DOI:10.13203/j.whugis2002.06.019
第27卷第6期 武汉大学学报·信息科学版 Vol.27 No.6
2002年12月 Geomatics and Information Science of Wuhan University Dec.2002

文章编号:1000-050X(2002)06-0642-05 文献标识码:A

主动实时数据库系统触发器模型的研究

张沪寅[1] 陈 珉[1] 文小军[1] 吴建江[1]

(1 武汉大学计算机学院,武汉市珞喻路129号,430079)

摘 要:在主动和实时数据库理论的基础上,提出了一种支持现代应用的主动实时数据库系统(ARTDBS)的触发器机制,并基于Windows NT平台实现了触发器的执行。

关键词:主动数据库;实时数据库;触发器;线程

中图法分类号:TP311.13;TP332

传统的数据库系统只是处理永久性数据,其设计和开发的主要目的是维护数据的正确性及一致性,而不考虑与数据处理相关的定时限制,因而,无法满足实时应用环境的需要。而实时系统所支持的应用具有很强的时间性要求,其任务具有定时限制,所处理的数据也是实时的。因此,将数据库与实时系统相结合,集成两者的概念和功能建立实时数据库系统(RTDBS),能同时满足定时性和一致性要求,有效地支持各种实时应用。

现在实时应用的另一个普遍要求是DBMS能自动监视关于数据库和外部环境的状态。当一定的情形出现时,能自动、实时地作出相应的反应,即执行特定的活动,也就是要求系统有主动能力。因而,现代数据库应用的主动性要求促使了主动数据库系统(ADBS)的研究,而实时性和主动性的结合导致了主动实时数据库系统(ARTDBS)的产生与发展。

1 触发器的概念

主动数据库具有主动性的关键是触发器机制,但并非有了触发器就是主动数据库。早在Codasyl数据库中,其ON条件就有触发器的思想。System R中则显式地使用了Trigger和Assert命令。然而,这些触发器本身是系统设定的,不能随意定义与动态维护。因而,传统数据库的触发器常常仅用于数据的完整性检验,根本无法满足主动实时应用的要求。本文中研究的ARTDBS触发器引入了识时机制,支持时间事件及对多种事件的探测,可由用户定义各种复杂的条件。当多个触发器被同一事件触发时,可以按一定的优先级进行条件评价,也可按一定的优先级选择被触发的活动。

ARTDBS的触发器采用了<情形:活动>模式,触发器的说明为:

Trigger::=Trigger<T-Id>(<Situation-spec>,<Action-spec>)

2 触发器库的设计

触发器库(T-B)是实现主动机制的数据结构之一,其结构如图1所示。

由于事件和触发器之间存在着一对多的关系,为此采用了二级索引链表的形式来组织触发器结构。在触发器类型索引(trigger-kind-index)入口有一指针,指向事件触发器链表(event-trigger-list)。由于每一个事件可能触发多个触发器,因此对事件链表中的每一个事件用一个指针指向其对应的触发器链表Trigger-List,触发器链表中按照条件评价的优先级顺序存放着对应于该事件的多个触发器项,每个触发器中都有情形限制和被触发活动。被触发活动部分比较简单,可以放在触发器中。情形限制的条件谓词(predicate)则可能是个复杂表达式,因此用指针PRED将它连到触发器表中。触发器链表以单循环链表的方式构成,以便对触发器库进行动态维护。

收稿日期:2002-08-18。

项目来源:国家测绘局测科科技发展基金资助项目(98012)。

3　触发器的执行模型

ARTDBS 的触发器主要包含事件探测器、事件处理器、情形评价器及触发监控器 4 个部分。其执行模型如图 2 所示。

事务的某些操作原语、事务原语或者外部请求原语及时钟向触发器监控器发消息。触发器监控器收到信息之后, 就将触发事务挂起, 并启动事件探测器。事件探测器在探测到一个事件并作出相应处理后, 就向触发器监控器发信号, 然后, 触发监控器调用情形评价器, 情形评价器按 EV-PRIORITY 的优先级顺序评价 所有被激活触发器(可能有多个触发器)中的情形。在有的情况下, 情形可能没有限制, 这时条件谓词就是恒真, 此时触发器的语义为: 当特定事件发生时就启动被触发的活动的执行。如果评价结果为真, 即相应情形出现, 触发器监控器就点燃对应的触发器。点燃一个触发器, 并不意味着立即开始执行被触发活动, 何时开始执行依赖于触发活动与触发事务的关系及"情形-活动匹配关系"。最后解挂触发事务, 恢复正常运行。

