(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 111367932 B
(45)授权公告日 2023.05.12

(21)申请号 202010151722.2
(22)申请日 2020.03.06
(65)同一申请的已公布的文献号
申请公布号 CN 111367932 A
(43)申请公布日 2020.07.03
(73)专利权人 深圳市今天国际物流技术股份有限公司
地址 518000 广东省深圳市罗湖区笋岗东路1002宝安广场A座10楼F、G、H
(72)发明人 邵健锋 崔巍
(74)专利代理机构 深圳市精英专利事务所 44242
专利代理师 武志峰
(51)Int.Cl.
G06F 16/23(2019.01)
G06F 16/22(2019.01)
(56)对比文件
CN 105677774 A,2016.06.15
CN 1673972 A,2005.09.28
CN 110096518 A,2019.08.06
审查员 程潇杰

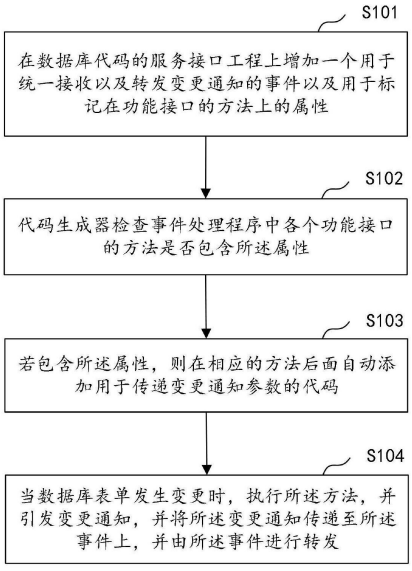
权利要求书2页 说明书9页 附图1页

(54)发明名称

数据库表单变更通知方法、装置、计算机设备及存储介质

(57)摘要

本发明公开了一种数据库表单变更通知方法、装置、计算机设备及存储介质，所述方法包括：在数据库代码的服务接口工程上增加一个用于统一接收以及转发变更通知的事件以及用于标记在功能接口的方法上的属性；代码生成器检查事件处理程序中各个功能接口的方法是否包含所述属性；若包含所述属性，则在相应的方法后面自动添加用于传递变更通知参数的代码；当数据库表单发生变更时，执行所述方法，并引发变更通知，并将所述变更通知传递至所述事件上，并由所述事件进行转发。本发明解决了程序代码对数据库数据变更的感知，并将感知到的变化通知程序的其它部分以实现对变更的响应。



CN 111367932 B

1.一种数据库表单变更通知方法,其特征在于,包括:

在数据库代码的服务接口工程上增加一个用于统一接收以及转发变更通知的事件以及用于标记在功能接口的方法上的属性;

代码生成器检查事件处理程序中各个功能接口的方法是否包含所述属性;

若包含所述属性,则在相应的方法后面自动添加用于传递变更通知参数的代码;

当数据库表单发生变更时,执行所述方法,并引发变更通知,并将所述变更通知传递至所述事件上,并由所述事件进行转发;

所述属性包括:用于标记在功能接口的方法上的第一属性和用于标记在功能接口的方法的参数上的第二属性;所述第一属性表示所标记的方法执行后需引发变更通知;所述第二属性表示标记的参数需包含在变更通知中,并为该参数指定一个别名;

所述将所述变更通知传递至所述事件上,包括:

将变更通知的参数传递至所述事件上,所述变更通知的参数包含:变更通知的名称、参数表和数据库实例的名称;

所述变更通知的参数还包括:方法的返回值、事务令牌、上下文令牌中的一种或几种;

所述第一属性包括:变更通知的名称、需要包含的参数表、是否需要包含返回值、是否需要将返回值也记录为参数;所述第二属性包括:需要被包含在参数表中的参数和该参数指定的别名;

所述将所述变更通知传递至所述事件上包括:

