(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104168250 A
(43)申请公布日 2014.11.26

(21)申请号 201310180911.2

(22)申请日 2013.05.15

(71)申请人 腾讯科技(深圳)有限公司
地址 518044 广东省深圳市福田区振兴路赛格科技园2栋东403室

(72)发明人 周龄

(74)专利代理机构 广州华进联合专利商标代理有限公司 44224
代理人 何平 邓云鹏

(51)Int.Cl.
H04L 29/06(2006.01)

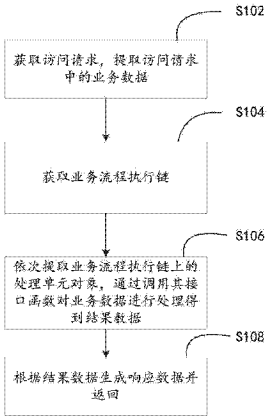
权利要求书1页 说明书5页 附图3页

(54)发明名称

基于CGI框架的业务流程控制方法及装置

(57)摘要

一种基于CGI框架的业务流程控制方法，包括：获取访问请求，提取所述访问请求中的业务数据；获取业务流程执行链；依次提取所述业务流程执行链上的处理单元对象，通过调用其接口函数对所述业务数据进行处理得到结果数据；根据所述结果数据生成响应数据并返回。此外，还提供了一种基于CGI框架的业务流程控制装置。上述基于CGI框架的业务流程控制方法和装置提高了扩展性。



CN 104168250 A

1. 一种基于 CGI 框架的业务流程控制方法，包括：

获取访问请求，提取所述访问请求中的业务数据；

获取业务流程执行链；

依次提取所述业务流程执行链上的处理单元对象，通过调用其接口函数对所述业务数据进行处理得到结果数据；

根据所述结果数据生成响应数据并返回。

2. 根据权利要求 1 所述的基于 CGI 框架的业务流程控制方法，其特征在于，所述通过调用处理单元对象的接口函数对所述业务数据进行处理得到结果数据的步骤还包括：

获取中间数据接口类定义；

根据所述中间数据接口类定义将所述接口函数生成的中间数据封装成中间数据对象，并在所述处理单元对象之间传递所述中间数据对象。

3. 根据权利要求 1 所述的基于 CGI 框架的业务流程控制方法，其特征在于，所述方法还包括：

获取处理单元对象定义，生成处理单元对象；

获取业务流程执行链定义；

根据所述业务流程执行链定义通过拼接所述处理单元对象生成业务流程执行链。

4. 根据权利要求 1 所述的基于 CGI 框架的业务流程控制方法，其特征在于，所述方法还包括：

判断所述接口函数是否抛出异常，若是，则输出并记录所述异常。

5. 根据权利要求 4 所述的基于 CGI 框架的业务流程控制方法，其特征在于，所述判断所述接口函数是否抛出异常的步骤之后还包括：

若是，则判断所述异常是否需要中止，若是，则中止所述业务流程执行链的执行。

6. 一种基于 CGI 框架的业务流程控制装置，其特征在于，包括：

请求获取模块，用于获取访问请求，提取所述访问请求中的业务数据；

执行链获取模块，用于获取业务流程执行链；

链式处理模块，用于依次提取所述业务流程执行链上的处理单元对象，通过调用其接口函数对所述业务数据进行处理得到结果数据；

响应模块，用于根据所述结果数据生成响应数据并返回。

7. 根据权利要求 6 所述的基于 CGI 框架的业务流程控制装置，其特征在于，所述链式处理模块还用于获取中间数据接口类定义，根据所述中间数据接口类定义将所述接口函数生成的中间数据封装成中间数据对象，并在所述处理单元对象之间传递所述中间数据对象。

8. 根据权利要求 6 所述的基于 CGI 框架的业务流程控制装置，其特征在于，所述装置还包括执行链生成模块，用于获取处理单元对象定义，生成处理单元对象；获取业务流程执行链定义；根据所述业务流程执行链定义通过拼接所述处理单元对象生成业务流程执行链。

9. 根据权利要求 6 所述的基于 CGI 框架的业务流程控制装置，其特征在于，所述装置还包括异常处理模块，用于判断所述接口函数是否抛出异常，若是，则输出并记录所述异常。

