Transaction Datalog: a Compositional Language for Transaction Programming

Anthony J. Bonner

University of Toronto, Department of Computer Science
Toronto, ON, Canada M5S 1A4
bonner@db.toronto.edu

Abstract

In the classical model of database transactions, large transactions cannot be built out of smaller ones. Instead, transactions are modelled as atomic and isolated units of work. This model has been widely successful in traditional database applications, in which transactions perform only a few simple operations on small amounts of simply-structured data. Unfortunately, this model is inappropriate for more complex applications in which transactions must be combined and coordinated to achieve a larger goal. Examples include CAD, office automation, collaborative work, manufacturing control, and workflow management. These applications require new transaction models, new methods of transaction management, and new transaction languages. This paper focuses on the latter issue: languages for specifying non-classical transactions, and combining them into complex processes. In particular, we develop *Transaction Datalog*, a deductive language that integrates queries, updates, and transaction composition in a simple logical framework. This integration extends the deductive-database paradigm with several new capabilities. For instance, Transaction Datalog supports all the properties of classical transactions, such as persistence, atomicity, isolation, abort and rollback. It also supports properties found in many new transaction models, such as subtransaction hierarchies, concurrency within individual transactions, cooperation between concurrent activities, a separation of atomicity and isolation, and fine-grained control over abort and rollback. These capabilities are all provided within a purely logical framework, including a natural model theory and a sound-and-complete proof theory. This paper outlines the problems of developing a compositional transaction language, illustrates our solution (Transaction Datalog) through a series of examples, and develops its formal semantics in terms of a logical inference system.

A short version of this paper appears in *Proceedings of the Sixth International Workshop on Database Programming Languages (DBPL)*, Estes Park, Colorado, August 18–20 1997, pages 303–322.

This and related papers are available at the Transaction Logic web-page:
http://www.cs.toronto.edu/~bonner/transaction-logic.html