将底层代码引发的变更通知逐层向上传递至上层代码,并最终传递至所述事件。

2.根据权利要求1所述的数据库表单变更通知方法,其特征在于,所述功能接口的方法包括:数据库表单插入和数据库表单更新。

3.一种数据库表单变更通知装置,其特征在于,包括:

增加单元,用于在数据库代码的服务接口工程上增加一个用于统一接收以及转发变更通知的事件以及用于标记在功能接口的方法上的属性;

检查单元,用于代码生成器检查事件处理程序中各个功能接口的方法是否包含所述属性;

添加单元,用于若包含所述属性,则在相应的方法后面自动添加用于传递变更通知参数的代码;

引发单元,用于当数据库表单发生变更时,执行所述方法,并引发变更通知,并将所述变更通知传递至所述事件上,并由所述事件进行转发;

所述属性包括:用于标记在功能接口的方法上的第一属性和用于标记在功能接口的方法的参数上的第二属性;所述第一属性表示所标记的方法执行后需引发变更通知;所述第二属性表示标记的参数需包含在变更通知中,并为该参数指定一个别名;

所述引发单元包括:

第一传递单元,用于将变更通知的参数传递至所述事件上,所述变更通知的参数包含:变更通知的名称、参数表和数据库实例的名称;

所述变更通知的参数还包括:方法的返回值、事务令牌、上下文令牌中的一种或几种;

所述第一属性包括:变更通知的名称、需要包含的参数表、是否需要包含返回值、是否需要将返回值也记录为参数;所述第二属性包括:需要被包含在参数表中的参数和该参数

指定的别名；

所述引发单元还包括：

第二传递单元，用于将底层代码引发的变更通知逐层向上传递至上层代码，并最终传递至所述事件。

4.一种计算机设备，其特征在于，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如权利要求1至2任一项所述的数据库表单变更通知方法。

5.一种计算机可读存储介质，其特征在于，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器执行时实现如权利要求1至2任一项所述的数据库表单变更通知方法。

数据库表单变更通知方法、装置、计算机设备及存储介质

技术领域

[0001] 本发明涉及计算机技术领域，具体涉及数据库表单变更通知方法、装置、计算机设备及存储介质。

背景技术

[0002] 当用户在终端设备(例如应用程序、手持设备、网页等)录入一个表单，并要求按表单出库时，后端的服务器设备即开始执行对应的动作。而用户则希望可以看到表单的出库状态信息，例如每一个货物已经完成的出货量等，为了实现这个查询状态的功能，终端需要在必要的时候将信息重新显示出来，因此，终端需要知道在何时需要执行这个重新显示的动作。具体地，有以下几种方法实现上述功能：

[0003] 一种方法(即方案1)是定时重新刷新，即不论当前执行状态是否有变化，终端按一定的时间间隔，自动重新获取数据，并将该数据显示到界面上；

[0004] 另一种方法是服务器主动将数据推送至终端(即当服务器发现数据变化后，主动将变化的数据推送到终端)，终端则将接收到的数据显示在界面上。而对于第二种方法，服务器需要知道数据库中的数据是否已经发生变更，关于这个问题则通常有三种解决方案：一是通过数据库的自身功能实现，例如利用触发器与数据库反向调用程序(即方案2)；二是使用数据库客户端提供的实时监测功能(即方案3)；三是当出现代码会修改数据库时，主动告知其它程序部分(即方案4)。

[0005] 关于上述四种方案，虽然每一种方案都提出了解决方法，但是在实际操作中，每一种方案都存在一定的缺陷，例如在方案1中，刷新时间难以预测，同时密度高对服务器负载大，密度低信息更新慢。以及由于无法预知数据是否发生变化，刷新产生的数据消耗不一定会产生实际价值，从而降低工作效率；

[0006] 在方案2中，需要在数据库上进行复杂编码，而且反向调用需要用到数据库的高级功能，这对数据库的版本与功能会有明确要求，且触发器与事务同时使用时，会产生诸多的数据问题；