10. 根据权利要求 9 所述的基于 CGI 框架的业务流程控制装置，其特征在于，所述异常处理模块还用于判断所述异常是否需要中止，若是，则中止所述业务流程执行链的执行。

基于 CGI 框架的业务流程控制方法及装置

技术领域

[0001] 本发明涉及互联网技术领域，特别是涉及一种基于 CGI 框架的业务流程控制方法及装置。

背景技术

[0002] CGI（Common Gateway Interface，通用网关接口）是外部应用程序（CGI 程序）与 Web 服务器之间的接口标准，是在 CGI 程序和 Web 服务器之间传递信息的规程。CGI 规范允许 Web 服务器执行外部程序，并将它们的输出发送给 Web 浏览器，CGI 将 Web 的一组简单的静态超媒体文档变成一个完整的新的交互式媒体。

[0003] 传统的 CGI 程序在对业务流程的执行进行控制时，通常通过依次调用 CGI 框架的用于增加、删除、修改和查看的库函数实现对业务流程的控制。

[0004] 然而，发明人经研究发现，传统技术中至少存在以下问题：在编写业务流程的代码时，不同的开发人员对某些可复用的业务逻辑代码进行重复定义（例如，根据 CGI 程序的 API 返回的处理状态码进行异常处理的业务逻辑代码），使得代码的复用率较低，从而导致扩展性不高。

发明内容

[0005] 基于此，有必要提供一种能提高扩展性的基于 CGI 框架的业务流程控制方法。

[0006] 一种基于 CGI 框架的业务流程控制方法，包括：

[0007] 获取访问请求，提取所述访问请求中的业务数据；

[0008] 获取业务流程执行链；

[0009] 依次提取所述业务流程执行链上的处理单元对象，通过调用其接口函数对所述业务数据进行处理得到结果数据；

[0010] 根据所述结果数据生成响应数据并返回。

[0011] 此外，还有必要提供一种能提高扩展性的基于 CGI 框架的业务流程控制装置。

[0012] 一种基于 CGI 框架的业务流程控制装置，包括：

[0013] 请求获取模块，用于获取访问请求，提取所述访问请求中的业务数据；

[0014] 执行链获取模块，用于获取业务流程执行链；

[0015] 链式处理模块，用于依次提取所述业务流程执行链上的处理单元对象，通过调用其接口函数对所述业务数据进行处理得到结果数据；

[0016] 响应模块，用于根据所述结果数据生成响应数据并返回。

[0017] 上述基于 CGI 框架的业务流程控制方法及装置，将 CGI 框架中的业务流程代码拆分并封装成对应多个业务流程的子过程的处理单元对象，并按照与实际业务流程对应的业务流程执行链的顺序依次调用处理单元对象的接口函数来实现业务流程的逻辑，使得不同的开发人员在各自实现需要编写的业务流程逻辑的代码过程中，可直接通过向业务流程执行链中添加或注册相应的处理单元对象，使得代码的复用性得到提高，从而在面对新的业

务流程需求时，提高了扩展性。

附图说明

[0018] 图 1 为一个实施例中基于 CGI 框架的业务流程控制方法的流程图；

[0019] 图 2 为一个实施例中基于 CGI 框架的业务流程控制方法中链式调用处理单元对象的过程示意图；

[0020] 图 3 为一个应用场景中运行所述基于 CGI 框架的业务流程控制方法的程序框架的运行过程示意图；

[0021] 图 4 为一个实施例中基于 CGI 框架的业务流程控制装置的结构示意图；

[0022] 图 5 为另一个实施例中基于 CGI 框架的业务流程控制装置的结构示意图。

具体实施方式

[0023] 在一个实施例中，如图 1 所示，一种基于 CGI 框架的业务流程控制方法，该方法完全依赖于基于 CGI 框架的计算机程序，该计算机程序可运行于基于冯洛伊曼体系的服务器设备上。该方法包括以下步骤：

