

# ORDBMS 中主动性规则的研究与实现\*

李红燕<sup>1,2</sup>, 李战怀<sup>3</sup>, 唐世渭<sup>1,2</sup>

<sup>1</sup>(北京大学 信息科学中心, 北京 100871);

<sup>2</sup>(北京大学 视觉与听觉信息处理国家重点实验室, 北京 100871);

<sup>3</sup>(西北工业大学 计算机科学与工程系, 陕西 西安 710072)

E-mail: lihy@cis.pku.edu.cn

http://www.pku.edu.cn

**摘要:** 归纳了 ORDBMS(object-relational database management system) 主动性规则的一般描述形式, 对规则各种语义、规则继承、规则操作、多规则触发以及规则的级联触发进行了全面的探讨, 提出并实现了一种适用于 ANGEL 系统的简化 E-C-A(event-condition-action) 模型。该模型提供了对“更新/更新”、“查询/更新”、“更新/查询”和“查询/查询”这 4 类规则的支持, 对 C-A 分离耦合模式以及对冲突执行触发器的优先级定义手段, 并通过在触发约束表示中将 NEW 与 OLD 扩展为一种路径量词而提供了对集合约束的表示和实施能力。另外, 还讨论了该模型中基于事务的规则行为分析方法。

**关键词:** ORDBMS(object-relational database management system); E-C-A(event-condition-action) 模型; 规则; 触发器

中图法分类号: TP311

文献标识码: A

通过规则机制实现主动性是 ORDBMS(object-relational database management system) 的基本特征之一<sup>[1]</sup>。尽管功能强大的规则系统可支持非常复杂的 DBMS 应用, 但事实上, 规则的聚合与终止等问题往往很难判断, 这使得用户对规则的使用望而生畏<sup>[2]</sup>。同时, 要实现文献[1]所阐述的规则系统需要较大的系统开销。

目前, 在商用数据库系统中大都采用触发器来实现规则系统的部分功能。本文在全面分析与总结 ORDBMS 主动性规则的描述形式和各种语义的基础上, 考虑到方便用户使用和利于系统高效实现也是实现 DBMS 的关键因素, 因此在 ANGEL 这个由西北工业大学软件工程中心完全从底层研制开发的 ORDBMS 中, 提出了一种实现规则机制的新途径: 立足于传统 RDBMS 的触发器机制, 并对其进行扩展以尽可能对规则系统提供更多的支持。

ANGEL 实现的简化 E-C-A(event-condition-action, 简称 E-C-A) 模型对 SQL3<sup>[3]</sup> 中定义的触发器机制作了如下扩展: 允许触发事件及触发动作为查询语句, 实现了“事件”和“动作”分别是“查询”或“更新”这 4 种组合的规则形式, 对审计跟踪、警报器、虚数据元素及灵活精细的安全性保护<sup>[1]</sup> 提供了支持; 提供了 C-A 分离耦合模式; 允许直接在子表模式上创建触发器; 实现了在其他 DBMS 中难以实现的对对象集合实施约束的功能; 提供了对处理触发器冲突执行的优先级定义手段。

\* 收稿日期: 2000-05-17; 修改日期: 2001-02-27

基金项目: 国家自然科学基金资助项目(60073055); 国家重点基础研究发展规划 973 资助项目(G1999032705)

作者简介: 李红燕(1970—), 女, 重庆人, 硕士, 主要研究领域为数据库系统, Internet 应用软件; 李战怀(1961—), 男, 陕西旬邑人, 博士, 教授, 主要研究领域为数据仓库, Web 数据管理, 实时数据库; 唐世渭(1939—), 男, 浙江镇海人, 教授, 博士生导师, 主要研究领域为数据库, 信息系统。

## 1 ORDBMS 主动性规则的研究

数据库系统的主动性规则研究始于 20 世纪 70 年代中期. 目前, 绝大多数 DBMS 产品都引进了一些有限的主动规则功能, 并将其作为自身的主要优势之一. 规则还作为一个显著特征出现在 SQL3 标准中<sup>[3]</sup>.

