METU Object-Oriented DBMS

Asuman Dogac Budak Arpinar Cem Evrendilek Cetin Ozkan Ilker Altintas
Ilker Durusoy Mehmet Altinel Tansel Okay Yuksel Saygin
TUBITAK Software Research and Development Center
Middle East Technical University
06531, Ankara Turkiye
asuman@srdc.metu.edu.tr

MOOD (METU Object-Oriented DBMS) is developed on the Exodus Storage Manager (ESM) [ESM 92], and supports a SQL-like object-oriented query language (MOODSQL) [Ozk 93, Dog 94] and a graphical user interface, called MoodView [Arp 93], developed using Motif. The system is coded in C++ on Sun Sparc II workstations and has a type system derived from C++, eliminating the impedance mismatch between MOOD and C++. ESM provides the MOOD the following kernel functions: Storage management, concurrency control, backup and recovery of data.

Additionally, the MOOD kernel provides the following functions : Catalog management, and optimization and interpretation of SQL statements. During this interpretation, functions (which have been previously compiled with C++) within SQL statements, are dynamically linked and executed. The advantage of this approach is to eliminate the interpretation overhead and increase the overall efficiency of the system.

Each object is given a unique Object Identifier (OID) at object creation time by the ESM which is the disk start address of the object returned by the ESM. The object encapsulation properties are supported through the public and private declarations of C++. Objects are grouped in the abstraction level of a class, in other words, classes have extensions. Class inheritance mechanism of the MOOD is multiple inheritance. The name resolution is handled as in standard C++. Aggregate definitions are handled in the MOOD system by introducing type constructors (Set, List, Ref and Tuple). Aggregate classes can be constructed by recursive use of these type constructors. The query optimizer of the MOOD is generated by using the Volcano Query Optimizer Generator. The Volcano Query Optimizer Generator [McK 93] provides for very fast and easy development of a query optimizer. The MOOD Optimizer uses database statistics obtained from the MOOD catalog in computing the selectivities and the costs for each optimization step. The set of MOOD Algebra operators is given in [Dog 94]. A graphical user interface, namely MoodView is implemented. MoodView provides the database programmer with tools and functionalities for every phase of OODBMS application development. Current version of MoodView allows a database user to design, browse, and modify database schema interactively. Furthermore, a database administration tool, a full screen text-editor, a SQL based query manager, and a graphical indexing tool for the spatial data, i.e., R Trees are also implemented.

References

[Arp 93] Arpinar, I. B., Dogac, A., Evrendilek, C., 'MoodView: An Advanced Graphical User Interface for OODBMSs', *SIGMOD Record*, Vol. 22, No. 4, December 1993.

[Dog 94] Dogac, A., Ozkan, C., Arpinar, B., Okay, T., Evrendilek, C., 'METU Object-Oriented DBMS,' *in Advances in Object-Oriented Database Systems*, Dogac, A., Ozsu, T., Biliris, A., Sellis, T., edtrs., Springer-Verlag 1994.

[ESM 92] Using the Exodus Storage Manager V2.1.1, June 1992.

[McK 93] McKenna, W. J., 'Efficient Search in Extensible Database Query Optimization: The Volcano Optimizer Generator', PhD thesis, Department of Computer Science, University of Colorado, 1993.

[Ozk 93] Ozkan, C., Dogac, A., Evrendilek, C., Gesli, T., 'Efficient Ordering of Path Traversals in Object-Oriented Query Optimization', *in Proc. of the Intl. Symp. on Computer and Information Sciences*, Istanbul, November 1993.