[0024] 步骤 S102，获取访问请求，提取访问请求中的业务数据。

[0025] CGI 框架是 CGI 程序与 web 服务器之间的接口标准。web 服务器(例如 Apache)用于处理接收网页访问请求(html 请求)，并根据网页访问请求提取相应的 web 服务器上存储的网络资源的资源地址(即 url，Uniform Resource Locator，统一资源定位符)，然后根据资源地址返回网络资源。若网页访问请求对应有需要对 web 服务器后端的数据库进行增加、删除、修改或查看的操作，通常需要 web 服务器通过 CGI 框架进行处理。因此 web 服务器会将接收到的访问请求转发给基于 CGI 框架的计算机程序(也叫外部应用程序)，该程序接收到 web 服务器转发的访问请求后，则可提取访问请求中包含的业务数据。

[0026] 例如，若访问请求为 POST (HTTP 请求中 method-type 为 POST) 请求，其中包含有需要保存的表单数据，则提取到的业务数据即为该表单数据。若访问请求为查询请求，则提取到的业务数据即为访问请求(可以是 POST 也可以是 GET) 中包含的关键字。

[0027] 步骤 S104，获取业务流程执行链。

[0028] 业务流程执行链为按照注册顺序拼接而成的处理单元对象的序列。

[0029] 在一个实施例中，生成业务流程执行链的步骤可具体为：

[0030] 获取处理单元对象定义，生成处理单元对象；获取业务流程执行链定义；根据业务流程执行链定义通过拼接处理单元对象生成业务流程执行链。

[0031] 处理单元对象可以是程序语言中的对象(例如，c++ 或 python 中的 Object)，也可以是结构体(例如，c 语言中 Struct)。处理单元对象定义可以是类文件或函数头文件。

[0032] 也就是说，开发人员可预先根据业务流程的执行将业务流程分为多个子过程，每个子过程中执行的业务逻辑可定义在相应的处理单元对象定义中。根据处理单元对象定义生成的处理单元对象即具有了处理相应业务流程的子过程的功能。

[0033] 业务流程执行链定义即定义了业务流程执行链包含的处理单元对象的类型以及处理单元对象的调用顺序。在生成处理单元对象之后，可根据业务流程执行链定义将处理单元对象按照其定义添加到业务流程执行链中。

[0034] 步骤 S106，依次提取业务流程执行链上的处理单元对象，通过调用其接口函数对业务数据进行处理得到结果数据。

[0035] 在一个实施例中，处理单元对象可具有统一的接口函数（实现同一接口或继承自同一抽象类）。使用统一的接口函数可以通过动态加载或反射等技术在运行时动态地加载并调用处理单元对象。从而进一步地提高扩展性。

[0036] 如图 2 所示，由于业务流程执行链上的处理单元对象具有执行顺序，因此，可先取出业务流程执行链上的按顺序排列的第一个处理单元对象，将由访问请求提取到的业务数据作为参数传递给该处理单元对象的接口函数。接口函数执行完毕后得到返回值，该返回值即为中间数据。然后将中间数据传递给该处理单元对象在业务流程执行链上按照顺位的下一个处理单元对象，并以此类推一直到业务流程执行链上的所有处理单元对象均被调用，最终得到结果数据。

[0037] 需要说明的是，业务数据以及中间数据可以以全局变量缓存，传递进入业务流程执行链上的处理单元对象的接口函数的参数可以不仅是相邻的前一处理单元对象生成的中间数据，也可以包括缓存中存储的业务数据或中间数据。结果数据可不仅仅包括位于业务流程执行链上的最后一个处理单元对象的接口函数的返回值，还可以包括业务流程执行链上其他处理单元对象生成的中间数据。

[0038] 在一个实施例中，通过调用处理单元对象的接口函数对业务数据进行处理得到结果数据的步骤还包括：

[0039] 获取中间数据接口类定义；根据中间数据接口类定义将接口函数生成的中间数据封装成中间数据对象，并在处理单元对象之间传递中间数据对象。

[0040] 中间数据接口类定义可以是类文件或头文件。中间数据接口类定义可以是具有一定继承关系的一组抽象类或接口的集合。封装中间数据的过程中，中间数据接口类定义与业务流程执行链具有对应关系，即特定的中间数据接口类定义限定了特定的业务流程执行链中处理单元对象之间传递的数据。