对规则的描述机制大多采用 E-C-A 模型, 其基本形式为: on 事件 if 条件 do 动作. 其中, 事件-条件-动作是规则的 3 个基本组件. 尽管有的系统采用 S-A 模型来描述规则, 但 S-A 模型是将事件和条件统一到情形这个概念之中, 实质上, S-A 与 E-C-A 在描述能力方面是等价的.

### 1.1 规则描述

要完整地描述 ORDBMS 中一个已存在的规则, 应该采用七元组形式:  $RULE ::= (E, C, A, E, C, C-A, G, S)$ . 其中 E 表示能够激活规则的触发事件; C 表示规则的触发约束条件, 用布尔表达式表示. 按照触发约束作用范围的不同, 可以将其细分为: 单对象约束(同一个对象中不同属性值之间应满足的约束条件)、多对象约束(多个对象之间由于存在某种联系而需满足的限制条件)、单对象集合约束(描述某个对象集合的特征)以及多对象集合约束(描述多个对象集合之间所应维持的特定联系); A 表示当触发事件发出并且触发条件为真时, 活跃规则所应执行的动作.

E-C 用于指定规则触发事件与触发条件之间的耦合模式. E-C 耦合模式有 3 种: 立即式(immediate)、延迟式(deferred)和分离式(separate). “立即”耦合的含义很自然, 而“延迟”和“分离”的区别在于: “延迟”是在本事务提交前进行条件判定, “分离”则是在另一个事务中进行条件判定. 根据两个事务之间是否存在提交依赖关系, 分离模式又细分为分离提交依赖式和分离非提交依赖式两种; C-A 指定规则条件与动作之间的耦合模式, 也有上述 3 种.

G 表示规则触发粒度, 可有语句触发和元组(行)触发两种; S 表示规则的当前状态. 每个规则应有活跃和休眠两种状态. 只有当规则处于活跃状态时才能够被激活, 当其处于休眠状态时, 即使触发事件发生, 触发约束条件为真, 它也不会执行相应动作.

### 1.2 对规则的操作

在 ORDBMS 中, 规则作为一种数据库对象, 可以被施加创建、更新及删除操作. 此外, 还有下述特殊操作可以影响对规则的处理: ① 触发操作, 又称激活操作或点火操作, 用于评价规则的条件, 并当条件满足时, 执行规则动作. 此操作的执行要随相应耦合模式而定; ② 使能操作, 用于将规则置于能够被激活的活跃状态; ③ 使不能操作, 用于将规则置于不能被激活的休眠状态.

当然, 对规则的操作还应满足事务语义的规定: 任何对规则进行操作的事务必须执有相应锁. 除触发操作需要读锁外, 其他所有对规则的操作必须执有写锁.

### 1.3 规则的继承

ORDB 应该利用系统的继承机制对规则继承提供支持. 从关系模型的观点来看, 规则作为一种完整性约束存在, 因此其继承语义与约束继承一致; 而根据面向对象的观点, 则把规则看做是类(表)上的方法, 其继承语义就与方法相同. 这两种方式在实现上相差甚微.

### 1.4 多规则触发

当某一事件同时触发多条规则时, 由于对具有更新能力规则的执行是过程性的, 即系统的最终状态依赖于它们的执行顺序, 因而在实际应用中往往需要数据库系统采取一定的策略来保证其状态的可预见性. 有两类方法可以用来处理多规则触发问题:

(1) 保守法, 该方法通过限制在表上所定义规则的类型及数目来避免多规则触发现象. ORACLE 是该方式的代表<sup>[4]</sup>.

(2) 激进法, 允许多条规则同时被触发, 根据这些规则是否并发执行, 又可细分为:

(2.1) 并发执行法, 该方法通过嵌套事务模型来实现, 详见文献<sup>[5]</sup>.

(2.2) 顺序执行法, 使用该方法的系统将确定出多个规则的执行顺序. 对于顺序的确定可根据 3 种语义: ① 随机语义, 按系统对规则的随机选择次序执行, 用户无法控制规则执行顺序. 在规则行为不具备聚合性的情况下, 将导致数据库最终状态的不可预见性; ② 确定语义, 按用户事先预定的顺序执行; ③ 确定与随机相结合, 即高优先数规则先执行, 相同优先数规则随机执行.

