

XIV Conferencia Latinoamericana de Informática
17avas. Jornadas Argentinas de Informática
e Investigación Operativa
Buenos Aires, Setiembre 1988.

MECANISMOS DE CONTROLE PARA UM SGBDR ATIVO

Décio Fonseca (*)

SUMÁRIO

Este artigo apresenta um Sistema de Gerenciamento de Banco de Dados Relacional e Ativo, através da inclusão no SGBDR de mecanismos de controle. Estes mecanismos fazem parte de um sistema global de gatilhos. Os controles serão realizados em diferentes níveis, tais como: Interface, Autorização e Execução.

ABSTRACT

This work presents a Data Base Management System Active, with the control mechanisms. This mechanisms are included in the triggers global system. The controls in the different levels are: Interface, Authorisation and Execution.

(*) Doutor em Informática Universidade Paris VI-INRIA, Professor Visitante do Departamento de Informática da UPPE: áreas de interesse Bancos de Dados, Sistemas de Informação e Sistemas Especialistas. AV. Agamenon Magalhães, 1500 apt 201 Torreão - 52030 - Recife-PE

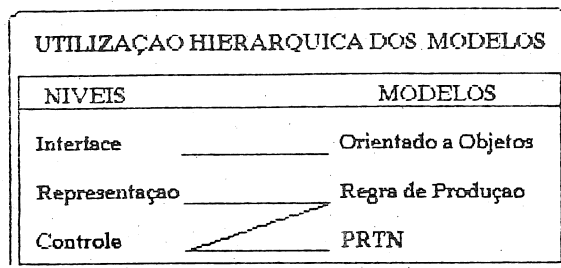
1) INTRODUÇÃO

O conceito de gatilhos é muito utilizado na modelagem dos aspectos dinâmicos dos sistemas de informação, ou seja, do seu funcionamento e de sua ativação automática. A dinâmica de um sistema de banco de dados é caracterizada por seus estados e pelas modificações válidas. É preciso que as transições dos estados do banco de dados, a partir de um conjunto de regras aplicadas nos objetos, atinjam sempre um estado coerente para exprimir corretamente o comportamento do sistema de informação. Várias foram as publicações que destacaram a importância dos gatilhos nos sistemas de banco de dados (BUNE79, ESWA76, FONS87, STON85....).

A evolução convergente das áreas de bancos de dados e de inteligência artificial autoriza o desenvolvimento de ferramentas tais como os gatilhos, permitindo exprimir e resolver certos problemas clássicos de bancos de dados. A motivação para retomar o estudo sobre os gatilhos tem sua origem na possibilidade de abordar este assunto sob um novo ângulo, na fronteira entre as áreas de banco de dados e de inteligência artificial.

Dois razões principais são favoráveis a um enfoque através de gatilhos. Eles simplificam consideravelmente a descrição das operações de atualização deixando ao SGBD o controle automático dos problemas de integridade, e simplificam também a utilização do SGBD, onde o usuário não precisa conhecer as regras de comportamento do sistema e suas dependências.

Neste artigo trataremos particularmente dos aspectos relativos aos controles para a modelagem global de um sistema de gatilhos. A proposição de um mecanismo para gatilhos se inspira em três modelos: orientado a objetos, os sistemas de produção e as Redes de Petri a predicado (PRTN). Estes três modelos são utilizados em três níveis.



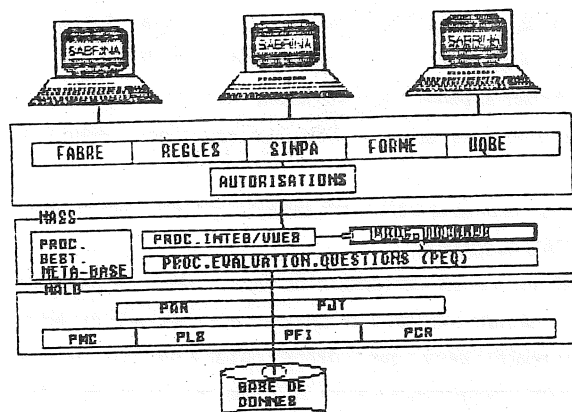
A utilização dos modelos em níveis permite uma melhor adequação dos modelos. A utilização combinada dos objetos com regras de produção oferece a possibilidade de estruturar bem os objetos do sistema, o que permite um controle mais estrito sobre as seqüências das operações. Três razões motivaram a escolha do modelo de PRTN para o controle da execução dos gatilhos: o modelo é bem adaptado à modelagem das regras de produção; êle modela bem as repercussões das atualizações e constitui uma ferramenta formal e gráfica potente, capaz de validar e de verificar a consistência das especificações.

