The SAMOS Active DBMS Prototype

Stella Gatziu, Andreas Geppert, Klaus R. Dittrich
Institut für Informatik, Universität Zürich
{gatziu, geppert, dittrich}@ifi.unizh.ch

SAMOS is an active object-oriented DBMS implementing "reactive behavior" by detecting situations in the database and its environment and performing user-specified actions.

SAMOS extends an object-oriented DBMS with active features [1]. It provides a rule definition language as a means to specify ECA-rules including constructs for the definition of events, conditions, and actions [2]. Events can be primitive (time events, message sending events, etc.) or composite, where the latter one are constructed by operators like conjunction, sequence, negation applied to primitive or other composite events. The event history consists of all occurrences of the defined events. It begins at the point in time when the first event is defined and ends when all event definitions are deleted. Usually, the history lasts over many sessions and over several transactions. Thus, it is possible to signal a composite event based on events that have occurred in different application sessions or transactions. For event definitions, a time interval can be specified to determine when an event has to occur in order for it to be relevant for the appropriate rule(s). Event parameters (e.g., the identifier of the object that received a certain message) are supported and can be used to further restrict composite events to actually interesting ones. Both, conditions and actions must be programmed in the data manipulation language. Coupling modes like immediate, deferred or decoupled specify the point of time when condition evaluation and action execution begin.

The prototype implementation of SAMOS has a layered architecture. All active components are built "on top" of the passive object-oriented DBMS ObjectStore. The SAMOS prototype consists of three layers. Its topmost layer comprises tools, such as a compiler and a rule analyzer. The compiler checks the syntactic and semantic correctness of the defined rules, and updates the rulebase appropriately. The analyzer is responsible for checking further semantic properties of rulebases, such as termination. The intermediate layer consists of a number of components such as a rule manager for the retrieval of information about event and rule definitions, a detector for composite events [3], a new class for SAMOS' own transaction management on top of ObjectStore, a rule execution component for condition evaluation and action execution. The third layer is ObjectStore itself.

The major parts of SAMOS (including rule definition, persistent storage of rule and event objects, event detection, and rule execution) are operational. We are currently investigating performance of active object-oriented DBMSs and are comparing SAMOS with other active object-oriented DBMSs from a performance point of view [4]. We are also looking into some potential applications of active database systems. These application domains include DBMS-internal tasks such as consistency maintenance as well as applications on top of DBMSs, such as banking, workflow management, and software engineering environments.

The demonstration shows how reactive applications can be implemented, maintained, and executed in SAMOS.

References

[1] S. Gatziu, A. Geppert, K.R. Dittrich. *Integrating Active Concepts into an Object-Oriented Database System.* Proc. 3th Intl. Workshop on Database Programming Languages, August 91.

[2] S. Gatziu, K.R. Dittrich *Events in an Active Object-Oriented Database System.* Proc. 1st Intl. Workshop on Rules in Database Systems, September 93.

[3] S. Gatziu, K.R. Dittrich. *Detecting Composite Events in an Active Database Systems Using Petri Nets.* Proc. 4th Intl. Workshop on Research Issues in Data Engineering: Active Database Systems, Houston, February 94.

[4] A. Geppert, S. Gatziu, K.R. Dittrich. *007 Meets the Beast: Performance Evaluation of an Active Object-Oriented Database Management System.* Technical Report, Institut fuer Informatik, Universitaet Zuerich, 94.

We thank Hans Fritschi, Anca Vaduva, Thomas Meyer and Dimitrios Tombros for their help in implementing the SAMOS prototype and this demonstration.

SIGMOD ' 95,San Jose , CA USA