[0007] 在方案3中，可以使用第三方的功能类库(例如github上dyatchenko发布的SqlDependencyEx)或者数据库开发商提供的功能(例如微软的SqlDependence)实现，但是这种方法并不稳定，因为上述组件均不能保证一直有效，且不能保证抓取到所有用户关注的变更；

[0008] 在方案4中，由于数据变更方与终端开发者通常不是同一个开发人，对于数据结构的定义、理解可能会存在一定的差别。而且在多数情况下数据变更方本身并不使用该功能，其开发者对场景的理解与开发测试可能会存在差异以及工作未尽的情况。另外，由于终端开发者的工作较为滞后，并且需要对接多个数据，从而容易导致数据接口的不一致性，增大开发难度。

[0009] 因此，如何消除上述几种方法的缺陷是本领域技术人员需要解决的问题。

发明内容

[0010] 本发明实施例提供了一种数据库表单变更通知方法、装置、计算机设备及存储介质，旨在解决程序代码对数据库数据变更的感知，并将感知到的变化通知程序的其它部分以实现对变更的响应。

[0011] 第一方面，本发明实施例提供了一种数据库表单变更通知方法，所述方法包括：

[0012] 在数据库代码的服务接口工程上增加一个用于统一接收以及转发变更通知的事件以及用于标记在功能接口的方法上的属性；

[0013] 代码生成器检查事件处理程序中各个功能接口的方法是否包含所述属性；

[0014] 若包含所述属性，则在相应的方法后面自动添加用于传递变更通知参数的代码；

[0015] 当数据库表单发生变更时，执行所述方法，并引发变更通知，并将所述变更通知传递至所述事件上，并由所述事件进行转发。

[0016] 进一步的，所述属性包括：用于标记在功能接口的方法上的第一属性和用于标记在功能接口的方法的参数上的第二属性。

[0017] 进一步的，所述将所述变更通知传递至所述事件上，包括：

[0018] 将变更通知的参数传递至所述事件上，所述变更通知的参数包含：变更通知的名称、参数表和数据库实例的名称。

[0019] 进一步的，所述变更通知的参数还包括：方法的返回值、事务令牌、上下文令牌中的一种或几种。

[0020] 进一步的，所述第一属性包括：变更通知的名称、需要包含的参数表、是否需要包含返回值、是否需要将返回值也记录为参数；所述第二属性包括：需要被包含在参数表中的参数和该参数指定的别名。

[0021] 进一步的，所述功能接口的方法包括：数据库表单插入和数据库表单更新。

[0022] 进一步的，所述将所述变更通知传递至所述事件上包括：

[0023] 将底层代码引发的变更通知逐层向上传递至上层代码，并最终传递至所述事件。

[0024] 第二方面，本发明实施例还提供了一种数据库表单变更通知装置，包括：

[0025] 增加单元，用于在数据库代码的服务接口工程上增加一个用于统一接收以及转发变更通知的事件以及用于标记在功能接口的方法上的属性；

[0026] 检查单元，用于代码生成器检查事件处理程序中各个功能接口的方法是否包含所述属性；

[0027] 添加单元，用于若包含所述属性，则在相应的方法后面自动添加用于传递变更通知参数的代码；

[0028] 引发单元，用于当数据库表单发生变更时，执行所述方法，并引发变更通知，并将所述变更通知传递至所述事件上，并由所述事件进行转发。

[0029] 第三方面，本发明实施例还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上述的数据库表单变更通知方法。

[0030] 第四方面，本发明实施例还提供了一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器执行时实现上述的数据库表单变更通知方法。

[0031] 本发明实施例提供了一种数据库表单变更通知方法，所述方法包括：在数据库代码的服务接口工程上增加一个用于统一接收以及转发变更通知的事件以及用于标记在功能接口的方法上的属性；代码生成器检查事件处理程序中各个功能接口的方法是否包含所述属性；若包含所述属性，则在相应的方法后面自动添加用于传递变更通知参数的代码；当数据库表单发生变更时，执行所述方法，并引发变更通知，并将所述变更通知传递至所述事件上，并由所述事件进行转发。本发明实施例解决了程序代码对数据库数据变更的感知，并将感知到的变化通知程序的其它部分以实现对变更的响应。