## 1.5 规则的级联触发

有时候, 一个规则的动作会作为另一个规则的触发事件, 从而引起级联触发现象.

**定义 1.** 设活跃规则  $R$  的触发事件集为  $Events(R)$ , 动作集为  $Actions(R)$ . 如果存在另一个活跃规则  $R'$ , 使得  $Actions(R) \cap Events(R') \neq \emptyset$  成立, 那么  $R$  的动作执行势必导致  $R'$  的触发. 在这种情况下, 我们称规则  $R$  级联触发规则  $R'$ .

级联触发与多规则触发都是因相同事件造成多条规则的触发, 但级联触发涉及到的各条规则具备明确的因果关系, 因而执行顺序也确定. 级联触发的一个潜在危险是所构成的规则链中会产生环, 从而使系统处于无限或有限循环触发规则的状态之中. 无限循环将导致规则死锁, 对系统危害极大. 由于在语义上考察级联触发是否造成循环以及该循环是否有限是相当困难的, 尤其是对于间接循环的情况, 因此, 在实现时可以采用超时、超触发规则数等监测机制加以控制.

## 2 ANGEL 对规则系统的支持

ANGEL 主要采用触发器机制来提供对规则系统的支持. 从减轻系统处理的复杂度及利于用户灵活使用的角度出发, ANGEL 设计了一种简化的 E-C-A 模型 (simplified event-condition-action, 简称 S E-C-A), 该模型仍可采用前述七元组的表示形式, 只不过对以下成分进行了特化:

(1) 触发事件是能够激活规则的 SQL 操作, 它可以是 SELECT, INSERT, UPDATE 与 DELETE 操作, 也可以是这些操作的组合;

(2) 触发动作可以是 PRINT 操作 (用于实现“弱约束”: 可以在前台打印警告信息, 但并不中止当前操作)、事务控制操作 (用于实现“强约束”: 可使当前事务自动夭折或回滚到指定保存点)、对数据库的更改操作、查询操作这些原子操作及其组合形式;

(3) E-C 耦合模式为 immediate. 目前之所以没有提供延迟模式是因为: 将触发条件延迟到事务结束之前检查额外需要的系统开销, 如对触发条件及规则动作的保存、对触发条件所涉及数据对象在本事务更改操作前后的新旧镜像保存等. 尤为复杂的是, 由于工程应用领域对长事务的支持, 上述保存不止在内存中进行, 还应保存到外存设备上, 以便在系统突然失效时可以进行恢复. 但 ANGEL 将立即耦合模式进一步细分为触发事件发生前和发生后立即检查触发条件两种. 另外, separate 耦合模式在实际应用中意义不大, 因此 ANGEL 也没有对其提供支持.

(4) C-A 耦合模式分为立即式和分离非提交依赖式 (以下简称分离式) 两种.

迄今为止, E-C 与 C-A 的延迟耦合模式还没有在任何一个商用 DBMS 中得以实现<sup>[1]</sup>.

值得提出的是, 在触发条件中, ANGEL 把 NEW 与 OLD 这两个关键字扩展为一种路径量词. 这不但将触发约束条件的表示与检查和系统中的普通表达式处理统一起来, 而且增强了嵌套关系

中复杂触发约束的表示能力. 考虑这样的触发约束条件: 设  $b, c$  分别为表  $A$  的原子属性, 那就可以用“WHEN (OLD  $b$ ) > (NEW  $c$ )”来表达这样的条件: “ $b$  的旧值大于  $c$  的新值”. 进一步设  $B$  和  $C$  为  $A$  的子表属性,  $bb$  和  $cc$  分别为  $B$  和  $C$  的原子属性, 对于诸如“ $B$  中每个  $bb$  的新值必须大于  $C$  中某个  $cc$  的旧值”这样的复杂条件可表示为: “WHEN (EACH  $B$ ). (NEW  $bb$ ) > (ANY  $C$ ). (OLD  $cc$ )”. 这样, ANGEL 实现了在其他 DBMS 中难以实现的对对象集合实施约束的功能.

