Triggers, Rules and Constraints in Databases

Juan M. Ale  
*Universidad de Buenos Aires, Argentina*

Mauricio Minuto Espil  
*Universidad Católica Argentina, Argentina*

**INTRODUCTION**

Databases are essentially large repositories of data. Since the mid-1980s up to the mid-1990s, considerable effort has been paid to incorporate reactive behavior to the data management facilities available. Reactive behavior is characterized by variants of the event–condition–action model. Applications areas include checking for integrity constraints, system alerts, materialized view maintenance (especially useful in data warehousing), replication of data for audit purposes, data sampling, workflow processing, implementation of business rules, scheduling, and many others. Practically all products offered today in the database marketplace support complex reactive behavior on the client side. Nevertheless, the reactive behavior supported by those products on the server side is poor. Recently, the topic has regained attention because of the inherent reactive nature demanded in Web applications and the necessity of migrating many of the functionalities of browsers to active Web servers (Bonifati, Braga, Campi, & Ceri, 2002).

**BACKGROUND**

Several applications that support reactive behavior in the electronic commerce arena appeared recently, as is the case in the following: the Active Views system, described in Abiteboul et al. (1999); the event–condition–action (ECA) rule language for XML repositories, described in Bailey, Poulouvassilis, and Wood (2002); and the set of classes for remote notification within a Web service environment, described in Bonifati, Ceri, and Paraboschi (2001).

Supporting reactive behavior implies that a database management system must be viewed from a production rule system perspective (Baralis, Ceri, & Paraboschi, 1996). An active database system must be viewed from a production rule system perspective (Baralis, Ceri, & Paraboschi, 1996). A central issue in the knowledge model of active databases is the concept of active rule. An active rule is defined throughout three dimensions: event, condition, and action. In this case, this is termed an ECA rule.

Active rules and integrity constraints are related topics (Ceri, Cochrane, & Widom, 2000). Database engines do not bring a full support of declarative integrity constraints in their kernels. When a complex constraint must be enforced on data, and the constraint cannot be declared, it must be emulated by means of triggers.

From a user’s point of view, reactivity is a concept related to object state evolution over time. Dynamic constraints, constraints making assertions on the evolution of object states, may be needed to control changes in the states of data objects (Sistla & Wolfson, 1995b).

**ACTIVITY WITHIN DATABASES**

Usually, a database system performs its actions in response to requests from users in a passive way. In some cases, it is desirable that actions be taken with no human intervention, that is, automatic response to certain events.

Traditionally, the latter behavior has been obtained by embedding it into user applications; that is, the application software recognizes some events triggered by an user and performs some actions in response.

Because of the complexity in supporting reactive behavior, it would be desirable that the active functionality be provided by the database system. A database with a capability of reacting to external or internal stimuli is called an active database. An active database system can be thought of as coupling a database management system with a rule-based programming environment (Paton & Diaz, 1999). Among the applications that use active database systems nowadays, we can mention inventory control systems, online reservation systems, and portfolio management systems, just to name a few.

**Knowledge Model**

A central issue in the knowledge model of active databases is the concept of active rule.

An active rule is defined throughout three dimensions: event, condition, and action. In this case, this is termed an ECA rule.
Related Content

The Impact of Training on Virtual Project Teams: A TIP Investigation
Peggy M. Beranek and M. Cathy Clairborne (2012). International Journal of Information Technology Project Management (pp. 36-48).
www.igi-global.com/article/impact-training-virtual-project-teams/62573?camid=4v1a

Quality Assurance Issues for Online Universities
www.igi-global.com/chapter/quality-assurance-issues-online-universities/14046?camid=4v1a

Implementing Software Metrics at a Telecommunications Company - A Case Study
www.igi-global.com/chapter/implementing-software-metrics-telecommunications-company/44602?camid=4v1a

Embracing Change: How South Ayrshire Council Library Service Became a World Leader in Electronic Resources Management
www.igi-global.com/chapter/embracing-change/90183?camid=4v1a