

面向对象数据库触发器的实现研究

郑羽洁, 黄远鸣

(广西经济管理干部学院 计算机系, 广西 南宁 530007)

摘 要: 利用 OO(Object-Oriented)语言中事件这一结构, 实现了面向对象数据库触发器事件的注册、订阅及触发过程, 模拟实现了面向对象数据库的触发器功能. 相比其他一些面向对象数据库触发事件, 提出的触发器订阅、引发事件实现方法不会污染用户定义的对象模型.

关键词: 面向对象数据库; 触发器; 触发器的实现; 触发器事件

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2010)05-0217-04

Study for Object-oriented Database Trigger Implementation

ZHENG Yu-jie, HUANG Yuan-ming

(Department of Computer, Guangxi Economic Management Cadre Collage, Nanning 530007, China)

Abstract: The paper study for object-oriented database trigger mechanism, and classify the trigger events. Event structure in OO(Object-Oriented) Language are used for implementing object-oriented database trigger event register, subscribe and trigger procedure, imitating object-oriented database trigger function.

Key words: object-oriented database; trigger; trigger implementation; trigger event

1 引言

触发器在关系数据库中已经得到广泛的应用, 它的作用是当某种数据库相关事件发生时相应地自动完成一些工作. 同样, 相似的功能在纯面向对象数据库中通过向数据库对象池中添加事件监听一样可以轻松实现. 相比其他一些面向对象数据库触发事件, 触发条件及触发动作注入对象类中, 文中提出的实现方法不会污染用户定义的对象模型^[1].

开发 OO(object-oriented) 系统的一个主要原因是为了能够表示结构化复杂对象, 但是对复杂对象的管理又会有一些问题. 例如一个对象有许多成员, 而它的成员又有复杂结构, 要修改这样的复杂对象, 数据库必须向下修改多层的对象, 如果全部检索到内存中会引起性能下降, 此时可以运用触发器来设置要修改的某子层对象; 再如, 如果要删除一个复杂对象, 而它的子对象仍然被其他对象所引用, 此时我们应该设置触发器在删除该对象前检查是否被其他

对象引用, 如果有则停止删除. 此外触发器还有完成数据库初始化处理、控制数据完整性、数据瀑布式修改或删除、统计计算等作用^[2].

2 触发器的实现机制

通过 OO 语言中事件这一结构, 可以轻松实现面向对象数据库的触发器功能. 文中选择用 C# 语言来实现, C# 语言中事件启用了后联编操作, 方法调用和回调过程(callbacks). 在运用事件实现触发器功能过程中, 首先定义各类触发器事件, 然后根据用户需求订阅相关的事件, 当调用对象池中相关操作时即可引发事件, 调用用户自己创建的事件处理程序, 完成触发器功能. 当不再需要该触发器事件的时候, 取消订阅该事件即可^[3].

3 触发器的结构

图 1 表示了面向对象数据库触发器实现的静态结构, 以对象模型向数据库注册触发器事件, 对象删

除之前触发器事件,对象删除之后触发器事件为例介绍触发器实现的静态结构.这三个触发器事件分别覆盖了实现范围为数据库对象池一级,对象一级

的触发器事件,覆盖了功能为类,对象触发器事件,覆盖了执行时间为操作前后引发的触发器事件.

