IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Hershey PA 17033-1117, USA ITB7510
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

Chapter 8

Using Business Rules Within a Design Process of Active Databases

Youssef Amghar, Madjid Meziane and André Flory
National Institute of Applied Sciences, France

Modeling behavior is an important task of the information system engineering process. This task is especially important when information systems are centered on active databases, which allow the replacement of parts of application programs with active rules. To relieve programmers from using either traditional or ad hoc techniques to design active databases, it is necessary to develop new techniques to model business rules. For that reason, inclusion of rules during analysis and design stages becomes an actual requirement. In this paper, we propose a uniform approach to modeling business rules (active rules, integrity constraints, etc.). To improve the behavior specification, we extend the state diagrams that are widely used for dynamic modeling. This extension is a transformation of state transitions according to rule semantics. In addition, we outline new functionalities of Computer Aided System Engineering (CASE) to take into consideration the active database specificities. In this way, the designer can be assisted to control, maintain, and reuse a set of rules.

Current design methods for information systems do not consider rules at the design level. In systemic methods such as "Structured Analysis Design Technique" (SADT) (Yourdon, 1979), rules are considered as a part of the design process but they are not modeled explicitly. In Object-Oriented methods such as the Object Modeling Technique (OMT) (Rumbaugh, 1991) or Object-Oriented Analysis (OOA) (Booch, 1994), rules are partially represented in dynamic models, particu-

Previously Published in the *Journal of Database Management, vol.11, no.3,* Copyright © 2000, Idea Group Publishing.
This chapter appears in the book, Data Warehousing and Web Engineering by Shirley Becker

larly in state diagrams. Moreover, at the development level, rules are often coded in the application programs implying a hard maintenance of business rules. These methods are generally supported by CASE.

To allow designers to exploit the specificities and features of active databases, it is important to build prototyping and monitoring tools to assist the designer during the design and development stage. This kind of tool offers indicators about choice relevancy and writing of rules. An active database management system (active DBMS) is an extension of a passive (relational or object) DBMS by adding trigger mechanisms. The notion of trigger appeared in the seventies, and has been generalized to the notion of active rule that is based on the Event-Condition-Action (ECA) formalism. The semantics of an ECA rule is as follows: when an event E is produced, if the condition C is satisfied, then the action A is executed. Actions are initiated by the DBMS when appropriate events occur, independently of external requests. These rules allow database designers to specify the active behavior of a database application that provides the enforcement of database integrity.

In the literature, several approaches were proposed to integrate active concepts into databases. For most systems, the knowledge model is based on ECA rules and the execution model on the nested transaction model, which authorizes different coupling modes (immediate, separate, deferred). Other systems use a weakened version of ECA rules. Furthermore, a number of research projects on active databases have focused on the rules' management and their evaluation. Several commercial DBMS include event/trigger mechanism proposed initially by Kotz (1988), such as the Postgres rule system (Stonebraker, 1990), Starburst's production and alert rules (Lohman, 1991), Ariel's production rule system (Hanson, 1989), the (ECA) model of HiPAC (Dayal, 88), and the event-action (EA) model of Ode (Gehani, 1992). In addition, there is a general agreement to consider that the new generation of DBMS systems would include active capabilities (Buchman, 1993) to support non-conventional applications such as documentation, geographic systems, workflow, and project management.

The design issue of active database applications is known as one of the most open research problems. Indeed, to design active database applications, programmers use either traditional or ad hoc techniques, which increases the complexity of applications by forcing the user to defer several modeling decisions concerning the active behavior to the development stage.

To gain benefits of active database capabilities, new approaches require inclusion of rules during both analysis and design stages. Few researchers have addressed the conceptual specification of behavioral aspects of applications independently from any active DBMS. To our knowledge, only IDEA (Ceri, 1993) and SORAC (Peckham, 1995) projects have treated the design of active database. However IDEA methodology is strongly linked to Chimera that is a DBMS