附图说明

[0032] 为了更清楚地说明本发明实施例技术方案，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发明的一些实施例，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。

[0033] 图1为本发明实施例提供的一种数据库表单变更通知方法的流程示意图；

[0034] 图2为本发明实施例提供的一种数据库表单变更通知装置的示意性框图。

具体实施方式

[0035] 下面将结合本发明实施例中的附图，对本发明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发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，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，都属于本发明保护的范围。

[0036] 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。

[0037] 还应当理解，在此本发明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发明。如在本发明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。

[0038] 还应当进一步理解，在本发明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。

[0039] 下面请参见图1，图1为本发明实施例提供的一种数据库表单变更通知方法的流程示意图，具体包括：步骤S101～S104。

[0040] S101、在数据库代码的服务接口工程上增加一个用于统一接收以及转发变更通知的事件以及用于标记在功能接口的方法上的属性；

[0041] S102、代码生成器检查事件处理程序中各个功能接口的方法是否包含所述属性；

[0042] S103、若包含所述属性，则在相应的方法后面自动添加用于传递变更通知参数的代码；

[0043] S104、当数据库表单发生变更时，执行所述方法，并引发变更通知，并将所述变更通知传递至所述事件上，并由所述事件进行转发。

[0044] 本实施例中，用户只需要将自行编写的事件处理程序(即上述的功能接口的方法)连接在服务接口工程的事件(DatabaseOperationNoticed)上，代码生成器即会对该功能接

口的方法进行检查，判断其是否标记有相应的属性，若有，则在该功能接口的方法后面自动生成相应的代码。当数据库表单发生变更时，数据库表单变更时对应的方法即开始执行，其程序代码即可在代码内部通过传入变更通知引发事件，从而由事件对该变更通知进行转发。

[0045] 本实施例提供的方法相对于背景技术提到的四种方案均存在一定优势，例如相对于方案1，本实施例不必进行定时查询刷新，在无需刷新时则不会主动进行刷新，从而节省服务器资源；又例如相对于方案2，本实施例不需要进行数据库内触发器编写，也不需要使用数据库的高级功能，同时还不依赖高版本数据库组件，进而不影响数据库性能；还例如相对于方案3，本实施例不会出现不稳定的情况，也不会出现变更通知丢失的情况；再例如相对于方案4，本实施例使用自动编码技术，完整实现了方案4的设计目标，但无需用户本身完成方案4所需的工作量，并确保编码符合统一的标准，保障不同组件功能的一致性，从而减少了工作量，降低开发难度，使用户具有更好的体验度。

[0046] 另外，由于程序数据库通常不建议第三方直接修改，且由于本发明实施例提供的方法是基于自动生成技术的，因而可以有效的兼容各种分布式部署模型(分布式部署模型是指程序会被分散成多个独立组件，部署于同一台或多台计算中，因此变更通知依赖于在这些组件之间的有效通讯)。本发明实施例提供的方法基于自动生成技术，生成端只需要为分布式功能编码一次即可，而背景技术提到的方案4则需要开发者每次均手工实现分布式模型的支持，因此在一般使用情况下，相关限制(即当数据库在程序范围之外被修改)对本发明实施例提供的方法并不明显。

[0047] 在一实施例中，所述属性包括：用于标记在功能接口的方法上的第一属性和用于标记在功能接口的方法的参数上的第二属性。

[0048] 本实施例中，将属性标记在用户自行编写的功能接口的方法上，用于代码生成器判断是否在该方法后面添加相应的代码。具体的，所述属性包括第一属性和第二属性，其中，第一属性(DatabaseOperationNotifiableAttribute)标记在方法上，用于说明此方法执行后需要引发变更通知事件；第二属性(DatabaseOperationNotifiableParameterAttribute)标记在方法的参数上，用于说明此参数需要包含在变更通知事件中，并可以为该参数指定一个别名。

