

The Algres Testbed of CHIMERA: An Active Object-Oriented Database System

Stefano Ceri Piero Fraternali Stefano Paraboschi Giuseppe Psaila

Politecnico di Milano — Dipartimento di Elettronica e Informazione

{ceri,fraterna,parabosc,psaila}@elet.polimi.it

Chimera is a novel database language integrating an object-oriented data model, a declarative query language based on deductive rules, and an active rule language for reactive processing. In most active object-oriented databases, active rules are associated with objects through methods—rules are triggered by method activations, and are used as devices for testing pre- and post-conditions for method applications to individual object instances. The *Chimera* approach is substantially different: it uses *set-oriented* active rules, activated as the effect of several, logically indistinguishable events affecting multiple object instances. This approach is consistent with the rest of *Chimera*, which supports a set-oriented, declarative query and update language.

Object-orientation in *Chimera* guarantees important advantages over relational active rules, due to the use of *object identifiers*. In object-oriented databases, object identifiers are uniquely associated with each object instance. In rules of *Chimera*, object-valued variables (i.e., variables ranging over object identifiers) provide a powerful *binding passing* mechanism for linking events to conditions and conditions to actions, thus bridging the three syntactic components of active rules.

Active rules in *Chimera* have several innovative features:

- They support different modes for processing events (called *event consumption modes*) when rules are activated.
- They optionally support event composition (called *net effect computation*) when the same object is the target of multiple operations.
- They support mechanisms for accessing *intermediate states* of affected object instances during transaction execution.

These features are orthogonal to each other and can be generalized to any active rule system. In *Chimera*, they yield a powerful language that supports a variety of applications, including integrity checking for both static and dynamic constraints, derived data and view materialization, monitoring, and bookkeeping.

Two testbeds of *Chimera* were built within the IDEA project, by ECRC (in the Eclipse environment) and by

Politecnico (in the Algres environment). The Algres testbed has yielded several interesting observations:

- 1 The set-oriented nature of active rules in *Chimera* has proved to be useful for the compilation and optimization of rules. For example, optimization techniques for the condition part of rules are identical to those used for the *select* query primitive of the language.
- 2 Early evaluation of event predicates (*occurred* and *holds*) has proved to be useful for immediately suspending the evaluation of a rule condition if the relevant predicates are empty. This built-in optimization reinforces the generally held principle that rules should be *incremental*, i.e., they should use state changes (deltas) in their condition whenever possible.
- 3 In deciding how to compute event instances, a trade-off analysis was performed between incorporating event management strategies in the translation of triggers or doing some of it at run-time. As a result of this analysis, we have chosen to compile most of event management algorithms, including the computation of events propagating due to generalization hierarchies.
- 4 To implement active rules easily and efficiently there is a need for mechanisms in the underlying DBMS that automatically and transparently select objects affected by operations and store their identifiers in appropriate collections.
- 5 Preliminary measures of performance are encouraging: active rule selection is very fast and thus the overall execution times are dominated by execution time of the Algres code produced for transactions and for rules.

The Algres testbed will be extended in two directions:

- The user interface will gradually evolve into a complete design tool for active rules. Such a tool can be used for understanding rule behavior at a conceptual level, then mapping the behavior into a set of *Chimera* rules, or rules that can be supported by commercial relational products.
- The system will incorporate design techniques and tools for mapping views and constraints defined in *Chimera* into suitable rules, thereby implementing missing parts of the full *Chimera* language.

For further information about the IDEA project and the Algres testbed, please refer to the WWW server <http://www.ecrc.de/idea/idea.html>. *Chimera* is described in J. Widom, S. Ceri, U. Dayal (eds), "Active Database Systems", Morgan-Kaufmann, Spring 1995.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association of Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

SIGMOD '95, San Jose, CA USA

© 1995 ACM 0-89791-731-6/95/0005..\$3.50