Nosso enfoque visa alcançar os seguintes objetivos: - Oferecer ferramentas que permitam controlar de maneira eficaz a especificação e a execução de um sistema de gatilhos; - Controlar a definição precisa dos objetos e a seqüência da seleções das funções de cada objeto; - Controlar as repercussões das atualizações graças a uma modelagem orientada a objetos, a utilização das meta-regras, a interação amigável com os usuários e ao gráfico das transições(PRTN); - Controlar as mensagens enviadas pelos usuários ao sistema e vice-versa;

Na seção 2 mostramos a arquitetura geral do sistema. Na seção 3 apresentamos o sistema de controles, e na seção 4 concluímos com algumas observações sobre o trabalho e propomos novas orientações para futuros trabalhos.

2. ARQUITETURA GERAL

Esta seção define a arquitetura geral onde se insere o monitor de gatilhos. A arquitetura do SGBD SABRINA [GARD85] é composta de três máquinas virtuais: interface,asserccional e algébrica lógica. Este trabalho engloba as máquinas de interface (interface orientada a objetos) e asserccional (monitor de gatilhos).



O monitor de gatilhos é definido na camada de controle da máquina assercional, com uma alta interação com o monitor de restrições de integridade e com o monitor de autorizações. A complexidade do gerenciamento das restrições de integridade justifica a definição de um monitor independente de gatilhos. Esta solução apresenta como vantagens uma melhor integração dos elementos do sistema e o aumento do desempenho na execução dos gatilhos. O monitor vai se comportar como um motor de inferência de um sistema de regras de produção integrado ao SGBD, selecionando os gatilhos e ativando automaticamente suas ações. Estas ações serão controladas pelo sistema de integridade para garantir a coerência do banco. O funcionamento do monitor é baseado fundamentalmente nos encadeamentos para frente. A independência entre o monitor de gatilhos, o banco de dados e o conjunto de gatilhos torna o sistema evolutivo.

3. SISTEMA DE CONTROLES

O sistema de controle se encarrega do seqüenciamento das operações, e mais especificamente gerencia as informações de controle, testa as pre-condições de ativação e estabelece as post-condições para assegurar a coerência do sistema. Ele também envia mensagens ao mecanismo de ativação para provocar a execução das ações e controlar todas as interações com os usuários inclusive as autorizações de acesso ao sistema.

3.1 MODELO DE CONTROLE

O modelo de representação e controle (Rede de Pétri A Predicado) é constituído de:

(1) Um grafo de disparo (TAD, T, AC) onde TAD, T e AC são chamados de tipos abstratos de dados, transição e ação. Os vértices correspondem aos TAD. As transições definem as condições de disparo do gatilho e os arcos podem representar ou uma conexão entre as relações fonte e as transições, ou as ações do gatilho com uma conexão entre as transições e a relação destino.

(2) Relações fonte onde as operações que são nelas aplicadas como também os efeitos colaterais constituem o TAD do sistema.

(3) Vários TAD que contêm um conjunto de regras, onde cada uma destas regras (gatilhos) é constituída de uma parte condição e de uma parte ação. Na parte das ações podem existir várias ações sobre as relações destino que pertencem ou não a um TAD. Se as ações não agem sobre as relações destino que pertencem a um TAD, não teremos repercussões de atualização. Esta situação é um caso favorável de fim de ciclo. Cada TAD é representado uma só vez no grafo de transição.