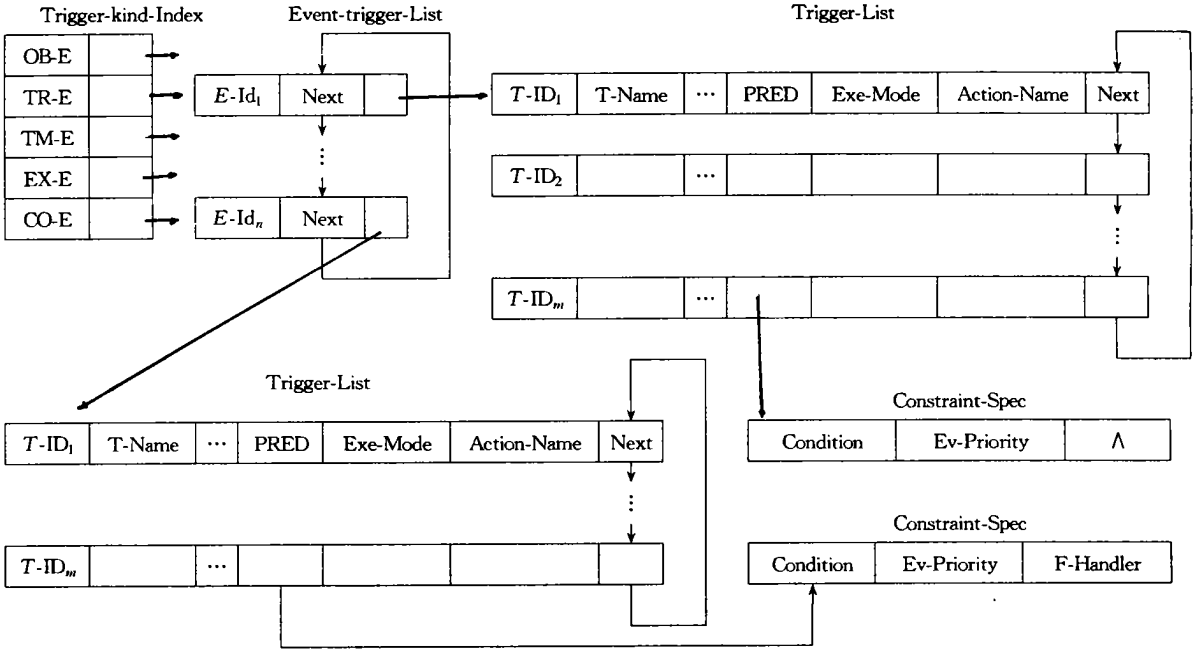

图 1　触发器库结构

Fig. 1　Structure of Trigger

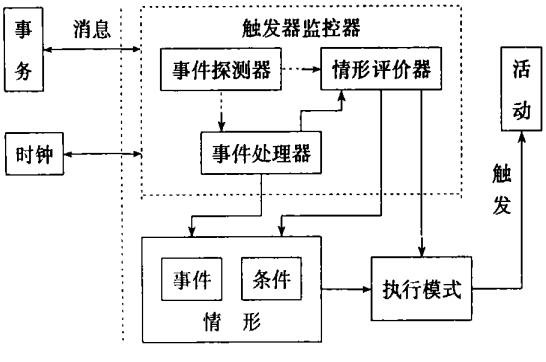

图 2　触发器的执行模型

Fig. 2　Execution Model of the Trigger

4　执行模型的设计方法

此触发器的设计是基于 Windows NT 4.0, 以 Delphi 4.0 为开发工具, SQL Server 7.0 为后台数据库, 利用多线程机制来设计。