在对触发器的操作过程中, ANGEL 允许具备相应权限的用户执行触发器的创建、更新、删除、使能以及使不能操作, 但触发操作由系统自动实施.

在触发器继承方面, 由于系统中语义完整性约束就是用于保证数据库状态的一致性, 如果这种一致性不能得到维持, 则引起一致性破坏的事务被夭折. 与此类似, 触发器也监视数据库的某种约束条件, 只不过其违约反应动作由用户指定. 它们共同承担对规则系统的支持任务, 因此二者的继承机制就完全一样.

ANGEL 允许在一个(子)表上定义多个相同类型的触发器, 当同一事件引发多个触发器的触发时, 需要系统采取一定的调度策略. 通过对前面介绍的几种方式进行分析可知, 保守法要求保证在每个表上不能存在响应同一事件且类型相同的触发器, 这虽然实现简单, 但难以满足实际应用的需要. 并发执行法在触发器数量较大的情况下势必导致因并发执行的事务很多而降低系统执行效率, 同时还使事务管理变得复杂. ANGEL 选用了确定与随机相结合的处理方法, 并在其触发器定义语句中, 提供了对冲突执行触发器的优先级定义的手段<sup>[6]</sup>. 但为了语义的确定性, 设计时最好使读写集之间有依赖关系的触发器具有完全序, 即允许多条触发器具有相同的优先数, 但这些触发器之间不应存在读写依赖关系. 对于多条具有相同优先数且同时被触发的触发器之间是否具有行为聚合性的判定, 可采用后面将介绍的对规则聚合问题的判定方法.

此外, 在级联触发方面, 虽然文献[7~9]给出了对于规则行为终止性判定的充分不必要条件, 但在实际应用中, 特别是在系统中规则数量较大时, 很难使规则库完全满足这些条件的限制. 为此, 系统还是要提供一定的监控手段. ANGEL 采用控制级联触发的触发器数目的方式来避免系统陷入触发器调用的死循环之中.

有关 ANGEL 中 S E-C-A 模型的具体实现在文献[6]中有详尽的叙述.

### 3 S E-C-A 模型中的规则行为分析

规则行为的复杂性, 往往使其设计人员很难预计他们所设计的规则将给数据库状态带来什么样的影响, 缺乏对规则设计和分析的方法论支持已成为主动数据库的发展瓶颈<sup>[10]</sup>.

具体地说, 规则行为的复杂性主要体现在终止性和聚合性两个方面. 以下就 ANGEL 系统的 S E-C-A 模型, 给出对规则(触发器)上述行为的判定方法. 这些方法可以有效地指导 ANGEL 触发器以达到应用设计的正确性和合理性, 同时, 也具有一定的推广价值.

先给出下文中各符号的含义说明:  $RS$  表示系统中的活跃规则集(以下除非特别说明, 所谈到的规则均为活跃规则); 对于规则集  $RS$  中的任意规则  $R$ ,  $Events(R)$  表示  $R$  的触发事件集,  $Actions(R)$  表示  $R$  的动作集,  $Triggers(R) = \{R' \mid Actions(R) \cap Events(R') \neq \emptyset\}$  表示执行  $R$  的动作所能触发的规则集; 对于操作  $OP$ ,  $SB(OP)$  表示以  $OP$  为触发事件的语句前触发器集,  $RB(OP)$  表示以  $OP$  为触发事件的行前触发器集,  $RA(OP)$  表示以  $OP$  为触发事件的行后触发器集,  $SA(OP)$  表示以  $OP$  为触发事件的语句后触发器集.

### 3.1 终止性判定

在 ANGEL 的 S E-C-A 模型中, 规则触发与事务执行密切相关, 因此其行为终止性判定也可以针对事务.