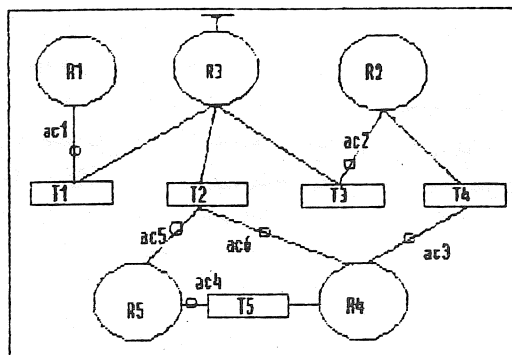
(4) Gatilhos disparáveis. Um gatilho é dito disparável se as seguintes condições são satisfeitas: - as ocorrências das nuplets das relações satisfazem as condições do gatilho; - as condições temporais são verificadas; - as condições agindo nas operações de atualização são satisfeitas.

(5) Gatilhos dos TAD e das transações atômicas. No modelo todas as ações são definidas como unidades atômicas de tratamento. Se um gatilho tem uma ou mais ações a disparar elas devem ser executadas até o fim de sua execução normal. Uma nova execução do gatilho não se realizará antes do final das execuções das ações do gatilho precedente. O efeito de cada atualização sobre o TAD provocará uma execução atômica dos gatilhos selecionados no TAD e em seguida é feita a análise dos efeitos colaterais dos gatilhos.

(6) Verificações e de tratamentos sobre relações temporária, ou seja, o sistema não trabalha diretamente sobre as relações permanentes. Entre as ações, as operações de atualização atuam sobre as relações temporária apenas dentro do espaço de trabalho alocado, enquanto outras como as mensagens ou a ação Solicitar são enviadas diretamente ao usuário. Em relação às condições, os critérios também são aplicados no espaço de trabalho e correspondem ao caso mais geral de nosso enfoque.

(7) Uma sequência de disparos, respeitando o caráter atômico de cada gatilho e de cada TAD. O encadeamento dos gatilhos se faz a partir das relações destino.

Exemplo geral do modelo: Seja o grafo de transições com as relações, as transições e as ações:



Suponhamos que uma atualização chega ao sistema e que modifica uma tupla da relação R3 que pertence a um TAD. A sequência do ciclo durante a execução é a seguinte:

- 1) T1((R3,ac1,R1),
- 2) T2(R3,ac5,R5), 3) T2(R3,ac6,R4),
- 4) T2(R3,ac2,R2), (5) T4(R2,ac3,R4),
- (6) T5(R4,ac4,R5).

Este exemplo permite de destacar os principais aspectos do ciclo de execução por etapas:

- 1) Na chegada de uma query ativa sobre uma relação pertencente a um TAD inicia o ciclo de execução.
- 2) Para prevenir os ciclos, a relação fonte do início de ciclo e o tipo de atualização são memorizados. Além disso, para cada gatilho que segue as repercussões das atualizações são guardados o tipo de atualização e a relação destino.
- 3) As transições são construídas por prioridade da esquerda para a direita.
- 4) A pesquisa das transições se faz da esquerda para a direita.
- 5) Para cada transição todas as ações são executadas, igualmente na horizontal da esquerda para a direita até a execução da última ação.
- 6) As relações destino são memorizadas na fila de espera durante a execução das ações, eliminando-se as eventuais relações duplicadas. Assim, após a execução de todos os gatilhos de cada tipo, o ciclo de execução continua tomando como TAD corrente o primeiro TAD seguinte da fila de espera. Este processo se desenvolve iterativamente até que não se tenha mais gatilhos disparáveis.

Este exemplo mostra também que uma relação destino (R2) pode tornar-se uma relação fonte e vice-versa. Ainda neste exemplo todas as relações pertencem a um TAD, exceto R5.

3.2 CONTROLES GERAIS E META-CONTROLES

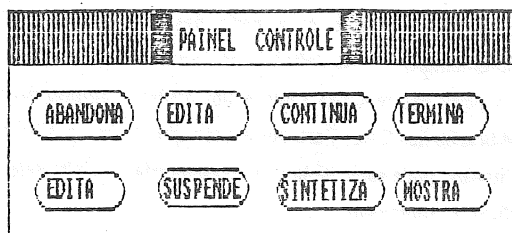
O ambiente de trabalho para propor o sistema global integrado é essencialmente constituído pelos seguintes elementos: os gatilhos, as restrições de integridade ativas e as meta-regras. A manutenção da coerência do banco é feita na inserção de cada um destes elementos. A seleção de um gatilho se faz por um procedimento de unificação baseado na comparação das condições. O controle preventivo de ciclos eventuais provocados pelos gatilhos é também controlado pelo sistema. Todas as vezes que esta situação se verifica o usuário é advertido. Se o usuário deseja a existência eventual de ciclos ele indicará através da interface de definição de gatilhos.