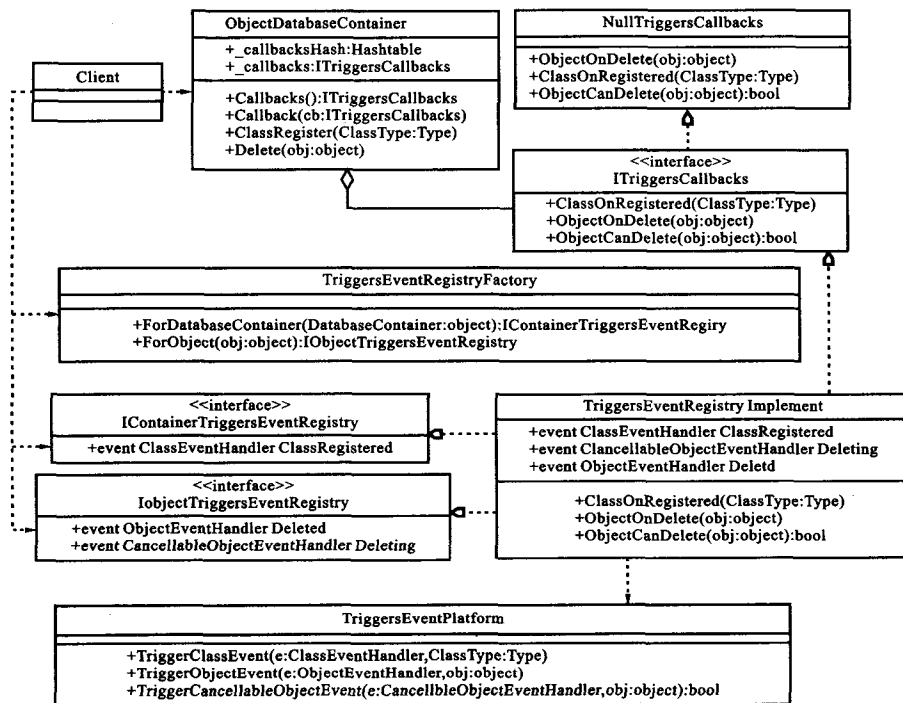


图1 面向对象数据库触发器实现的静态结构

3.1 声明事件处理委托

```

public delegate void ClassEventHandler (object sender,
ClassEventArgs args).

public delegate void ObjectEventHandler (object sender,
ObjectEventArgs args).

public delegate void
CancellableObjectEventHandler (object sender, Can-
cancellableObjectEventArgs args).
  
```

第1个参数 sender 是对引发事件的对象的引用.第2个参数 args 是用于描述事件的 EventArgs 子类对象,通过该对象事件可以向事件订阅者传递附加信息.其中 CancellableObjectEventArgs 类的对象为可以设置取消操作的事件参数,通过调用该对象的 Cancel 方法,可以取消相关事件的操作.

3.2 定义事件

(1)event ObjectEventHandler Deleted, 事件基于 ObjectEventHandler 委托类型,对象删除之后触发器事件,属于对象一级触发器事件.

(2)event CancellableObjectEventHandler Deleting, 对象删除之前触发器事件,属于对象一级触发器事件.

(3)event ClassEventHandler ClassRegistered, 对

象模型向数据库注册触发器事件,属于数据库对象池一级触发器事件.

把声明的事件 ClassRegistered 封装在接口 I-ContainerTriggersEventRegistry 中.这样做可以为数据库对象池获得一个实现 IContainer TriggersEventRegistry 接口的对象实例,通过该对象实例,我们可以获得对象池可以订阅的全部触发器事件.我们也可以为对象池中的每种对象模型获得一个实现 IObjectTriggersEventRegistry 接口的对象实例.

如何获取实现 IContainerTriggersEventRegistry 或 IObjectTriggersEventRegistry 接口的对象实例呢? 我们可以通过触发器事件注册的工厂类 TriggersEventRegistryFactory 来实现.类中实现的静态函数 ForDatabaseContainer (object DatabaseContainer)负责获取对象池可以订阅的全部触发器事件,即返回一个实现 IContainer-TriggersEventRegistry 接口的对象实例.它的参数 DatabaseContainer 是一个对象池对象.类中实现的静态函数 ForObject(object obj)负责获取对象模型可以订阅的全部触发器事件,即返回一个实现 IOb-jectTriggersEventRegistry 接口的对象实例.参数 obj

为一个用户定义的对象实例,通过该对象实例可获得该类对象的模型.如图1所示, IContainer TriggersEventRegistry, IObjectTriggersEventRegistry 两个接口由 TriggersEventRegistryImplement 类实现.