**定义 2.** 对于一个给定的规则集  $RS$ , 如果在某个事务的执行过程中, 经历任意的数据库变化之后,  $RS$  中的规则触发操作都肯定能终止, 则称规则集  $RS$  在该事务中具备可终止性.

**定义 3.** 对于一个给定的规则集  $RS$  和某操作  $OP$  执行前的初始状态, 如果在  $OP$  执行之后,  $RS$  中的规则触发操作肯定能终止, 则称规则集  $RS$  对  $OP$  具备可终止性.

**定理 1.** 对于一个给定的规则集  $RS$  和某操作  $OP$  执行前的初始状态, 如果  $\forall R \in RS, Events(R) \cap \{OP\} = \emptyset$ , 那么规则集  $RS$  对  $OP$  具备可终止性.

**证明:** 由条件  $\forall R \in RS$  且  $Events(R) \cap \{OP\} = \emptyset$  可知, 操作  $OP$  不成为  $RS$  中任何规则的触发事件,  $RS$  对  $OP$  的可终止性是显然的.  $\square$

**定义 4.** 对于一个给定的规则集  $RS$  和某操作  $OP$  执行前的初始状态, 如果满足下述条件, 且在  $OP$  执行之后  $RS$  中的规则触发操作肯定能终止, 则称从  $RS'$  出发,  $RS$  对  $OP$  具备可终止性:

①  $\exists RS' \subseteq RS, RS' \neq \emptyset$ ; ②  $\forall R \in RS', Events(R) \cap \{OP\} \neq \emptyset$ .

**定理 2.** 规则集  $RS$  在事务  $T$  中具备可终止性的充分条件是:  $RS$  对于  $T$  中每个操作都是可终止的.

**证明:** 因为每次事务执行是由一系列有序操作所构成, 所以  $RS$  对于  $T$  中每个操作的可终止性保证了它在事务  $T$  中的可终止性.  $\square$

由于每个触发事件可触发  $SB, RB, RA, SA$  这 4 类规则, 而其触发过程也是有序的, 因此有以下定理.

**定理 3.** 规则集  $RS$  对于  $T$  中某操作  $OP$  具备可终止性的充分条件是: 分别从  $SB(OP), RB(OP), RA(OP), SA(OP)$  这 4 个规则子集出发,  $RS$  对于  $OP$  都是可终止的.

判定从  $RS'$  出发, 规则集  $RS$  对  $OP$  是否具备可终止性, 可采用触发依赖图  $G_R(OP)$ , 其生成如下:

- (1) 对于  $\forall R \in RS'$ , 在  $G_R(OP)$  中有顶点  $V_R$  与其一一对应;
- (2) 对于  $\forall V_R \in G_R(OP)$ , 如果  $R$  的 C-A 耦合方式为 immediate, 且  $\exists R' \in RS, V_R \in G_R(OP), R' \in Triggers(R)$ , 那么在  $G_R$  中增加顶点  $V_{R'}$  及由  $V_R$  到  $V_{R'}$  的有向边;
- (3) 除了上述(1)、(2)之外,  $G_R$  中没有其他的顶点和有向边存在.

**定理 4.** 如果从  $RS'$  出发, 规则集  $RS$  对操作  $OP$  的触发依赖图  $G_R(OP)$  中无环, 那么从  $RS'$  出发,  $RS$  对于  $OP$  具备可终止性.

**证明:** 设  $RS$  中有  $N$  条规则, 那么  $G_R(OP)$  中可能存在的最大路径长度不超过  $(N-1)$ . 容易证明, 从  $RS'$  出发的规则处理序列经过有限个规则触发后必然终止.  $\square$

在  $G_R(OP)$  中存在有环的情况下, 可进一步检验环中是否存在这样的规则  $R_i$ , 在多次执行之后, 其触发条件不可能得到满足, 或者  $R_i$  将产生不触发环中下一个规则  $R_{i+1}$  的操作. 若  $G_R(OP)$  中的每个环上都有这样的规则存在, 则可终止性仍能够得到保证. 另外, 利用文献[9]中的方法也可以判定在有环时的可终止性.