4.1　多线程机制在主动实时数据库中的应用

线程是一个应用程序中的一条基本的执行路径, 也是进程中的最小执行单元。一个进程由一个或多个线程、代码、数据等组成, 每个进程都分配给它一个或多个线程。Windows 把极短的一段时间分配给这些线程。由于 ARTDBS 是在 Windows NT 平台上实现, 其触发器执行模型充分利用了 Windows NT 支持多线程的特点, 在实现时引入了多线程机制。利用多线程机制, 主动实时

数据库系统可以高效地响应各种事件的产生，可以同时进行多事件的处理和评价，触发同事件相关的触发器。

当触发器机制在初始化完成后，启动触发器监控器(主控线程)。作为一个集中控制的主体，触发器监控器的主要功能是创建 4 种线程并协调各线程间的通信。

1) 时钟线程。确定定时器的大小，用于控制时间事件的探测。创建时钟线程：

threadTimer := TtimerThread.Creat(true);

TtimerThread 是标准 Delphi 线程类 Tthread 的派生类。Creat 函数用于创建线程对象，其参数 True 表明该线程创建时立刻执行。

2) 非时间事件探测线程，用于控制非时间事件的探测。创建非时间线程：

threadNonTimeInspector := TNonTimeInspectorThread.Creat(Falst);

TnonTimeInspectorThread 是标准 Dephi 线程类 Tthread 的派生类。Creat 函数用于创建线程对象，其参数 False 表明该线程创建时立刻挂起。

3) 时间事件探测线程，用于控制时间事件的探测。

threadTimeInspector := TTimeInspectorThread.Creat(False);

4) 情形评价线程，用于进行情形评价。

threadEvaluate := TTimeEvaluateThread.Create(False, Message);

另有两个 Windows NT 同步事件——时钟线程同步事件和时间事件探测同步事件，用于协调时间事件的探测。

1) 时钟线程同步事件的创建。

evHTimeSynEvent := Windows.CreateEvent(nil, False, False, nil);

2) 时间事件探测同步事件的创建。

evHTimeDetEvent := Windows.CreateEvent(nil, False, False, nil);

利用 Windows NT 事件同步机制对时钟线程和时间事件探测线程进行同步。在触发器机制初始化时，还创建一个时间事件队列和发生事件表，结构如下。

时间事件队列的结构：

```
TEQ-Item= Record
  Ev-ID: Smallint;
  Time-Ev-Type: Char;
  Period: Smallint;
  Occ-Time: Longint;
  Com-Ev-Flag: Longword;
  Next: ^TEQ-Item;
End;
```

发生事件表的结构：

```
Event-List-Item= record
  EvName: ShortString;
  Ecode: Longword;
  Tm-Cons-Flg: Char;
  Occ-Time: Longint;
End;
```

为了在各线程及进程传递事件消息，监控器为每一种基本事件创建了一个消息队列，即对象事件消息队列、事务事件消息队列、外部事件消息队列、时间事件消息队列。一旦触发器监控线程探测到有消息到达时，就可以唤醒相应的探测线程。

在使用多线程存取消息队列时，会涉及到多个线程同时访问同一部分数据(如相同的文件，DDL，通信资源等)。本设计利用 Windows NT 提供的临界区(Critical Section)来加以控制。触发器执行模型的实现结构如图 3 所示。

4.2 执行模型的实现

4.2.1 非时间事件的探测

在主动实时数据库系统中，产生事件的事件源是以消息的形式向系统发送的。当数据操纵子系统、事务管理子系统及 I/O 操作子系统向消息队列发送事件消息后，触发器监控器(主控线程)创建非时间事件探测线程：

NonTimeInspectorList[i] := TNonTimeInspectorThread.Create(True);

其中参数 True 表明该线程创建时立即执行，i 表示创建第 i 个非时间探测线程。

对象事件、事务事件及外部事件探测器从各自的消息队列中取出消息，并根据消息的有关参数确定该事件是否为事件库中已定义的事件。如果是，表示触发事件发生，将事件的发生时间及有关参数写入到发生事件表中，并发消息给主控线程进行情形评价。

