Design and Implementation of Advanced Knowledge Processing in the KBMS KRISYS

Demonstration Description

J. Thomas
University of Kaiserslautern, Dept. of Computer Science
67653 Kaiserslautern, Germany
thomas@informatik.uni-kl.de

S. Deßloch, N. Mattos
IBM Database Technology Institute, Santa Teresa Lab.
San Jose, CA, 95150 USA
dessloch@almaden.ibm.com, mattos@stlvm14.vnet.ibm.com

KRISYS is a prototypical knowledge base management system (KBMS) conceived for client/server environments. The server DBMS is responsible for general data management tasks which are carried out in the data model of the server. Processing in the client buffer, termed knowledge processing, is performed in the scope of the object-oriented knowledge model of KRISYS. As user interface, KRISYS supports the set-oriented and declarative query language KOALA. In contrast to other OODBMS, KRISYS features a sophisticated client infrastructure for main-memory query processing in combination with a query server component. Our demonstration presents the salient features of the KRISYS knowledge model and of knowledge processing from an end user's point of view as well as from an implementational perspective.

We firstly discuss the KRISYS knowledge model, its features, and the query language KOALA. An object contains a set of attributes for representing the object's properties or for expressing object behavior. Attributes can be further described by aspects, defining, e.g., the cardinality of a slot. Objects are typically organized in hierarchies or lattices defined via abstraction concepts. KRISYS supports the abstraction concepts of classification, generalization, association, and aggregation whose semantics (e.g., multiple inheritance) is automatically maintained by the system. KOALA features two powerful operations, ASK and TELL, for querying and modifying knowledge bases according to the semantics of the knowledge model.

Subsequently, we turn to the implementational aspects of knowledge processing. After an overview of the architecture of KRISYS, we have a closer look at the *Working Memory*, the application buffer in the client where knowledge processing takes place. For object representation, main-memory pointer structures are employed that directly reflect objects and their relationships as defined by the knowledge model. Moreover, *Access Structures* provide a facility for indexing arbitrary collections of objects, e.g., (intermediate) results during knowledge processing. A set of basic operations for accessing and manipulating the buffer contents complements the functionality of the Working Memory.

The *KOALA Processing System* and the steps of knowledge processing are based on an algebraic processing model comparable to relational DBS: A query is transformed into an algebra graph and optimized; then, a plan-operator graph is constructed; finally, executable code is generated, and the query is actually evaluated. With the help of example queries, we give an insight into the implementation of each step and present the outcome of each transformation, i.e., algebra graphs, plan-operator graphs, and graphs made up of units of execution.

Our plan-operator approach involves the following concepts: *Plan-Operator Templates* realize a simple processing paradigm for plan operators, as well as extensibility at the plan-operator level. *Base Predicates* introduce knowledge-model semantics into query processing and guarantee extensibility of the query language without affecting existing plan operators. Units of execution, called *Blocks*, comprise subgraphs of a plan-operator graph, thus ensuring efficient dynamic code assembly, i.e., the construction of flexible units of execution at run time. Blocks employ *Logical Access Structures* for organizing internal data flow which ensure efficient data flow between plan operators.

These characteristics were achieved by a modular design and realization of the plan level, and, although implemented in LISP, we resorted only to implementational techniques that can be found in C-like programming languages as well.

SIGMOD ' 95,San Jose , CA USA
© 1995 ACM 0-89791-731-6/95/0005..$3.50

Refere3nces

Deßloch, S., Mattos, N., Mitschang, B., Thomas, J.: Design and Implementation of Advanced Knowledge Processing in the KBMS KRISYS, ZRI-Report 5/94, University of Kaiserslautern.

Mattos, N.: An Approach to Knowledge Base Management Systems, LNCS 513, Springer-Verlag, 1991.

Thomas, J., Deßloch, S.: A Plan-Operator Concept for Client-Based Knowledge Processing, Proc. 19th VLDB, Dublin, Ireland, August 1993.

Thomas, J., Gerbes, T., Härder, T., Mitschang, B.: Implementing Dynamic Code Assembly for Client-Based Query Processing, Proc. 4th Int. Conf. on Database Systems for Advanced Applications (DASFAA95), Singapore, April 1995.