### 3.2 聚合性判定

**定义 5.** 对于一个给定的规则集  $RS$ , 如果在某个事务的执行过程中可能有多个规则同时被触

发,但所有规则触发操作终止之后,数据库状态与规则的触发次序无关,则称规则集  $RS$  在该事务中具备可聚合性或行为一致性。

**定义 6.** 对于一个给定的规则集  $RS$  和某操作  $OP$  执行前的初始状态,如果在  $OP$  执行之后,无论以何种有效次序对  $RS$  中的规则进行触发,都能终止于同一数据库状态,则称规则集  $RS$  对  $OP$  具备可聚合性。

**定理 5.** 规则集  $RS$  在事务  $T$  中具备可聚合性的充分条件是: $RS$  对于  $T$  中每个操作是可聚合的。

证明:设  $T$  由有序操作  $OP_1, \dots, OP_n$  组成. 根据条件可知,  $OP_1$  执行后,必然终止于状态  $S_1; \dots; OP_n$  执行后,必然终止于状态  $S_n$ , 则  $T$  提交后,必然终止于状态  $S_n$ .  $\square$

**定理 6.** 规则集  $RS$  对于  $T$  中某操作  $OP$  具备可聚合性的充分条件是: $SB(OP), RB(OP), RA(OP), SA(OP)$  这 4 个规则子集对于  $OP$  都是可聚合的。

证明:根据  $SB(OP), RB(OP), RA(OP), SA(OP)$  这 4 个规则子集的触发有序性以及对于  $OP$  的可聚合性,本定理的正确性是显然的.  $\square$

**定理 7.** 如果规则  $R_1$  与  $R_2$  是可交换的,那么互换二者的触发次序不会影响所得到的数据库状态.  $\square$

有关规则的可交换性定义及本定理的证明请参阅文献[9].

**定理 8.** 如果规则集  $RS'$  为  $SB(OP), RB(OP), RA(OP), SA(OP)$  中之一,并且下述条件之一成立,那么  $RS'$  对于  $OP$  是可聚合的:①  $RS' = \emptyset$ ; ②  $RS' \neq \emptyset$  但其中所有规则都是有序的(有确定的优先级关系); ③  $RS' \neq \emptyset$ , 对于  $\forall R_1 \in RS', \forall R_2 \in RS', R_1$  与  $R_2$  之间无确定的执行顺序,但它们是可交换的。

证明:情形①,②的正确性是显然的. 现在证明情形③.

设  $RS'' = \{R_1, R_2, \dots, R_k\}$  为  $RS'$  中的无序子集,由本定理的条件可知,  $RS''$  中的任意两个规则是可交换的. 又设  $OP$  触发  $RS''$  中的规则之后,从某状态  $S_0$  开始有两个任意的无序规则处理序列  $\delta$  和  $\eta$ :

$$\begin{aligned} \delta: S_0 &\xrightarrow{R_{11}} S_{11} \xrightarrow{R_{12}} \dots \xrightarrow{R_{1k}} S_{1k}, R_{1i} \in RS'' (1 \leq i \leq k), \\ \eta: S_0 &\xrightarrow{R_{21}} S_{21} \xrightarrow{R_{22}} \dots \xrightarrow{R_{2k}} S_{2k}, R_{2i} \in RS'' (1 \leq i \leq k). \end{aligned}$$

$\delta$  和  $\eta$  中都必有  $R_m (1 \leq m \leq k)$  的一次出现. 根据  $RS''$  中任意两个规则的可交换性可知,必能在经过有限次的顺序交换之后,把  $RS''$  中的  $R_m (1 \leq m \leq k)$  调整到序列的第  $m$  个处理步上,且交换后的最终状态不变. 从而得到:

$$\delta: S_0 \xrightarrow{R_1} S'_{11} \xrightarrow{R_2} \dots \xrightarrow{R_k} S'_{1k} = S_{1k}, \quad \eta: S_0 \xrightarrow{R_1} S'_{21} \xrightarrow{R_2} \dots \xrightarrow{R_k} S'_{2k} = S_{2k}.$$