[0041] 中间数据接口类定义对中间数据的传递格式进行了限定。由于传统技术中，开发人员在进行业务流程的逻辑表达的设计时，中间数据的格式通常由开发人员自主设计，因此，导致中间数据的格式因人而异，当多个开发人员联合开发时，最终需要花费较多时间对中间数据的格式进行整合。而通过中间数据接口类定义对中间数据进行封装，可以使开发人员在设计业务流程时，直接采用定义好的具有通用性的中间数据接口类定义对中间数据的格式进行限定，使得联合开发的开发人员编写的代码之间的数据传递得到了规范，不需要后期花费较多时间进行整合，因此，提高了开发效率。

[0042] 例如，在业务流程的执行中，可包括数据库操作（增加、删除、修改或查询）的子过程以及根据数据库操作返回的状态码生成状态信息的子过程。传统技术中，对于不同的开发人员，其根据状态码生成的状态信息可能因人而异。而在本实施例中，由于状态码对应的状态信息已由中间数据接口类定义进行了限定和规范，因此，多个联合开发的开发人员编写的代码中生成的状态信息是相同，也就不需要后期在进行整合，从而提高了开发效率。

[0043] 步骤 S108，根据结果数据生成响应数据并返回。

[0044] 响应数据即最终要返回给 web 服务器，并由 web 服务器通过 HTTP 响应返回，并展示在网页中的数据。

[0045] 在一个实施例中，还可判断接口函数是否抛出异常，若是，则输出并记录异常。

[0046] 进一步的，判断接口函数是否抛出异常的步骤之后还可包括：

[0047] 若是，则判断异常是否需要中止，若是，则中止业务流程执行链的执行。

[0048] 也就是说，处理单元对象的调用处于 CGI 框架的容器中。在依次调用处理单元对象的接口函数时，可检测其是否抛出异常，若抛出异常则获取异常信息输出到控制台或日志文件中。

[0049] 在本实施例中，可获取异常对应的处理单元对象，判断该处理单元对象是否为关键过程，若是，则中止提取处理单元对象，并调用其接口函数的操作，若否，则跳过该处理单元对象，继续提取业务流程执行链上按照顺位的下一个处理单元对象。

[0050] 业务流程执行链定义不仅包括业务流程执行链所包含的处理单元对象，还包括每个处理单元对象是否为关键过程。例如，对于数据库操作的子过程对应的处理单元对象，其通常可被定义为关键过程，若执行失败则中止后续的处理单元对象的调用，优选地，还可对数据进行回滚。而对于日志处理的子过程对应的处理单元对象，其通常可被定义为非关键过程，日志记录失败并不影响业务流程地执行。开发人员可预先通过配置文件将处理单元对象是否为关键过程定义在业务流程执行链定义中。

[0051] 在一个实施例中，在依次提取处理单元对象，调用其对应的接口函数的过程中，还可判断处理单元对象是否为异步对象，若是，则异步调用其对应的接口函数，否则同步调用其对应的接口函数。也就是说，如图 2 所示，若业务流程执行链上顺位的多个处理单元对象均为异步对象，则可以多线程异步调用该多个处理单元对象的接口函数，即该多个接口函数并行执行。若处理单元对象不为异步对象，即为同步对象，则以阻塞的方式在调用该处理单元对象的接口函数得到返回值之后再调用业务流程执行链上顺位的下一个处理单元对象的接口函数。可在业务流程执行链定义中配置处理单元对象是否为异步对象。

[0052] 在一个应用场景中，如图 3 所示，该方法的运行基于程序框架，该程序框架包括 frame、module 和 task 三种类型的对象。module 类型的对象均实现统一的接口，具有相同名称的接口函数。task 类型的对象均继承自统一的抽象类。frame 对象用于接收访问请求提取业务数据并生成 task 类型的对象，然后获取业务流程执行链，并按照顺序提取处理单元对象并调用其接口函数，module 对象则通过对 task 类型的对象进行读写来获取参数并封装生成的中间数据，也就是说通过 task 类型的对象在 module 对象之间传递中间数据。所有 module 对象均被调用完毕后，task 类型的对象中即存储有最终的结果数据。可根据调用完毕后得到的 task 类型的对象生成响应数据。

