Chapter VIII

Modeling of Business Rules For Active Database Application Specification

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

Active database applications require the classic cycle of analysis, design, prototyping and implementation. During analysis and design steps of the information system engineering process, modeling behavior is an important task. This task is both essential and crucial when information system is centered on active databases, which allow the replacement of parts of application programs with active rules. For that reason, the specification of business rules during analysis and design steps becomes an actual requirement. Business rules ensure the well-functioning of information system. They are descriptive (integrity constraints) or functional (derivation rules and 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. These techniques have to enhance the specification of dynamic aspect through a high-level description language able to express precisely and completely rule semantic. In this chapter, we propose a uniform approach to model 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.

INTRODUCTION

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 stages. 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.

Current design methods for information systems do not consider explicitly 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, Blaha, Lorensen, Eddy, & Premerlani, 1991), Object-Oriented Analysis (OOA) (Booch, 1994) or methods built upon unified process using unified modeling language (UML), rules are partially represented in dynamic models, particularly 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 facilitate the specification of applications. However, these tools present some insufficiencies regarding the requirements of business rules that need to be specified at a high level of description as well as at a logical level for a more detailed specification.

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 of 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, Dittrich, and Mulle (1988), such as the Postgres rule system (Stonebraker, Jhingran, Goh, & Potamianos, 1990), Starburst's production and alert rules (Lohman, Lindsay, Prahesh, & Schiefer, 1991), Ariel's production rule system (Hanson, 1989), the (ECA) model of HiPAC (Dayal et al., 1988), and the event-action (EA) model of Ode (Gehani, Jagadish, & Shmueli, 1992). In addition, there is a general agreement to consider that the new generation of DBMS systems would include active capabilities (Buchmann, 1993; Campin & William, 1997; Diaz & Jaime, 1997) 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 increase 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 & Manthey, 1993; Ceri & Fraternali, 1997) and SORAC (Peckham, MacKeller, & Doherty, 1995) projects have treated the design of active database. However, IDEA methodology is strongly linked to Chimera, a DBMS developed especially in the framework of IDEA project. In IDEA project, any rules' specification is proposed. Rules, identified from requirement analysis, are directly expressed to the syntax of Chimera. The SORAC model permits the schema designer to specify enforcement rules that maintain constraints on object and relationships to facilitate the task of the designer. With active

Related Content

Database Cooperation: Classification and Middleware Tools

Paolo Atzeni, Luca Cabibboand Giansalvatore Mecca (2000). *Journal of Database Management (pp. 3-11).*

www.irma-international.org/article/database-cooperation-classification-middleware-tools/3248

A Template for Defining Enterprise Modelling Constructs

Andreas L. Opdahland Brian Henderson-Sellers (2004). *Journal of Database Management (pp. 39-73).*

www.irma-international.org/article/template-defining-enterprise-modelling-constructs/3310

A Multiple-Bits Watermark for Relational Data

Yingjiu Li, Huiping Guoand Shuhong Wang (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications (pp. 2223-2244).*

www.irma-international.org/chapter/multiple-bits-watermark-relational-data/8032

Location-Aware Query Resolution for Location-Based Mobile Commerce: Performance Evaluation and Optimization

James E. Wyse (2006). *Journal of Database Management (pp. 41-65).*

www.irma-international.org/article/location-aware-query-resolution-location/3357

A Scalable Algorithm for One-to-One, Onto, and Partial Schema Matching with Uninterpreted Column Names and Column Values

Boris Rabinovichand Mark Last (2014). *Journal of Database Management (pp. 1-16).*

www.irma-international.org/article/a-scalable-algorithm-for-one-to-one-onto-and-partial-schema-matching-with-uninterpreted-column-names-and-column-values/138623