É fundamental num sistema de ativação automática garantir a correção das execuções, uma evolução coerente do banco de dados e a limitação dos efeitos das repercussões das atualizações sob pena de se colocar em perigo o sistema de informação. Para isto definimos várias estratégias de verificação: 1- uma interface orientada a objetos amigável, capaz de fazer um certo número de controles quando da definição e execução dos gatilhos; 2- definição de ferramentas de controle algorítmica; 3- utilização de modelos de representação e controle bem adaptado ao problema; 4- definição de um sistema para controlar as autorizações de acesso ao sistema de gatilhos. No entanto estas proposições estão ainda longe de serem suficientes; por isto propomos duas estratégias de simulação para melhor controlar o problema. A primeira corresponde ao controle algorítmico sintático e semântico das repercussões que podem provocar a inclusão no sistema de cada novo gatilho. A segunda é uma execução utilizando-se uma cópia do banco real e observando-se o comportamento do sistema, através de ferramentas gráficas de controle em cima do grafo de transição.

Para se dispor de uma estatística das execuções e das avaliações dos gatilhos, e contribuir para melhorar a administração da evolução do sistema de gatilhos definimos o objeto META_CONTROLE. Esta relação é atualizada em inserção quando o usuário define o gatilho em modificação após avaliação ou execução de cada gatilho, e em supressão após a supressão do gatilho correspondente.

Para se utilizar esta relação e para permitir um controle mais eficaz da evolução dos estados do banco de dados e do sistema de gatilhos foi definido um painel de controle. Este

painel usa interface orientada a objeto para simplificar as interações com o usuário e o administrador do banco de dados.



ABANDONA - abandona a transação ; SUSPENDE - suspende a execução do gatilho para analisar suas eventuais consequências ; EDITA - edita o grafo de transição para visualizar os comportamentos dos gatilhos e as propagações das atualizações provocadas por estes; CONTINUA - continua a execução interrompida pela função SUSPENDE; VISUALIZA- edita os gatilhos e para cada gatilho são associadas duas colunas. A primeira indica o estado do gatilho no ciclo da transação correspondente (ativo,) e a segunda indica o número de execuções do gatilho no ciclo ; SINTETIZA - permite consultar um conjunto de elementos do sistema antes de tomar certas decisões. A esta função são associadas outras funções de ajuda à investigação; MOSTRA - edita um certo número de informações sobre os gatilhos e sobre o estado da base; TERMINA - deixa o painel de controle.

Existem vários métodos para selecionar os gatilhos a executar: - Indeterminista - A sequência das transições candidatas ao disparo é escolhida arbitrariamente. A execução termina quando o conjunto de transições candidatas ao disparo engloba a totalidade das transições. - Interativa - a transição candidata ao disparo é função do diálogo interativo aleatoriamente escolhido ou explicitamente designado. - Prioridade - as transições candidatas são escolhidas seguindo sua prioridade. - Estatística - o tipo da regra(gatilho) determina as transições candidatas.- Temporal - pela ordem de satisfação das condições. O mecanismo combina estes métodos, o que garante uma grande eficácia no gerenciamento do processo de controle e de execução, e utiliza as meta-regras para controlar de maneira uniforme o sistema de gatilhos.