[0053] 需要说明的是，联合开发的开发人员可根据各自的职能编写业务流程中的子流程对应的 module 类型的对象，优选的，程序语言可采用 c 或者 c++（性能高），也可以采用 python 等脚本语言（易调试），module 类型的对象编写完毕后，可将其注册到 frame 类型的对象中，即配置业务流程执行链定义。

[0054] 在一个实施例中，如图 4 所示，一种基于 CGI 框架的业务流程控制装置，包括：

[0055] 请求获取模块 102，用于获取访问请求，提取所述访问请求中的业务数据；

[0056] 执行链获取模块 104，用于获取业务流程执行链；

[0057] 链式处理模块 106，用于依次提取所述业务流程执行链上的处理单元对象，通过调用其接口函数对所述业务数据进行处理得到结果数据；

[0058] 响应模块 108，用于根据所述结果数据生成响应数据并返回。

[0059] 在一个实施例中，链式处理模块还用于获取中间数据接口类定义，根据中间数据接口类定义将接口函数生成的中间数据封装成中间数据对象，并在处理单元对象之间传递中间数据对象。

[0060] 在一个实施例中，如图 5 所示，基于 CGI 框架的业务流程控制装置还包括执行链生成模块 110，用于获取处理单元对象定义，生成处理单元对象；获取业务流程执行链定义；根据所述业务流程执行链定义通过拼接所述处理单元对象生成业务流程执行链。

[0061] 在一个实施例中，如图 5 所示，基于 CGI 框架的业务流程控制装置还包括异常处理模块 112，用于判断接口函数是否抛出异常，若是，则输出并记录异常。

[0062] 在一个实施例中，异常处理模块 112 还用于判断异常是否需要中止，若是，则中止业务流程执行链的执行。

[0063] 上述基于 CGI 框架的业务流程控制方法及装置，将 CGI 框架中的业务流程代码拆分并封装成对应多个业务流程的子过程的处理单元对象，并按照与实际业务流程对应的业务流程执行链的顺序依次调用处理单元对象的接口函数来实现业务流程的逻辑，使得不同的开发人员在各自实现需要编写的业务流程逻辑的代码过程中，可直接通过向业务流程执行链中添加或注册相应的处理单元对象，使得代码的复用性得到提高，从而在面对新的业务流程需求时，提高了扩展性。

[0064] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机程序来指令相关的硬件来完成，所述的程序可存储于一计算机可读取存储介质中，该程序在执行时，可包括如上述各方法的实施例的流程。其中，所述的存储介质可为磁碟、光盘、只读存储记忆体（Read-Only Memory，ROM）或随机存储记忆体（Random Access Memory，RAM）等。

[0065] 以上所述实施例仅表达了本发明的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对本发明专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本发明构思的前提下，还可以做出若干变形和改进，这些都属于本发明的保护范围。因此，本发明专利的保护范围应以所附权利要求为准。

