OSAM*.KBMS: An Object-oriented Knowledge Base Management System for Supporting Advanced Applications

Stanley Y. W. Su, Herman X. Lam, Srinivasa Eddula,
Javier Arroyo, Neeta Prasad, Ronghao Zhuang

Database Systems Research and Development Center
University of Florida, Gainesville, FL 32611
E-mail: su@pacer.cis.ufl.edu

1 Introduction

Many advanced database application domains such as CAD/CAM, CIM, CASE, office automation, decision support, command/control/communication, etc., require the use of databases and the support of database management systems. The data characteristics and processing requirements of these application domains are quite different from those of business databases. Clearly, existing database management systems, which were mainly developed for business applications, do not provide the users of these advanced applications with adequate modeling, processing, and analysis tools and capabilities to meet their database needs. This short paper and its accompanying demo introduce an object-oriented knowledge base management system developed at the Database Systems Research and Development Center of the University of Florida. The main features to be presented and demonstrated include an object-oriented (OO) semantic model OSAM*, an OO query language OQL, a knowledge programming language K, and a set of graphical tools which serve as the user interface. The system runs on a network of IBM and SUN workstations.

2 Object-oriented Semantic Association Model (OSAM*)

OSAM* [SU89] is an object-oriented extension of SAM* reported in [SU83, SU85]. The model has been used as the basis for the development of several versions of OSAM*.KBMS [SU88, LAM89, LAM90, SU92] for testing and verifying our research results on an object-oriented query language, knowledge rule processing, graphical user-interface tools, and a knowledge base programming language. It allows the structural properties of any application object to be defined in terms of its various types of associations with other objects. The behavioral properties of the object are defined in terms of user-defined operations and knowledge rules which are used to declaratively specify integrity, security or other semantic constraints. The association types and rule specification capability of the model set it apart from the existing works on object-oriented models and systems. In this model, the information contents of a knowledge base are represented by different types of associations among object instances which are subject to the operations and constraints rules defined in their corresponding object classes. The OSAM*.KBMS makes use of the semantics of association types, operations, and constraint rules to automatically manage and maintain the consistency of a knowledge base. Multiple inheritance of attributes, operations and knowledge rules are supported.

SIGMOD /5/93/Washington, DC,USA
© 1993 ACM 0-89791-592-5/93/0005/0540...$1.50

3 Object-oriented Query Language (OQL)

To ease users' tasks in accessing and manipulating a knowledge base, we have designed and implemented a high-level, non-procedural, object-oriented query language called Object-oriented Query Language or OQL [ALA89]. The language is an "association- or pattern-based" query language instead of a more traditional "attribute-based" language. It allows data search conditions of different degrees of complexity to be specified uniformly and simply as patterns of object class associations instead of comparing values of keys and foreign keys. The language greatly simplifies the tasks of information users in their querying of the complex knowledge bases; particularly those complex queries that involve logical AND/OR branching and loop structures among multiple classes. It is also very suitable for graphical implementation. Query processing and query optimization in OSAM*.KBMS are based on the mathematical properties of an Association Algebra reported in [GUO91,SU93].

4 The Knowledge Programming Language K

This language and its implemented version K.1 [SHY91, ARR92] has the data definition constructs of OSAM* model, the set-oriented data manipulation constructs of OQL, trigger and rule specification constructs for declaratively specifying semantic constraints, and general-purpose programming language constructs for implementing arbitrary algorithms. It is used in OSAM*.KBMS as the textural language for defining object classes as well as for implementing methods. In this language, the association pattern specification of OQL is used to establish a subdatabase involving a single or multiple classes over which traditional programming

language can be used to process objects which satisfy the pattern specification. The association pattern specification is also used in knowledge rules so that a complex pattern of object associations can be stated as the condition for causing some operation(s) to be activated or some other complex pattern of object associations to be verified. The activation of a rule is subject to various trigger conditions and time constraints. The processing of queries and knowledge rules is supported by persistent knowledge bases. In the present implementation, K programs are translated in C++ code. During execution, the translated programs make calls to the KBMS to retrieve, store or update the persistent data, to make use of the trigger and rule processing facility, and to make use of other object management functions. The object manager is built on top of the commercial system ONTOS which provides all the needed storage management functions.