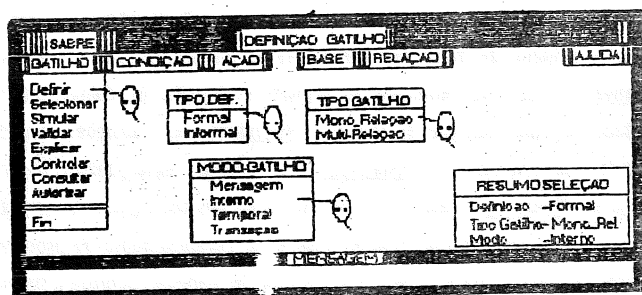
3.3 CONTROLES PELA INTERFACE

Na interface orientada a objetos a seleção dos objetos é efetuada em função da chegada de mensagens do usuário, portando sobre as classes dos objetos. Em cada objeto existe um

conjunto de ações permitindo responder as mensagens que são enviadas. A definição de um objeto gatilho pode ser feita de duas maneiras: a partir de um gatilho já existente incluindo outras propriedades como uma condição, ou a partir dos objetos CONDIÇÃO e AÇÃO e outras informações suplementares. A sua definição provoca a transformação das relações de base em relações ativas a partir das relações fonte e dos atributos referenciados na parte condição do gatilho, e a modificação dos TAD. Para que a execução de um gatilho seja protegida para o final da transação, o usuário pode explicitá-lo através da interface ou não selecionar o tipo de operação (inserção, modificação, supressão).

Todos os objetos à exceção das Relações, dos Atributos e dos Conhecimentos podem ser definidos e manipulados de maneira independente. Para evitar que o usuário defina CONDIÇÕES ou AÇÕES que eventualmente não serão jamais usadas, incluímos um controle a cada avaliação. Se após um certo tempo as condições e as ações não são jamais avaliadas, se estabelece um diálogo com o usuário para se decidir sobre a manutenção ou não destas ações.

Cada seleção de um objeto com o "Mouse" corresponde a uma mensagem para o sistema de interface. Contudo há certas mensagens que exigem um controle mais importante em função das repercussões que elas provocam, como por exemplo a escolha da função "Definir Gatilho". Para controlar os encadeamentos das seleções usamos as ferramentas disponíveis no sistema XWINDOWS e um mecanismo do tipo mapa de bits que controla as modificações de estado das funções do sistema.



Um objeto importante é chamado "CONHECIMENTO". Ele descreve o nível meta do sistema. Ele permite a definição das meta-regras e a descrição de todo tipo de conhecimento sobre a aplicação, sobre os gatilhos e sobre o sistema em geral. Ele permite além disso controlar a execução dos gatilhos através de um quadro de controle que é ativado a partir da função "meta-

controle". Este objeto corresponde a um menu fantasma e pode ser solicitado a todo momento a partir do "mouse" apoiado na janela principal.

3.4 CONTROLE DAS AUTORIZAÇÕES

O controle das autorizações para os gatilhos é muito importante. A especificação de um sistema confiável e ao mesmo tempo flexível é apresentado nas seguintes etapas:

ETAPA 1: ATRIBUIÇÃO DE AUTORIZAÇÃO

Na realidade a única autorização para se manipular as relações e os atributos deveria ser dada a aquele que as define. A autorização de se definir gatilhos poderá ser dada aos usuários que tenham pelo menos um bom conhecimento semântico da aplicação.

É preciso também definir o tipo de autorização de manipulação do usuário. O usuário pode ter uma autorização de modificar um atributo, mais eventualmente ele não poderá realizar operações do tipo agregado sobre este atributo. Neste caso incluímos uma autorização associada ao tipo de manipulação. Ainda nesta etapa é previsto um controle suplementar: quando a um usuário é atribuído uma autorização de modificação, o sistema verifica se todos os gatilhos podendo ser disparados pelas atualizações resultantes são possíveis de ser ativados pelo usuário. Se o usuário tem a autorização de modificar ou de suprimir uma relação do tipo principal, o sistema cria automaticamente a autorização para manipular as relações dependentes (dependência referencial).

Ainda na etapa 1 o administrador pode atribuir uma autorização de ativação, se o usuário tiver a autorização de modificação da relação fonte. Dispondo das autorizações necessárias o usuário pode passar a etapa 2. A autorização de se definir um gatilho evita que certos usuários que têm a autorização de manipular as relações fonte ou destino definam um gatilho sem terem o conhecimento necessário para esta definição e aumenta também a proteção de bases de gatilhos mal definidas.

