

# Using Smodels (Declarative Logic Programming) to Verify Correctness of Certain Active Rules

Mutsumi Nakamura, Ramez Elmasri  
Department of Computer Science and Engineering  
The University of Texas at Arlington, Arlington, TX 76019  
nakamura@asu.edu, elmasri@cse.uta.edu

## Abstract

*In this paper we show that the language of declarative logic programming (DLP) with answer sets and its extensions can be used to specify database evolution due to updates and active rules, and to verify correctness of active rules with respect to a specification described using temporal logic and aggregate operators. We classify the specification of active rules into four kind of constraints which can be expressed using a particular extension of DLP called Smodels. Smodels allows us to specify the evolution, to specify the constraints, and to enumerate all possible initial database states and initial updates. Together, these can be used to analyze all possible evolution paths of an active database system to verify if they satisfy a set of given constraints.*

Active databases have tremendous potential to simplify database programming by specifying appropriate ECA(Event-Condition-Action) rules. One impediment to their widespread use has been the need for design methodologies and verification techniques to show that a set of rules is consistent and satisfies the system requirements.

Database triggers are typically activated when a particular update occurs. When we have a set of triggers, an update which is activated by one trigger can activate other triggers and can create a sequence of updates. A designer of an active database should make sure that sequences of updates do not violate the "purpose" of the triggers. Triggers are an essential component of active database systems and are now part of the SQL3 database standard. After their implementation in several research prototypes, they are now incorporated in commercial database systems such as Oracle, Sybase and IBM's DB2-V2 [1].

There has been much work on the syntax of triggers, how they are to be executed, their execution models, and their implementations, and some work on formal frameworks for active databases in regards to semantics and expressiveness. However, comparatively fewer work has been done on methodologies to design triggers, specify what triggers

are supposed to achieve (their "purpose"), and ways to verify their correctness. We have made some progress on these issues in [2] where we developed a specification language for triggers which can be used to specify the purpose of triggers.

In this paper, we show how one can simulate the effect of active rules (triggers), specify their purpose, and a limited verification of the correctness of a set of triggers with respect to its purpose. For this we use declarative logic programming (DLP) with answer set semantics and the Smodels system [3].

Using Smodels, we can also enumerate all possible initial database states and initial updates, and specify active rules. In this way, the database evolution can be specified and tested against a given specification. If it satisfies all constraints of the specification we can conclude that the set of active rules is correct with respect to the specification.

Our work in this paper extends our earlier work in [2] where we identified state and trajectory invariance and maintenance constraints as important parameters in specifying the purpose of triggers and defined correctness of triggers. Here we show DLP and Smodels to be a candidate language for both specification and perhaps even to serve as the engine for a trigger verification system. Developing a general trigger verifications system with a DLP engine is one of our future goals. The full version of this paper is available at: <http://www.public.asu.edu/~mutsumi/active-db/papers/icde02.pdf> or [icde02.ps](#).

## References

- [1] D. Chamberlin. *Using the new DB2: IBM's Object-relational database system*. Morgan Kaufmann, 1996.
- [2] M. Nakamura and C. Baral. Invariance, maintenance and other declarative objectives of triggers – a formal characterization of active databases. In *CL 2000*, pages 1210–1224, 2000.
- [3] I. Niemela and P. Simons. Smodels – an implementation of the stable model and well-founded semantics for normal logic programs. In *LPNMR '97*, pages 420–429, 1997.