5 Graphical Tools

A set of graphical querying and development tools (GTOOLS) has been developed to ease the tasks of the users in defining, browsing, modifying and querying complex knowledge bases [LAM90, LAM92]. The toolset also provides facilities to assist users in database design and schema modification. The version of the toolset to be demonstrated is called XGTOOLS which uses X/Motif and runs on RS 6000 workstations. It consists of a graphical Browser, schema design and editing tools, and querying tools.

The graphical Browser provides a set of useful features to traverse a complex schema and "prune" it into a desired level of details and abstraction for the user's viewing and examination as well as in preparation for graphical querying. The schema design and editing tools allow the user to interactively create and update a schema, enter rules and methods associated with object classes. It supports two layers of functions: the lower layer functions are used to define, modify and delete the constructs of a schema, and the upper layer functions are devoted to perform operations on schemas, such as create, copy and delete schemas. The querying tools are used for graphically querying a knowledge base. There are three query modes in XGTOOLS: textual Mode, graphical Mode and Active Mode. The textual mode is intended for expert OQL users who can directly type in OQL queries in textual forms. In the graphical mode, the user selects the classes of interest and the relationship among them. The tool guides the user to generate complex patterns with multi-level nested branches (with AND and OR operators) and loops. After the CONTEXT clause has been generated, the user is asked to select the operation from a list of valid system and user-defined operations. The Active OQL mode is intended for users with the least amount of knowledge with OQL. It allows the user to select the classes of interest from a displayed schema. The system will automatically generate all possible query patterns based on the semantics of the selected classes. These query patterns are expressed in English. The selection of any expression will cause the system to graphically display its corresponding object diagram. Based on the English expression and the graphical display, the user can select the query pattern he/she wants. The system will construct the CONTEXT clause for the user. Finally, the system will guide the user to select the operation(s) to be performed on the identified objects. The final OQL query is constructed and displayed.

6 Summary

In this paper, we have provided an overview of the object-oriented semantic association model OSAM*, its query language OQL, the language K, and the graphical tools. They form an OO environment for application system development. The present system is a single user system without recovery and concurrency control features. Interested readers should refer to our publications on the various aspects of the system.

7 Selected Publications

[ALA89] Alashqur, A.M., Su, S.Y.W., and Lam, H., "OQL: A Query Language for Manipulating Object-oriented Databases", Proc. of 15th. Int'l Conf. on VLDB, Amsterdam, The Netherlands, Aug. 22-25, 1989.

[GUO91] Guo, M.S., Su, S.Y.W., and Lam, H., "An Association Algebra for Processing Object-oriented Databases", in: Proc. 7th IEEE Int'l Conf. on Data Engineering, Kobe, Japan, April 8-12, 1991.

[LAM92] Lam, H., Su, S.Y.W., et al., "GTOOLS: An Active Graphical User Interface Toolset for an Object-oriented KBMS", Int'l Journal of Computer Science and Engineering, Vol. 7, No. 2, April 1992, pp. 69-85.

[SHY91] Shyy, Y.M. and Su, S.Y.W., "K: A High-level Knowledge Base Programming Language for Advanced Database Applications", in the Proc. of the Conf. on Management of Data (SIGMOD), May 29-31, 1991.

[SU89] Su, S.Y.W., Krishnamurthy, V., Lam, H., "An Object-oriented Semantic Association Model (OSAM*)," Chapter 17, Artificial Intelligence: Manufacturing Theory and Practice, S. Kumara, A.L. Soyster, and R.L. Kashyap (eds.), Industrial Eng. and Management Press, 1989.

[SU92] Su, Stanley Y. W., and Lam, H., "An Object-oriented Knowledge Base Management System for Supporting Advanced Applications", 4th Int'l Hong Kong Computer Society Database Workshop, Hong Kong, Dec. 10-12, 1992.

[SU93] Su, S.Y.W., Guo, M.S., and Lam, H., "Association Algebra: A Mathematical Foundation for Object-oriented Databases", to appear in IEEE Trans. on Knowledge and Data Engineering, 1993.

[YAS91] Yaseen, R., Su, S.Y.W., and Lam, H., "An Extensible Kernel Object Management System", Proc. of OOPSLA '91, Phoenix, AZ, OCT. 1991.