ETAPA 2: DEFINIÇÃO DE GATILHO

Após o controle sobre as relações e os atributos fontes, existe o controle sobre os atributos e as relações destino. Resumindo, para se definir um gatilho é necessário se ter as

autorizações para modificar as relações fonte e destino, além da autorização para se definir o gatilho. Um vez definidos, os gatilhos são objetos do sistema onde são definidas as autorizações (atualização, consulta, ativação)

ETAPA 3 : AUTORIZAÇÃO DE ATIVAÇÃO E ATUALIZAÇÃO DOS GATILHOS

Nesta etapa é feita uma simulação do gatilho definido para garantir que no momento de sua execução o ciclo da transação não sera abandonado. É também atribuída a autorização de ativação para os gatilhos dependentes. Esta autorização é dada quando o usuário não tem a autorização de modificar a relação destino, mas tem a autorização de modificar a relação fonte. Esta autorização dá apenas o direito de executar o gatilho. No entanto o usuário não pode modificar diretamente o atributo da relação fonte correspondente. A execução dos gatilhos lhe é transparente.

ETAPA 4 : EXECUÇÃO DOS GATILHOS

Quando o gatilho vai ser executado ele pode provocar o abandono da transação. Se o usuário não tem as autorizações necessárias este abandono é natural. No entanto se ele tem uma autorização de execução concedida na etapa 2(dois), antes da execução do gatilho uma atualização temporaria das autorizações é efetuada para possibilitar as modificações da relações por outros gatilhos, evitando assim o abandono indevido de transações. Para se ter a autorização de execução é preciso se ter pelo menos a autorização de modificação sobre a relação fonte.

A autorização de ativação tem como vantagens:

- Uma transparencia de execução na medida em que certas ações podem ser executadas sem o conhecimento do usuário.
- Uma proteção de atributos sem rigidez no controle: a manipulação destes atributos sendo feita por gatilhos interpostos de maneira indireta.
- Uma redução nos controies das restrições , pois certos gatilhos podem compensar as incoerências.
- Uma maior garantia da coerência como consequência das duas vantagens precedentes.

5) CONCLUSÃO

Este trabalho apresenta uma solução eficaz para controlar a especificação e a execução de gatilhos, através de ferramentas historicamente ligadas a inteligência artificial como as regras de produção e de outras ferramentas (algorítmicas, interface orientada a objetos) . O sistema apresentado é ativo, possuindo capacidades reativas a partir dos gatilhos e da restrições de integridade ativas.

As orientações futuras deste trabalho são:

- explicitar as razões das incoerências;
- oferecer maiores possibilidades de dedução de informações e de detectar eventuais incoerências;
- definir ferramentas capazes de evitar incompatibilidades ou redundancias entre gatilhos;
- oferecer mais possibilidades gráficas que permitirão um controle mais simples e interativo com os usuários.

A modelagem de um sistema de gatilhos tende a evoluir em direção a soluções paralelas, beneficiando-se das máquinas com arquiteturas paralelas. Este novo ambiente paralelo modificará o conceito de gatilho temporal. Neste caso, teremos uma modelagem e uma avaliação de vários gatilhos ao mesmo tempo, sendo também incorporada a noção de estado .

BIBLIOGRAFIA

- [BUNE79] BUNEMAN O.P, CLEMONS E.K.: "Efficiently Monitoring Relational Databases". ACM TODS Vol4, N°3, Sept. 1979.
- [ESWA76] ESWARAN K. P.: Specifications, Implementations and Interactions of a Trigger Subsystem in an Integrated Database System". IBM Research RJ1820, Aug 1976.
- [FONS87] FONSECA D. "Un Mecanisme d'Activation et de Controle de Declencheurs Orienté Objets". Thèse de l'Université Paris VI, dec 1987.
- [GARD85] GARDARIN G. et al: SABRINA: Une Evolution du SGBD SABRE vers un Systeme Intelligent et Actif. Actes PRC BD3, St Pierre de Chartreuse, Mars 1985.

- [MELK83] MELKANOFF M., CHEN Q.: "Integrating Action Capabilities into Information Databases". 2nd Int. Conf. on Databases, Cambridge, Sept 1983.
- [STON85] STONEBRAKER M.: "Triggers and Inferences in Database Systems", College of Engineering University of California, Berkeley, 1985.
- [TUCH85] TUCHERMAN L., FURTADO A., CASANOVA M.: "A Tool for Modular Database Design". VLDB Conf. Stockholm 1985.