Middle East Technical University
Software Research and Development Center

Asuman Dogac, Director
Dept. of Computer Eng. 06531, Ankara, Turkey
Internet: asuman@srdc.metu.edu.tr
Tel: +90 (312) 210 12 98 Fax: +90 (312) 210 1259
WWW: http://www.srdc.metu.edu.tr

Middle East Technical University (METU) is the leading technical university in Turkey. The Software Research and Development Center was established by the Scientific and Technical Research Council of Turkey (TUBITAK) at the Department of Computer Engineering of METU in October 1991. The aim of this center is twofold: to lead large scale software research and development projects, and to foster international cooperation. SRDC is involved in a number of research and development projects supported by the government, industrial companies and international organizations. Although SRDC projects also cover other fields of computer science, the main emphasis is on database systems.

SRDC is organized around the ongoing projects and several engineers and graduate students work in these projects: M. Altinel, B. Arpinar, I. Cingil, Y. Ceken, C. Dengi, E. Gokkoca, C. Evrendilek, P. Karagoz, E. Kilic, P. Koksal, S. Mancuhan, S. Nural, F. Ozcan, G. Ozhan, V. Sadjadi, N. Tatbul. Its infrastructure includes a LAN with Sun Workstations and PCs and commercial software like Oracle7, Sybase, Informix, Adabas D, Ingres and DEC's ObjectBroker. SRDC is a beta test site for several products including SunSoft's Joe and Concerto and Orbix's Object Transaction Service. The remainder of this report describes the main projects.

I. METU Interoperable DBMS (MIND) Project

MIND is a multidatabase system based on OMG's distributed object management architecture. It is implemented on top of a CORBA compliant ORB, namely, DEC's ObjectBroker. In MIND all local databases are encapsulated in a generic database object. The interface of the generic database object is defined in CORBA IDL and multiple implementations of this interface, one for each component DBMSs, namely, Oracle7, Sybase, Adabas D and MOOD are provided. MIND provides its users a common data model and a single global query language based on SQL. Global layer of MIND consists of a global query manager, a global transaction manager, a schema integrator, interfaces to supported database systems and a graphical user interface.

The integration of export schemas is currently performed by using an object definition language (ODL) which is based on OMG's interface definition language. MIND global query optimizer aims at maximizing the parallel execution of the intersite operations of the global subqueries. Through MIND global transaction manager, the serializable execution of the global transactions (both nested and flat) is provided.

CORBA handles the heterogeneity at the platform level and in doing this it provides location and implementation transparency. In other words, the changes in object implementation, or in object relocation has no effect on the client. This reduces the complexity of the client code and allows clients to discover new types of objects added to the system and use them in plug-and-play fashion without any change in the client code. This feature of CORBA is very useful in registering new DBMSs to the system without affecting the already existing system and also this feature dramatically reduces the code that needs to be developed. Furthermore, CORBA and COSS (Common Object Service Specification) together provide much of the functionality to handle heterogeneity at the database level and some functionality to handle application interoperability. Note that COSS is a complementary standard developed by the OMG for integrating distributed objects.

The current implementation of MIND makes unified access possible to Oracle7, Sybase, Adabas D and MOOD (METU Object-Oriented Database System) through a global schema and a global query language based on SQL. When a client application issues a global SQL query to access multiple databases, this global query is decomposed into global subqueries and these subqueries are sent to the ORB (CORBA's Object Request Broker) which transfers them to the relevant database servers on the network. On the server site, the global subquery is executed by using the corresponding call level interface routines of the local DBMSs and the result is returned back to the client again by the ORB. The results returned to the client from the related servers are processed if necessary. This approach hides the differences between local databases from the rest of the system. Thus, what the clients of this level see are homogeneous DBMS objects accessible through a common interface.

The basic components of MIND are Global Database Agent (GDA) and Local Database Agent (LDA) class. A LDA class objects are responsible for maintaining export schemas provided by the local DBMSs represented in the canonical data model, translating the queries received in the global query language to the local query language providing an interface to the LDBMSs.

A GDA class objects are responsible from parsing, decomposing, and optimizing the queries according to the information obtained from the Schema Information Manager object, and also from global transaction management which ensures serializability of multidatabase transactions without violating the autonomy of local databases.

In addition to these, MIND has the following complementary components:

1. **Object Factory Server:** This server is responsible for the creation of other MIND objects such as LDA Object (LDAO), GDA Object (GDAO) and Query Processing Object (QPO). It provides the implementation of DBfactory Interface which has a single method, namely CreateObj. A client who needs an LDAO, GDAO, or QPO just calls the CreateObj method of FactoryServer. Once the FactoryServer is started, it creates an object of its own implementation and writes its object reference to the Advertisement Partition of ObjectBroker's Registry.

2. **Ticket Server:** This server provides a globally unique, monotonically increasing ticket number at each time it receives a request. These unique ticket numbers are used in the transaction management process of MIND. This server is started at the initialization of MIND system and it serves the whole MIND system continuously.