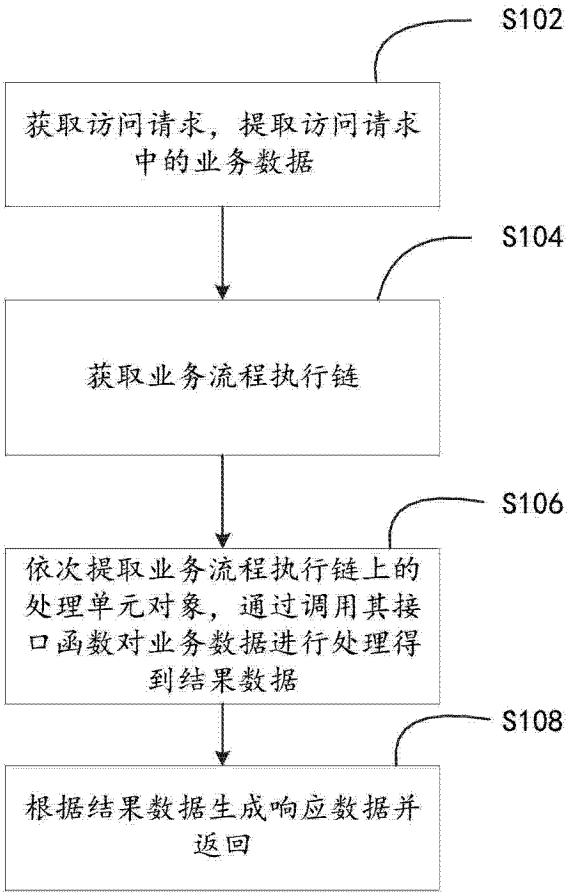

S102
获取访问请求，提取访问请求中的业务数据
S104
获取业务流程执行链
S106
依次提取业务流程执行链上的处理单元对象，通过调用其接口函数对业务数据进行处理得到结果数据
S108
根据结果数据生成响应数据并返回