[0049] 需要说明的是，标记属性本身不具有任何功能。对于标记第一属性来说，代码生成器的代码会检查每个方法是否有第一属性的标记，并对有第一属性的标记的方法进行相应的处理；对于标记第二属性来说，代码生成器的代码会检查包含第一属性标记的方法的每个参数是否有第二属性标记，并对有第二属性标记的参数进行相应的处理。对方法(或者方法的参数)进行标记，只是用户说明这个方法需要进行上述处理(即在包含属性的方法后面自动添加用于传递变更通知参数的代码)而已。

[0050] 还需说明的是，由于第一属性DatabaseOperationNotifiableAttribute提供了一个常见的、简单的方式，即要求用户在提供其他信息的同时，一次性提供方法所有需要处理的参数，但这种方式存在两个局限性：一是不够直观，不如直接在参数上做标记直观；二是这种方式不能支持参数别名功能，因此在本实施例中，将第二属性作为第一属性的补充存在。而当别名功能无法使用时，一个需要被变更通知包含的参数，既可以在第一属性DatabaseOperationNotifiableAttribute中定义，也可以单独的被第二属性DatabaseOperatio

nNotifiableParameterAttribute标记。

[0051] 在一实施例中，所述将所述变更通知传递至所述事件上，包括：

[0052] 将变更通知的参数传递至所述事件上，所述变更通知的参数包含：变更通知的名称、参数表和数据库实例的名称。

[0053] 本实施例中，将变更通知传递至事件上即是将变更通知中包含的参数传递至事件上。本实施例中的参数(即变更通知的名称、参数表和数据库实例的名称)为变更通知中必须包含的参数。

[0054] 在一实施例中，所述变更通知的参数还包括：方法的返回值、事务令牌、上下文令牌中的一种或几种。

[0055] 本实施例中，当功能接口的方法具有返回值时，则将该方法的返回值包含在变更通知参数中；当功能接口的方法使用事务时，则将该事务的事务令牌包含在变更通知参数中；当功能接口的方法使用数据库上下文时，则将所述上下文的上下文令牌包含在变更通知参数中。当上述场景出现一种或几种时，则可以将对应的一种或几种参数包含在变更通知的参数中。其中，事务令牌和上下文令牌为服务接口工程中定义的概念，用于唯一确定一个正在执行的事务(Transaction)和数据库操作上下文(DbContext)。

[0056] 在一实施例中，所述第一属性包括：变更通知的名称、需要包含的参数表、是否需要包含返回值、是否需要将返回值也记录为参数；所述第二属性包括：需要被包含在参数表中的参数和该参数指定的别名。

[0057] 本实施例中，当第一属性标记的方法执行时，即说明需要传递变更通知引发事件，而变更通知也就是传递变更通知的参数，因此需要将本实施例中的参数进行传递。需要说明的是，变更通知的参数中并不一定需要包含“是否需要将返回值也记录为参数”，换句话说，当用户认为该方法的返回值有必要作为变更通知的内容时，则将该方法的返回值记录为参数，并根据用户的相应规范对其设置一个名称；当用户认为该方法的返回值没有必要作为变更通知的内容时，则不需要将该方法的返回值记录为参数，这个功能可以提高用户统一处理一类数据时的便利性。

[0058] 还需注意的是，“是否需要将返回值也记录为参数”与“是否需要包含返回值”无关。举例来说，当用户处理订单变更时，需要知道被变更的订单实体，在订单新增时，方法的返回值为该订单实体；在订单更新时，参数之一为订单实体。为了方便用户使用统一的处理程序处理来自不同方法的订单变更，则允许在订单新增时，将方法的返回值也作为参数保存在变更通知事件的参数表中，以便处理程序统一提取。