4.2.2 时间事件的探测及处理

对于在系统初始时就已确定的时间事件(例如“AT 18:00:00, 99-01-01”，“5s AFTER 16:00:00, 99-02-12”)，当触发器监控器启动时即创建两个用于时间事件探测的线程——时钟线程和时间事件探测线程，同时创建两个同步事件——时钟线程同步事件和时间事件探测事件，并且将时间同步事件设为有信号。两个线程在创建后将自己挂起，等待各自的同步事件变为有信号。由于时

钟线程同步事件最初被设为有信号, 因此时钟线程首先醒来投入运行。它总是取 TEQ 中第一个元素, 根据其发生时间设置定时, 当定时时间到, 定时器向时间事件探测器发信号, 即置时间事件探测同步事件为有信号, 使得时间事件探测线程醒来投入运行, 同时时钟线程重新将自己挂起。当一次时间事件探测结束时, 则重新置时钟线程同步事件为有信号, 将它再一次唤醒并开始记时, 同时将时间事件探测同步事件挂起。

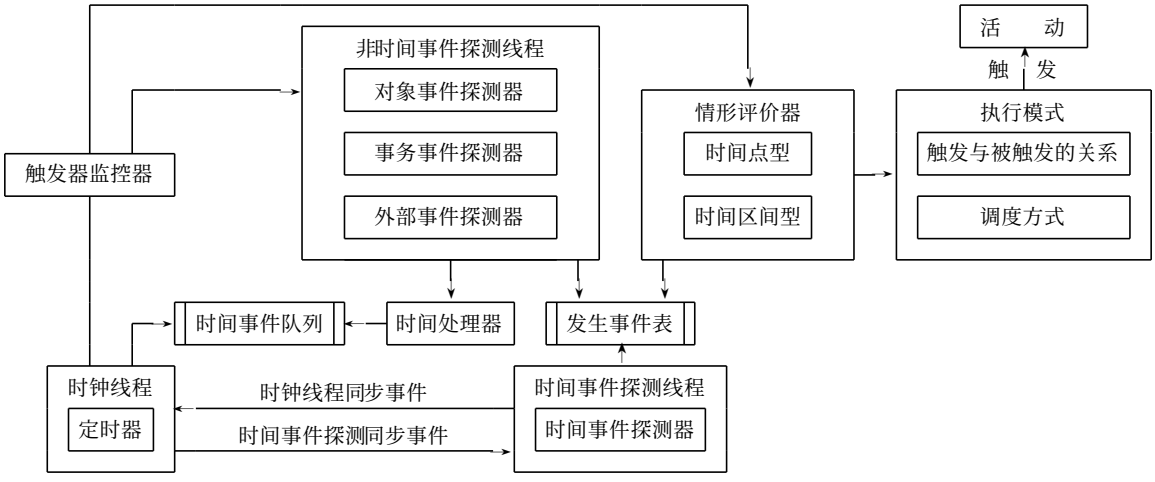

图 3　触发器执行模型的实现结构

Fig. 3　Structure of Trigger Model

对于包含基本事件或复合事件的时间事件(“E_3 IS 5s AFTER E_1”, “E_4 IS AFTER E_2 EVERY 10s”), 由于其发生事件依赖于其他事件的发生, 因此, 只能在系统运行过程中决定其发生时间, 所以这类时间事件则在系统运行时, 在适当时间插入 TEQ。在任何时候, TEQ 都是有序的, 即事件发生时间的先后顺序, 这类事件由时间处理器来处理。

当事件探测器探测到事件发生时, 若该事件为某一时间事件的组成部分, 则启动时间事件处理器, 时间事件处理器检查事件库, 找到已发生事件的时间事件, 并通过对时间操作符的处理, 将得出包含有基本事件或复合事件的时间事件的发生时间, 然后将该时间事件插入到时间事件队列 TEQ 中去, 而达到定时并探测的目的。

4.2.3　*情形的评价及被触发活动的执行*

事件探测器在探测到相应事件后, 向触发器(主控线程)发消息, 通知情形评价器进行条件的评价和处理。

当事件探测到事件发生后, 则利用 PostMessage 函数发送消息给主控线程中的消息响应函数(procedure CMEvaluateEvent(var Message: TMessage);)