图 1

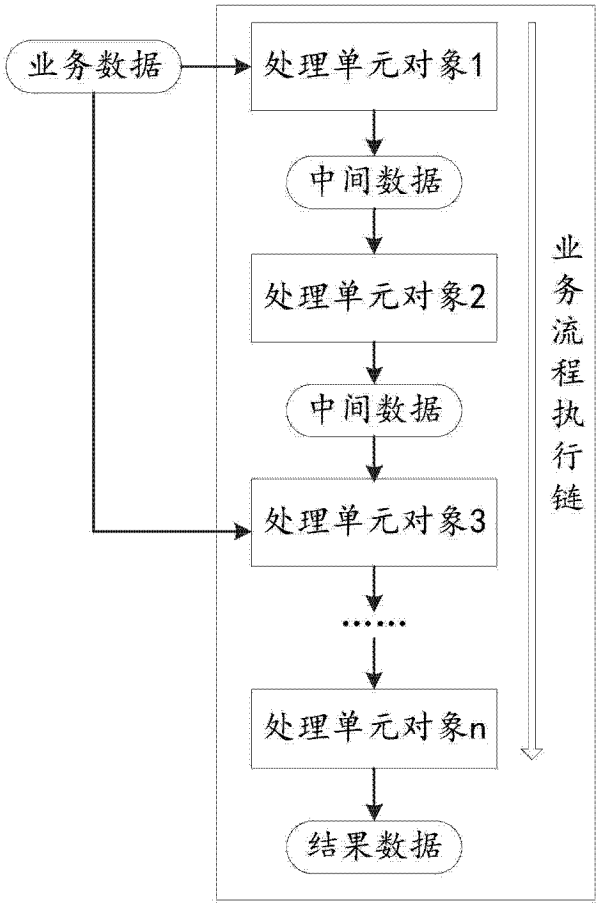

业务数据
处理单元对象1
中间数据
处理单元对象2
中间数据
处理单元对象3
……
处理单元对象n
结果数据
业务流程执行链

图 2

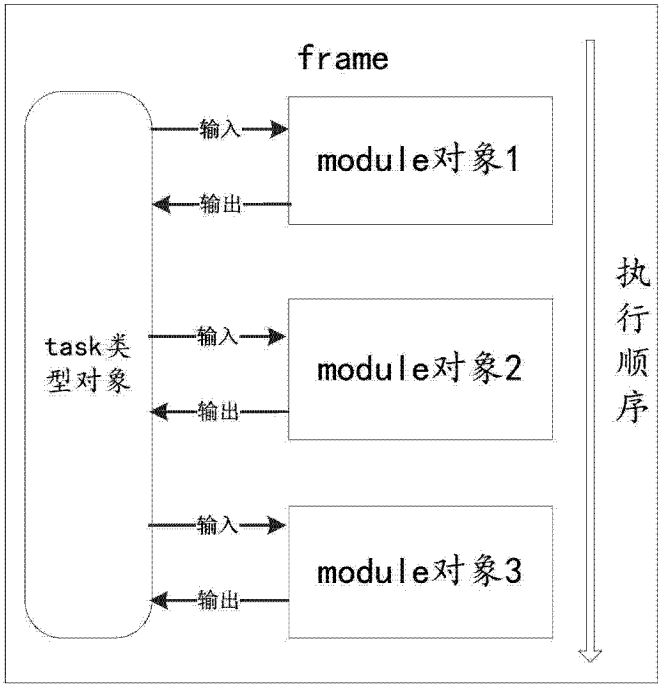

frame
输入
输出
module对象1
task类型对象
module对象2
module对象3
执行顺序

图 3

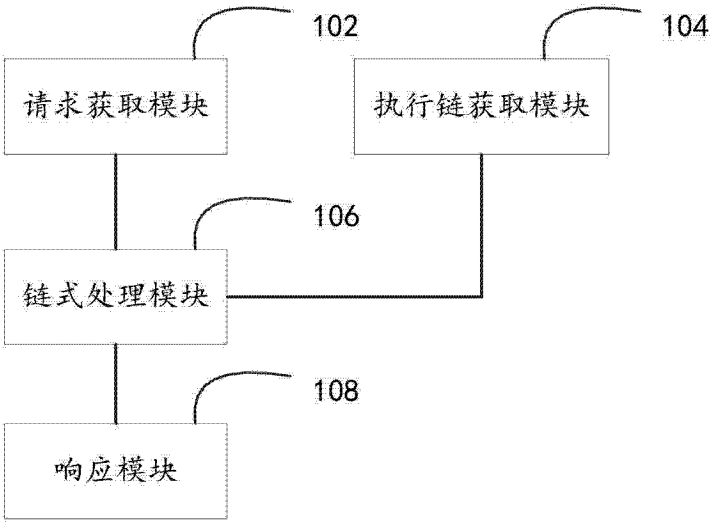

102
104
请求获取模块
执行链获取模块
106
链式处理模块
108
响应模块

图 4

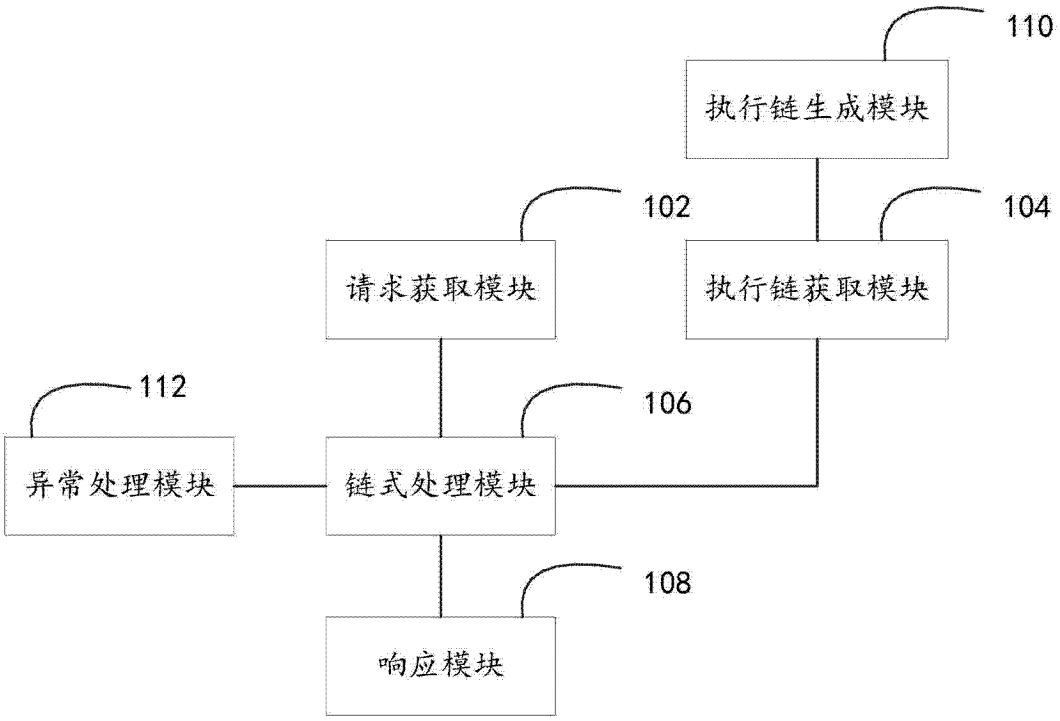

110
执行链生成模块
102
104
请求获取模块
执行链获取模块
112
106
异常处理模块
链式处理模块
108
响应模块

图 5