[0059] 在一实施例中，所述功能接口的方法包括：数据库表单插入和数据库表单更新。

[0060] 本实施例中，用户自行编写的功能接口的方法包括了数据库表单插入和数据库表单更新，其中，数据库表单更新可以理解为数据库表单增加和删除，也就是说，当数据库表单发生插入和更新(即增加和删除)时，对应的方法便开始执行，并传递变更通知引发事件。

[0061] 在一实施例中，所述将所述变更通知传递至所述事件上包括：

[0062] 将底层代码引发的变更通知逐层向上传递至上层代码，并最终传递至所述事件。

[0063] 本实施例中，当数据库表单发生变更时，对应的方法(也就是底层代码)即开始执行，同时引发变更通知，而该方法存在多层代码对其分别进行管理，方便用户调用，因此变更通知需要经过所述的多层代码逐层向上传递，并最终到达与用户代码交互的服务接口工

程中，引发事件。

[0064] 在使用本发明实施例提供的方法后，减少了用户的工作量，降低了开发难度，使用户具有良好的体验感。例如在需要发出变更通知的情况时，用户只需要在原有的用于数据库代码生成的接口方法上，增加属性描述即可；在需要监听变更通知的情况时，用户只需处理数据库组件的最外层接口上的变更通知事件即可。

[0065] 举例来说，在一个订单表的操作中，用户增加新的功能中，增加了方法Create、Update和Delete(2项)的变更通知，具体描述如下：

```
[DatabaseOperations]
    public interface IOrderHead
    {
        [DatabaseInsert]
        [DatabaseOperationNotifiable("OrderUpdated", false,
KeyToStoreReturnAsParameter = "Entity")]
        OrderHead Create([DatabaseTransaction]DatabaseTransactionToken
```

[0066]

```
databaseTransactionToken, [DatabaseEntityParameter]OrderHead orderHead);

        [DatabaseUpdate]
        [DatabaseOperationNotifiable("OrderUpdated", false)]
        void Update([DatabaseTransaction]DatabaseTransactionToken
databaseTransactionToken,
[DatabaseEntityParameter][DatabaseOperationNotifiableParameter("Entity")]OrderH
ead orderHead);
```

```
[DatabaseDelete(typeof(OrderHead))]
[DatabaseAdditionalQuery(typeof(OrderHeaderOperation),
nameof(OrderHeaderOperation.PlusOne))]
[DatabaseOperationNotifiable("OrderDeleted", false, "id",
"counterPlusOne")]
void Delete([DatabaseTransaction]DatabaseTransactionToken
databaseTransactionToken, [DatabaseConditionParameter]Guid id, ref int
counterPlusOne);
```

[0067]

```
[DatabaseDelete]
[DatabaseOperationNotifiable("OrderDeleted", false)]
void Delete([DatabaseTransaction]DatabaseTransactionToken
databaseTransactionToken,
[DatabaseEntityParameter][DatabaseOperationNotifiableParameter("Entity")]OrderH
ead orderHead);
...
}
```

[0068] Create要求变更通知名称为OrderUpdated、不包括返回值(第二个参数false)，并且需要将返回值(新建的订单实体)以Entity为名称保存在变更通知的参数表中；

[0069] Update要求变更通知名称为OrderUpdated、不包括返回值(第二个参数false)，并且需要将传入的OrderHead参数以Entity为名保存在变更通知的参数表中；

[0070] 第一个Delete要求变更通知名为OrderDeleted、不包括返回值(第二个参数false)，并且需要将id、counterPlusOne两个参数以参数的原名保存在变更通知的参数表中；

[0071] 第二个Delete要求变更通知名为OrderDeleted、不包括返回值(第二个参数false)，并且需要将传入的OrderHead参数以Entity为名保存在变更通知的参数表中。

[0072] 用DatabaseOperationNotifiable(第一属性)直接标记需要返回的参数，只需要从第三个参数起，依次给定参数的名称即可，但受制于语法格式要求，不能给定这个参数的别名，只能以参数的名字为名保存在变更通知的参数表中。用DatabaseOperationNotifiableParameter(第二属性)直接标记在需要包含的参数上，则可以给出这个参数的别名，用别名为名保存在变更通知的参数表中。