PostMessage(Fmain. handle, WM-EVALUATEEVENTMESSAGE, 0, 0);

式中, 第一个参数是接收此消息的句柄即主控线程的窗口句柄; 第二个参数 WM-EVALUATTEVENTMESSAGE 是消息名称; 第三个参数是消息参数: 0 表示请求评价时间事件, 1 表示请求评价非时间事件; 第四个参数是消息参数。在本系统中为一指针, 它所指向的内存块中含有探测器传给评价器的参数。0 表示无效指针, 即不需要参数。

在本系统的实现中, 消息响应函数通过判断 TMessage 中的值来区别要进行时间事件评价还是进行非时间事件评价, 然后创建条件评价器线程进行评价。

1) 非时间条件的评价。对于非时间条件的评价, 可根据触发器库中条件谓词进行评价。当条件评价为真, 则激活触发器, 否则进行失败处理。

2) 时间条件的评价。对于时间事件的评价, 则要根据时间参数来确定, 是时间点型还是时间区间型。其评价应采用相应的时间点型评价器和时间区间型评价器。时间条件评价所用到的最重要的数据结构是在主动机制子系统初始时创建的发生事件表, 其中记录了已定义用户事件发生时间的相关信息。

3) 被触发活动的执行。对于一个主动触发器, 当它的条件满足时, 有执行与之相联的活动。被触发的活动可以是触发事务本身的一部分, 也可以是其子事务或独立的事务。例如“考核评定”

是一触发事务,对于任何人,若“考核评定连续三次为优”非时间条件的评价为真,则“加薪”活动被触发,但该活动显然是一个与“考核评定”事务独立的事务,且不必在“考核评定”的结尾处执行,更无需在当时(评定通过)执行,完全可以单独地执行。

参 考 文 献

1 Stonebraker M. The Integration of Rule System and Database System. IEEE Trans. on Knowl. and Data Eng., 1992, 14(4): 415~423

2 Haritsa J R, Carey M J, Livey M. Earliest Deadline Scheduling for Real-time Database Systems. The Real-Time System Symposium, San Antonio, 1991

3 Hson S. Scheduling Real-Time Transactions Using Priority. Information and Saftware Technology, 1992, 34(6): 409~414

4 Stonebraker M, Hanson E N, Potaimianos S. The Postgers Rule Manager. IEEE Trans. Software Eng., 1988, 14(7): 884~915

5 朱 冰,梅 宏,杨芙清. 基于事件驱动的主动对象模型. 软件学报, 1996, 7(3): 145~149

6 刘云生. 一个实时主动数据库的触发器机制. 计算机研究与发展, 1997, 34(1): 33~37

7 刘云生,胡国玲,舒嘉雄. 一个主动实时内存数据库系统. 华中理工大学学报, 1996, 24(3): 31~34

8 卢炎生,韩 琪. 主动实时数据库事务及其处理. 计算机研究与发展, 1998, 35(2): 188~191

作者简介:张沪寅,副教授. 现主要从事计算机网络与通信、网络安全的研究。代表成果:主动实时数据库系统主动机制模型的研究。

E-mail: zhy2536@sohu.com

A Trigger Model of Active Real-time Database System

ZHANG Huyin[1] *CHEN Min*[1] *WEN Xiaojun*[1] *WU Jianjiang*[1]

(1 School of Computer, Wuhan University, 129 Luoyu Road, Wuhan, China, 430079)

Abstract: On the basis of the theory on active and real-time database, this paper suggests an active real-time database system ARTDBS supporting advanced application, which perfectly integrates traditional DBMS capabilities, active capabilities, and real-time constraint mechanism. The active mechanism in ARTDBS is the trigger mechanism. Effective work of the triggers depending on design of the system execution model isanalyzed in detail, and an execution model of the trigger is developed based on WINDOWS NT environment.

Key words: active database; real-time database; trigger; thread

About the author: ZHANG Huyin, associate professor. His major research fields include computer network & communication network security. His typical achievement is the research on the active mechanism model of active real-time database system.

E-mail: zhy2536@sohu.com