制模块和一授权服务模块。

[0044] 所述源代码自动生成脚本可对一个元数据文件单独执行代码生成,也可对多个元数据文件同时进行代码生成 ;当有多个元数据文件同时进行代码生成时,生成的源代码文件将按照所属元数据文件分包存储在该元数据文件所在的位置,即用一个元数据文件执行代码生成的源代码文件就放在该元数据文件所在位置,若有多个元数据文件,元数据文件要分包存储,然后各个元数据文件生成的源代码就生成在自己的元数据文件的包中,如图 2 所示,例如有两个元数据文件就分为 2 个包(包 1、包 2),包 1 中的元数据文件执行代码生成的源代码文件就存储于包 1 中,且该些源代码文件按照各自所提供的功能进行归类,分为三类即三个子包分别是数据模型层、数据持久层或业务逻辑层,对应于数据模型层的源代码文件就存储于该数据模型层的子包中,对应于数据持久层的源代码文件就存储于该数据持久层的子包中,对应于业务逻辑层的源代码文件就存储于该业务逻辑层的子包中 ;同理,包 2 的元数据文件生成的源代码文件也是按照如此规律进行分包存储,再分层分类。

[0045] 所述复数个源代码文件的逻辑架构分为数据模型层、数据持久层和业务逻辑层 ;所述数据模型层与一数据库对应,数据模型层中的每一实体类均对应于该数据库中的一数据库表 ;所述数据库持久层对应于所述数据库的基本数据操作 ;所述业务逻辑层为各功能模块提供业务逻辑 ;基本数据操作如查改增删,重复性很高,同样数据模型层中的实体类代码也具有较高重复性,当一个数据库的数据库表达到几十到上百张后,本发明对效率的提升显而易见。

[0046] 所述数据持久层调用数据模型层操作数据库,所述业务逻辑层调用数据持久层,把各种业务逻辑处理完的数据写入数据库相关联的数据库表中,同时处理写入过程中上报的异常,精确定位到具体实体类的何种异常。

[0047] 上述元数据的实体定义和属性定义对应于数据模型层,方法定义对应于数据持久层,而关联关系定义、异常定义和权限定义都可对应于业务逻辑层,关联关系定义有部分也对应数据持久层。所述数据模型层、数据持久层、业务逻辑层都是由源代码文件组成的,这三层相当于对不同的源代码文件进行归类 ;所述源代码文件是由字符代码组成的,源代码文件提供的功能是操作数据库(如查询、修改等)就归到数据持久层这一类中。

[0048] 所述权限控制模块用于对各功能模块设置访问权限,在权限控制模块中将维护一张权限控制表,记录某个用户对各功能模块是否具有访问权限 ;将各功能模块的访问权限组合成一个或多个权限集合,形成角色,然后权限控制模块可将权限集合(即角色)授权给

一个或多个用户,某个用户就有了对部分功能模块的访问权限。

[0049] 所述授权服务模块用于对用户访问数据模块的访问请求进行访问权限检测;授权服务模块作用是对用户的访问请求进行检测,所有对数据模块的访问使用,都会经过授权服务模块来检测其所属的用户及其访问权限,若用户无权限访问数据模块中的某一数据,授权服务模块将阻止其访问。

[0050] 步骤 40:代码生成模块再次执行所述源代码自动生成脚本,然后检测所述功能模块和业务逻辑层是否存在变更,若是,则执行步骤 50;若否,则结束流程;

[0051] 步骤 50:结合所述元数据的权限定义中为各功能模块所设置的权限,权限控制模块为各功能模块分别加上访问权限控制,生成具有权限控制的本地服务代码。

[0052] 可在所述源代码自动生成脚本的参数中设置是否生成对外服务接口(默认为否),然后可再次执行所述源代码自动生成脚本,判断是否生成对外接口,若否,则结束流程;若是,则生成具有权限控制的对外服务接口;对外接口服务是面向服务的 Internet 应用接口,它使用基于 XML 的消息处理作为基本的数据通讯方式,使用者可以通过网络远程调用授权服务模块判断权限,进而调用数据模块为系统服务。