[0073] 用户只需要将自己编写的事件处理程序连接在接口对象(服务接口工程)的DatabaseOperationNoticed(事件)上，其程序代码即可在内部通过对传入的Name(变更通知名称)、Parameters(参数表)、Return(返回值)、DatabaseInstance(数据库实例名)、DatabaseTransactionToken(事务Token)、DatabaseContextToken(上下文Token)，对变更通知做出自己的响应。在上述举例中，订单插入、订单更新都会使用OrderUpdated作为变更

通知,且订单实体都会以Entity为名保存在Parameters中,用户的代码即对此做相同的处理,即可同时适用于订单插入与更新的动作。

[0074] 图2为本发明实施例提供的一种数据库表单变更通知装置200,所述装置200包括:

[0075] 增加单元201,用于在数据库代码的服务接口工程上增加一个用于统一接收以及转发变更通知的事件以及用于标记在功能接口的方法上的属性;

[0076] 检查单元202,用于代码生成器检查事件处理程序中各个功能接口的方法是否包含所述属性;

[0077] 添加单元203,用于若包含所述属性,则在相应的方法后面自动添加用于传递变更通知参数的代码;

[0078] 引发单元204,用于当数据库表单发生变更时,执行所述方法,并引发变更通知,并将所述变更通知传递至所述事件上,并由所述事件进行转发。

[0079] 在一实施例中,所述属性包括:用于标记在功能接口的方法上的第一属性和用于标记在功能接口的方法的参数上的第二属性。

[0080] 在一实施例中,所述引发单元204包括:

[0081] 第一传递单元,用于将变更通知的参数传递至所述事件上,所述变更通知的参数包含:变更通知的名称、参数表和数据库实例的名称。

[0082] 在一实施例中,所述变更通知的参数还包括:方法的返回值、事务令牌、上下文令牌中的一种或几种。

[0083] 在一实施例中,所述第一属性包括:变更通知的名称、需要包含的参数表、是否需要包含返回值、是否需要将返回值也记录为参数;所述第二属性包括:需要被包含在参数表中的参数和该参数指定的别名。

[0084] 在一实施例中,所述功能接口的方法包括:数据库表单插入和数据库表单更新。

[0085] 在一实施例中,所述引发单元204还包括:

[0086] 第二传递单元,用于将底层代码引发的变更通知逐层向上传递至上层代码,并最终传递至所述事件。

[0087] 由于装置部分的实施例与方法部分的实施例相互对应,因此装置部分的实施例请参见方法部分的实施例的描述,这里暂不赘述。

[0088] 本发明实施例还提供了一种计算机可读存储介质,其上存有计算机程序,该计算机程序被执行时可以实现上述实施例所提供的步骤。该存储介质可以包括:U盘、移动硬盘、只读存储器(Read-OnlyMemory,ROM)、随机存取存储器(RandomAccess Memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0089] 本发明实施例还提供了一种计算机设备,可以包括存储器和处理器,存储器中存有计算机程序,处理器调用存储器中的计算机程序时,可以实现上述实施例所提供的步骤。当然电子设备还可以包括各种网络接口,电源等组件。

[0090] 说明书中各个实施例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部分互相参见即可。对于实施例公开的系统而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。应当指出,对于本技术领域的普通技术人员来说,在不脱离本申请原理的前提下,还可以对本申请进行若干改进和修饰,这些改进和修饰也落入本申请权利要求的保护范围

内。

[0091] 还需要说明的是，在本说明书中，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、物品或者设备所固有的要素。在没有更多限制的状况下，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、物品或者设备中还存在另外的相同要素。