3.3 订阅事件

订阅对象删除之前的触发器事件代码: Deleting + = new CancellableObjectEventHandler (OnDeletingObject). 订阅事件是通过事件加运算符 (+ =)完成的,同样的,取消事件用删除事件运算符 (- =). OnDeletingObject 是用户创建的事件处理函数,该函数接受 CancellableObjectEventArgs 类对象作为参数,如果想取消对象删除操作的执行,在用户定义的事件处理函数中调用 CancellableObjectEventArgs.Cancel() 即可. Cancel() 函数改变 CancellableObjectEventArgs 类对象中布尔型变量 _cancelled 的值为 true,如果 Deleting 事件的引发函数检测到 _cancelled 的值为 true,就会返回 false,当执行对象删除操作时,检测到 Deleting 事件的引发函数返回值为 false,就会停止对象删除操作继续执行.

3.4 引发事件

把事件的引发函数声明封装到接口 ItriggersCallbacks 中,文中声明的引发函数 ClassOnRegistered, ObjectOnDelete, ObjectCanDelete 分别对应事件 ClassRegistered, Deleted, Deleting.

(1)数据库对象池一级的触发器事件引发

类 NullTriggersCallbacks 实现了接口 ItriggersCallbacks,如果实现的函数中没定义任何操作,就不会引发任何事件处理.对象池中的操作,如文中的对象模型向数据库注册操作,要在执行对象模型注册操作代码前调用 ClassOnRegistered 事件引发函数来保证订阅了 ClassRegistered 事件的对象池在执行注册该模型时能自动引发事件,从而实现自动引发用户定义的事件处理函数,如果事件引发函数 ClassOnRegistered 中没实现任何操作代码,就不能引发该事件的处理函数,这就意味着没有为对象池注册触发器事件,就不用额外的操作去调用真正的触发器事件引发函数.

那么如何实现呢?我们为对象池维护了一个实现 ITriggersCallbacks 接口的对象成员变量 _callbacks,为数据库对象池一级的触发器事件提供引发函数,初始化为 NullTriggersCallbacks 对象.如果数

据库对象池获得了可订阅的触发器事件,那么在获得触发器事件的同时,对象池成员变量 _callbacks 被实现了真正引发事件的 TriggersEventRegistryImplement 类对象赋值,就可以实现事件引发功能.

(2)对象一级的触发器事件引发

如何实现对象池中对象一级的触发器事件的引发呢?我们为对象池维护了一个 Hash 表成员变量 _callbacksHash,键名为对象模型,文中用对象的类名模拟,键值为该对象模型对应的实现 ITriggersCallbacks 接口的对象实例.只有注册了触发器事件的对象模型才可以填加进 Hash 表,这样在执行对象操作的时候,先检查该对象模型是否在 Hash 表中,如果不在则不调用事件引发函数.有了这个 Hash 表,就可以为每种对象模型提供对象一级触发器事件,在 Hash 表中的每种对象模型对应的 ItriggersCallbacks 接口实现对象初始化为 TriggersEventRegistryImplement 对象.

TriggersEventPlatform 类实现的是根据触发器事件的功能,把事件引发函数分类的作用.例如,针对类触发器事件把相应的事件引发函数放在同一个函数 TriggerClassEvent 中实现,针对查询触发器事件把相应的引发函数放在 TriggerQueryEvent 中实现,这样为不同功能的触发器事件引发提供代码重用.

4 触发器的交互

图2描述了对象模型向数据库注册事件的注册,订阅及触发过程.当用户执行对象模型向数据库注册(ClassRegister)操作时,引发了事件,并动态调用事件处理函数 OnRegisteredClass.

