Overview

ROL is a deductive object-oriented database system that has been developed at the University of Regina [3]. It provides a uniform language [1, 2] for defining, querying and manipulating a database. The ROL language integrates important features of deductive databases and object-oriented databases with well-defined declarative semantics. It supports object identity, complex objects, classes, class hierarchies, multiple inheritance with overriding and blocking, and schema definitions. Unlike other deductive object-oriented proposals, ROL also supports structured values such as functor objects and sets, treating them as first class citizens, and providing powerful mechanisms for representing both partial and complete information about sets. As a result, it directly supports non-first normal form relations and is an extension of the pure value-oriented deductive database language such as Datalog (with negation) and LDL (without grouping) and subsumes them as special cases.

The ROL system is available through anonymous ftp from ftp.cs.uregina.ca/pub/rol. It is distributed with extensive examples. Since its first release in January of 1996, it has been used by hundreds of users worldwide for various data and knowledge-based applications. It has also been used to teach deductive databases, object-oriented databases and deductive object-oriented databases at a number of universities around the world. So far, four versions of ROL have been released. The latest version is 2.2 which is the one demonstrated.

A ROL database consists of three parts: a schema, a set of facts and a set of rules. The schema is a set of class definitions. The facts and rules are the extensional and intensional information about objects, their primary classes and/or their attribute values respectively.

The ROL system is organized into three layers. The first layer is the user interface. Two kinds of user interfaces are provided: textual user interface and graphic user interface. They provide different kinds of environment for the user to query and update the object database.

The second layer consists of the query manager and the update manager. They process query and update requests from the user interface. In ROL, an update request may imply a query, so that the update manager may send the query to query manager before performing the update.

The third layer contains the memory manager and the object manager. The memory manager manages the memory space that is used to store meta-information about facts, rules and class definitions, part of facts, rules, and class definitions that need to be used to process queries and updates, and intensional information derived with rules. The object manager provides rapid access to facts, rules, class definitions and meta information about them on the disk.

The ROL system has the following distinct features:

1. schema, facts, and rules are persistent
2. class definitions, facts, and rules needed for query evaluation are dynamically selected into memory
3. rules used for query evaluation are dynamically rewritten based on the query
4. intensional data is partially materialized in memory as opposed to being evaluated dynamically every time it is queried
5. least recently used (LRU) mechanism is used to remove intensional and extensional data in memory when memory space is needed
6. various evaluation strategies are used automatically based on the nature of the query and data in the database without user’s intervention so that there is no need for specifying query forms or modes.
7. set-at-a-time evaluation is used all the time
8. higher-order rules and queries are supported so that it is more convenient for the user to use the ROL system.

References


0-8186-8289-2/98 $10.00 © 1998 IEEE