Workflow, Transactions and Datalog

Anthony J. Bonner

University of Toronto
Department of Computer Science
Toronto, Ontario, Canada M5S 1A4

`www.cs.toronto.edu/~bonner`
`bonner@db.toronto.edu`

Abstract

Transaction Datalog (abbreviated $\mathcal{TD}$) is a concurrent programming language that provides process modeling, database access, and advanced transactions. This paper illustrates the use of $\mathcal{TD}$ for specifying and simulating workflows, with examples based on the needs of a high-throughput genome laboratory. In addition to traditional database support, these needs include synchronization of work, cooperation between concurrent workflows, and non-serializable access to shared resources. After illustrating workflows, we use $\mathcal{TD}$ to explore the computational properties of workflows in data-intensive applications. We show, for instance, that workflows can be vastly more complex than traditional database transactions, largely because concurrent processes can interact and communicate via the database (*i.e.*, one process can read what another process writes). We then investigate the sources of this complexity, focusing on features for data modeling and process modeling. We show that by carefully controlling these features, the complexity of workflows can be reduced substantially. Finally, we develop a sub-language called *fully bounded* $\mathcal{TD}$ that provides a practical blend of modeling features while minimizing complexity.

A short version of this paper appears in *Proceedings of the Eighteenth ACM Symposium on Principles of Database Systems (PODS)*, May 31 - June 2, 1999, Philadelphia, PA, pages 294–305.