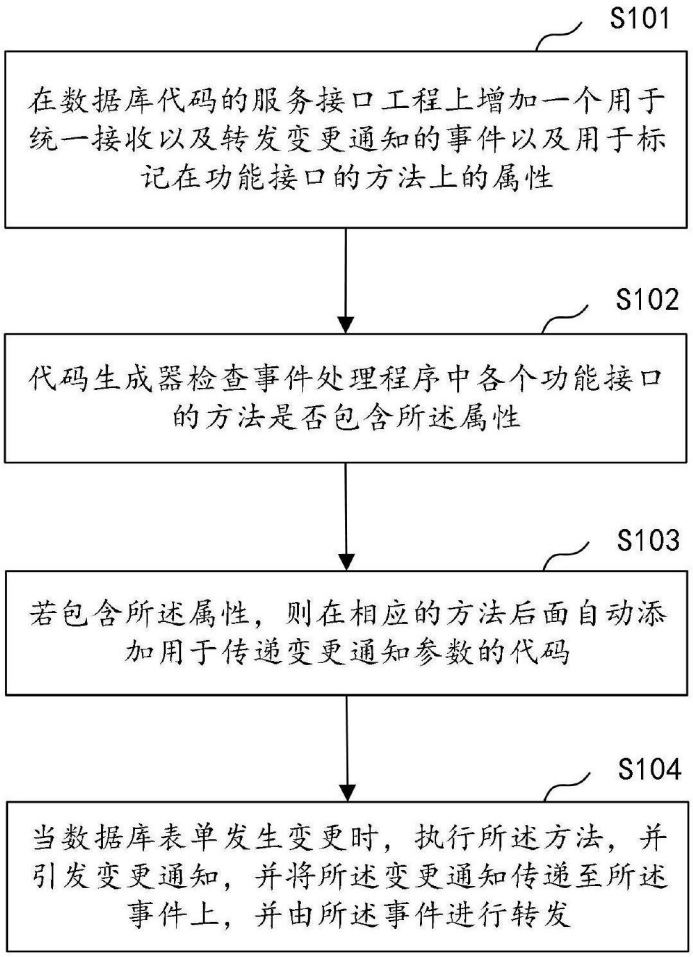

S101
在数据库代码的服务接口工程上增加一个用于统一接收以及转发变更通知的事件以及用于标记在功能接口的方法上的属性
S102
代码生成器检查事件处理程序中各个功能接口的方法是否包含所述属性
S103
若包含所述属性，则在相应的方法后面自动添加用于传递变更通知参数的代码
S104
当数据库表单发生变更时，执行所述方法，并引发变更通知，并将所述变更通知传递至所述事件上，并由所述事件进行转发

图1

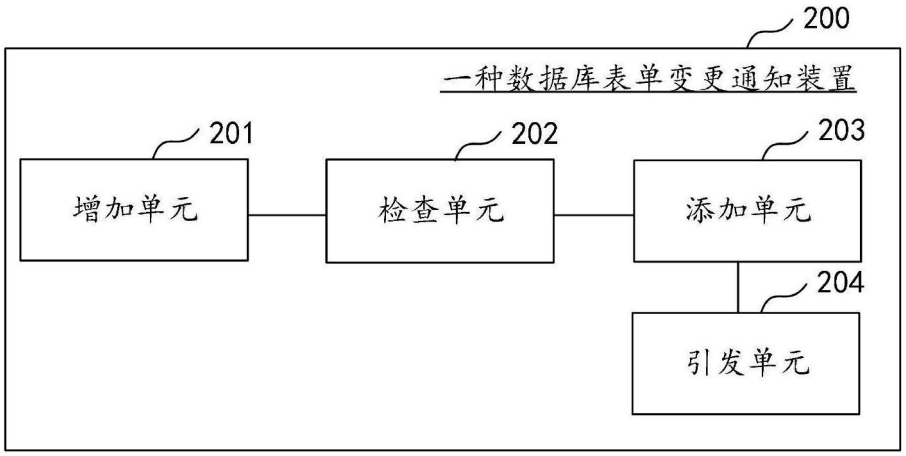

200
一种数据库表单变更通知装置
201
增加单元
202
检查单元
203
添加单元
204
引发单元

图2