图3描述了对象删除之前,之后两个触发器事件的注册,订阅及触发过程.当操作之前事件处理函数执行完毕后,事件引发函数询问 CancellableObjectEventArgs 类的对象是否满足终止条件,如果不满足则继续执行操作,如果满足则终止操作.本例中对象删除之前触发器事件的处理函数(OnDeletingObject)没调用 CancellableObjectEventArgs 类对象的 Cancel 操作(即不满足终止条件),所以程序没有退出 Delete 操作,对象删除之后触发器事件才可以引发.

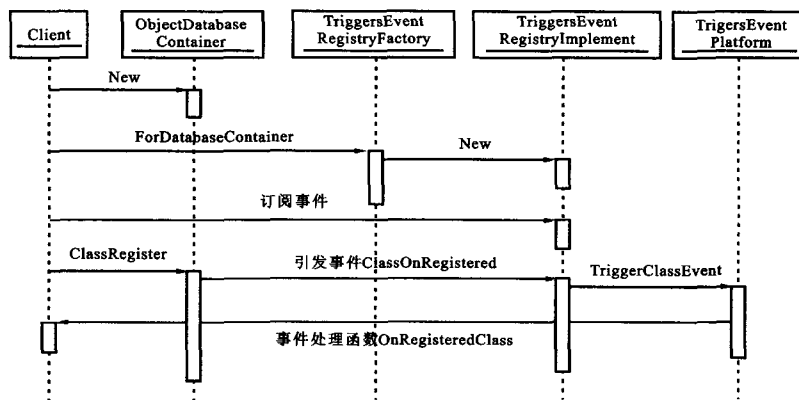


图2 数据库对象池一级触发器事件的注册、订阅及触发过程

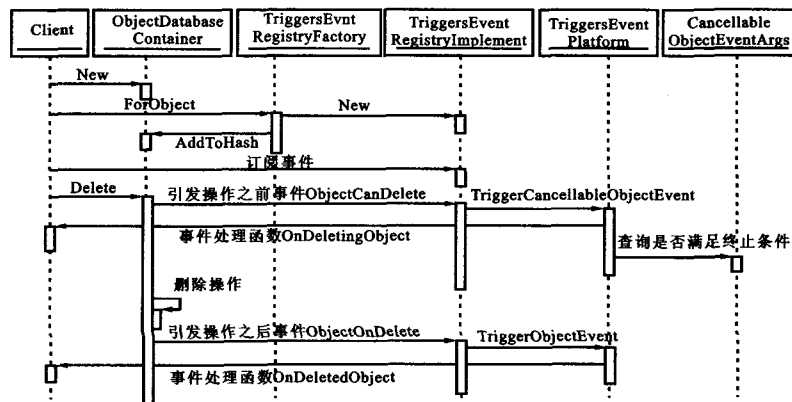


图3 对象一级触发器事件的注册、订阅及触发过程

5 结束语

面向对象数据库中引入触发器机制,可以使数据库拥有更精细,更复杂的数据控制能力.文中提出一种触发器的实现方案,即不会污染对象模型,同时又可以进行多层分级触发,进一步细化了触发事件.在性能方面,借用嵌入式的面向对象数据库引擎,可以轻松实现复杂度很大的触发事件处理函数,这是关系数据库触发器所没有的.

参考文献:

[1] Ramez Elmasri, Shamkant B Navathe. 数据库系统基础

[M]. 邵佩英,徐俊刚,王文杰,译.5版.北京:人民邮电出版社,2008:61-75.

[2] Clifton Nock. 数据访问模式-面向对象应用中的数据库交互[M]. 鄢爱兰,王安鹏,译.北京:中国电力出版社,2004:101-124.

[3] Jim Paterson, Stefan Edlich, Henrik Horning, et al. The definitive guide to db4o[M]. New York: Springer-Verlag, 2006:314-320.

作者简介:

郑羽洁 女,(1979-),硕士,讲师.研究方向为面向对象数据库.

黄远鸣 男,(1973-),副教授.研究方向为数据挖掘.