易见  $S'_{1j} = S'_{2j} (1 \leq j \leq k)$ , 所以  $S_{1k} = S_{2k}$  成立。

由序列  $\delta$  和  $\eta$  的任意性可知,情形(3)的正确性成立. 定理得证.  $\square$

## 4 小 结

本文通过对 RDBMS 中触发器机制的扩展来提供对 ORDBMS 中规则系统的支持手段,所讨论的 S E-C-A 模型已在 ANGEL 系统中完全实现。

**References:**

- [1] Stonebraker, M. Object-Relational DBMSs—the Next Great Wave. San Francisco; Morgan Kaufmann Publishers, Inc., 1996.
- [2] Raschid, L., Cobo, J. Semantics for update rule programs and implementation in a relational database management system. *ACM Transactions on Database Systems*, 1996, 21(4):526~571.
- [3] ISO-ANSI. (Working Draft) Database language SQL (SQL3). Technical Report, Washington, DC; International Standard Organization-American National Standard Institute, 1994.
- [4] Koch, G., Loney, K. ORACLE8; The Complete Reference. Berkeley; McGraw-Hill, 1997.
- [5] McCarthy, D. R., Dayal, U. The architecture of an active database management system. In: Clifford, J., Lindsay, B., Maier, D., eds. *Proceedings of the 1989 ACM SIGMOD International Conference on the Management of Data*. New York: ACM Press, 1989. 215~224.
- [6] Li, Hong-yan, Li, Zhan-huai, Deng, Jiang. The design of trigger in ANGEL system. *Journal of Lanzhou University (Natural Sciences)*, 1999, 35(supplement):29~34 (in Chinese).
- [7] Aiken, A., Hellerstein, J. M., Windom, J. Static analysis technique for predicting the behavior of active database rules. *ACM Transactions on Database Systems*, 1995, 20(1):3~41.
- [8] Baralis, E., Widom, J. An algebraic approach to rule analysis in expert database systems. In: Jorge, Bocca, Matthias, Jarke, Carlo, Zaniolo, eds. *Proceedings of the 20th International Conference on Very Large Data Bases*. Santiago; Morgan Kaufmann Publishers, Inc., 1994. 475~486.
- [9] Jiang, Yue-ping, Wang, Wei, Shi Bai-le. Model and behavioral determinism theory for ECA rules. *Journal of Software*, 1997, 8(3):190~196 (in Chinese).
- [10] Simon, E., Kotz-Dittrich, A. Promises and realities of active database systems. In: Dayal, U., Gray, P. M. D. Shojiro, Nishio, eds. *Proceedings of the 21st International Conference on Very Large Data Bases*. Zürich; Morgan Kaufmann Publishers, Inc., 1995. 642~653.

**附中文参考文献:**

- [6] 李红燕,李战怀,邓江. ANGEL系统的触发器设计. *兰州大学学报(自然科学版)*, 1999, 35(增刊):29~34.
- [9] 姜跃平,汪卫,施伯乐. ECA规则的模型和行为特定理论. *软件学报*, 1997, 8(3):190~196.

**Research and Implementation of the Active Rules in ORDBMS\***

LI Hong-yan<sup>1,2</sup>, LI Zhan-huai<sup>3</sup>, TANG Shi-wei<sup>1,2</sup>

<sup>1</sup>(Center for Information Science, Beijing University, Beijing 100871, China);

<sup>2</sup>(National Laboratory on Machine Perception, Beijing University, Beijing 100871, China);

<sup>3</sup>(Department of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China)

E-mail: lihy@cis.pku.edu.cn

http://www.pku.edu.cn

**Abstract:** In this paper, the formalized description of active rules in ORDBMS (object-relational database management system) is presented. The semantics of rule components, rule inheritance, rule operations, multiple rule triggering and rule cascade triggering is fully discussed. And a simplified E-C-A (event-condition-action) model called S E-C-A model suited to ANGEL system is achieved. Four types of rules (update-update rules, update-query rules, query-update rules and query-query rules), C-A coupling mode and priority definition of conflicting triggers are supported in S E-C-A. The capabilities of express and actualize set constraints are gained by extending NEW and OLD to a kind of path quantifier in expression forms of trigger constraint. And the analytic methods of rule behaviors based on transaction are also presented.

**Key words:** ORDBMS (object-relational database management system); E-C-A model; rule; trigger

\* Received May 17, 2000; accepted February 27, 2001

Supported by the National Natural Science Foundation of China under Grant No. 50073055; the National Grand Fundamental Research 973 Program of China under Grant No. G1999032705