[0053] 相对于现有的直接通过数据库表进行 Java 代码生成的方法,本发明使用了 XML 文本作为元数据的载体,不过度依赖于具体的数据库,将技术选择推迟到后期,本发明按照元数据定义文件即所述元数据文件单独进行代码生成,所述元数据文件可只对应一张数据库表,也可对应多个耦合度高的库表,且生成的源代码文件按所述元数据文件分包,再分层分类,结构更加简明清晰。本发明具有可维护性高、代码一致性好、可重用性强等优势,本发明通过在多个实际项目中的实践,自动生成的代码占了整个系统源代码的大部分,显著提高了开发效率,开发人员仅大约占总人员的 40%,降低了人力成本。

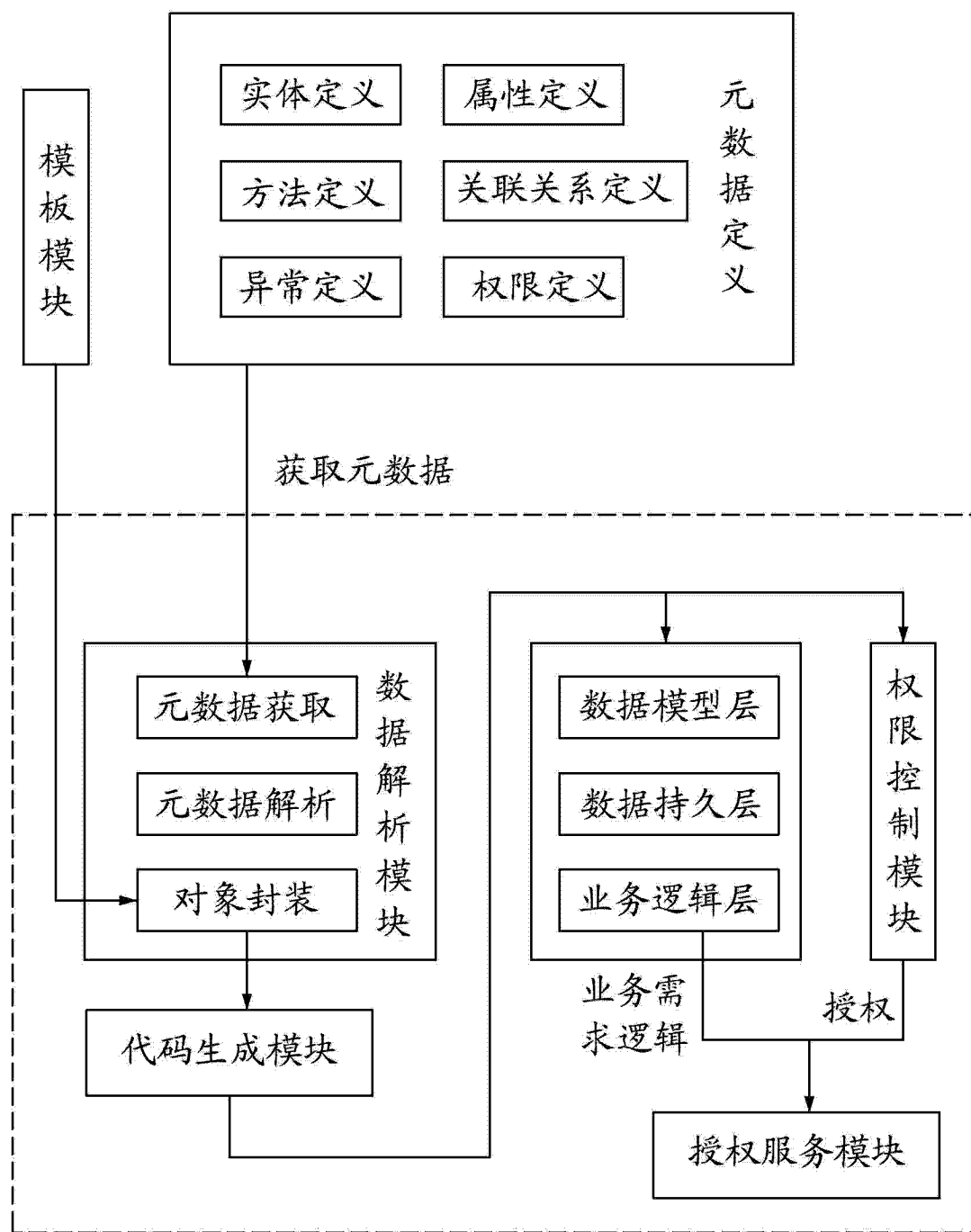


图 1

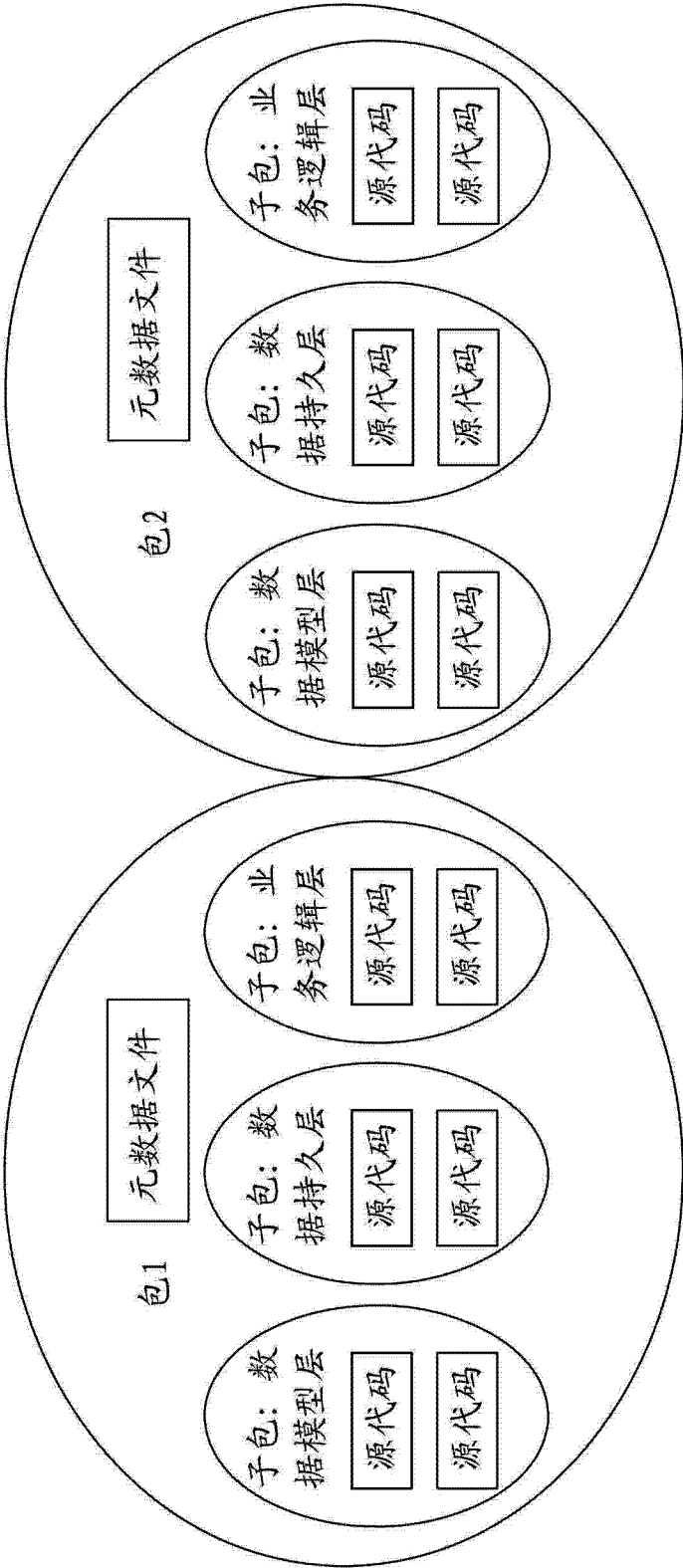


图 2