3. **Schema Information Manager:** This server, namely the SchemaServer provides and manages the global schema information necessary for the decomposition of global queries into subqueries. It provides the implementation of SchemaInt Interface which has methods necessary for processing the global schema information. SchemaServer is also started at the initialization phase of MIND system and stays alive during the life-time of the system. Query Manager is a client of this server.

4. **Transaction Manager:** This server is started on demand by the ORB and is responsible from the execution and global serializability of both flat and nested transactions. Its infact a part of GDA and is embedded in GDA together with Query Manager.

5. **Query Manager:** This server is also started on demand by the ORB and is responsible from the decomposition of global queries into subqueries. It gets the schema information necessary for the query decomposition from the SchemaServer. Query Manager is also a part of GDA. In addition to the decomposition of global queries, Query Manager also performs query optimization.

6. **Query Processor:** This server is responsible for query processing. It minimizes the total query processing time by enabling parallel execution of subqueries. It performs the necessary operations (such as join, outer-join, and union) for processing partial results coming from the local DBMSs. A Query Processors is started by the ORB as a result of a request from the Query Manager indicating that two partial results to be processed together are ready at the LDAs.

Funding: This project is currently being supported by the Scientific and Techical Research Council of Turkey, State Planning Organization (Turkey), Motorola (USA), Assistance (Austria), Sevgi Holding (Turkey).

A partial list of publications related with this project appears next.

1. Evrendilek, C., Dogac, A., Nural, S., Ozcan, F., "Query Optimization in Multidatabase Systems", in Proc. of the Next Generation Information Technologies and Systems, Israel, June 1995.

2. Kilic, E., Ozhan, G., Dengi, C., Kesim, N., Koksal, P., Dogac, A., "Experiences in Using CORBA for a Multidatabase Implementation", 6th International Workshop on Database and Expert Systems Applications, London, September 1995.

3. Dogac., A., Dengi, C., Kilic, E., Ozhan, G., Ozcan, F., Nural, S., Evrendilek, C., Halici, U.,

Arpinar, B., Koksal, P., Kesim, N., Mancuhan, S., "METU Interoperable Database System", in ACM SIGMOD Record, Vol.24, No.3, September 1995.

4. Dogac., A., Dengi, C., Kilic, E., Ozhan, G., Ozcan, F., Nural, S., Evrendilek, C., Halici, U., Arpinar, B., Koksal, P., Kesim, N., Mancuhan, S., "A Multidatabase System Implementation on CORBA", 6th Intl. Workshop on Research Issues in Data Engineering (RIDE-NDS '96), New Orleans, February 1996.

5. Dogac, A., U. Halici, E. Kilic, G. Ozhan, F. Ozcan, S. Nural, C. Dengi, S. Mancuhan, B. Arpinar, P. Koksal, C. Evrendilek, "METU Interoperable Database System", Demo Description, In Proc. of ACM Sigmod Intl. Conf. on Management of Data, Montreal, June 1996.

6. Ozcan, F., Nural, S., Koksal, P., Evrendilek, C., Dogac, A., "Dynamic Query Optimization on a Distributed Object Management Platform", Fifth International Conference on Information and Knowledge Management, Maryland, USA, November 1996.

II. METU Object-Oriented DBMS (MOOD)

MOOD is an object-oriented DBMS prototype developed on the Exodus Storage Manager (ESM). MOOD kernel provides the optimization and interpretation of SQL statements, dynamic linking of functions, and catalog management. SQL statements are interpreted whereas functions (which have been previously compiled with C++) within SQL statements are dynamically linked and executed. Thus the interpretation of functions are avoided increasing the efficiency of the system. A query optimizer is implemented first by using the Volcano Query Optimizer Generator and then with Cascades Query Optimizer Framework. The Cascades Query Optimizer Framework, which is being used in Microsoft's forthcoming SQL Server and Access query optimizers as well as Tandem's NonStop SQL product, is a tool to help the database implementor (DBI) in constructing a query optimizer for a DBMS. A region based query optimizer where different types of queries are optimized by using different search strategies in each region is developed for MOOD through Cascades. A graphical user interface, namely MoodView, is developed using Motif. MoodView displays both the schema information and the query results graphically. Additionally it is possible to update the database schema and to traverse the references in query results graphically.

The system is coded in GNU C++ on Sun Sparc 2 workstations. MOOD has a SQL-like object-oriented query language, namely MOODSQL which can be used both through MoodView and through the textual interface. MOOD has a type system derived from C++, eliminating the impedance mismatch between MOOD and C++. The users can also access the MOOD Kernel from their application programs written in C++. For this purpose MOOD Kernel defines a class named UserRequest that contains a method for the execution of MOODSQL statements.

Exodus Storage Manager (ESM) has a client-server architecture and each MOOD process is a client application in ESM. ESM provides the MOOD some of the kernel functions like storage management, concurrency control, backup and recovery of data. Additional kernel functions provided by the MOOD are the optimization and interpretation of SQL statements, the dynamic linking of functions and catalog management. During the interpretation of SQL statements functions (which have been previously compiled with C++) are dynamically linked and executed. This late binding facility is essential since database environments enforce runtime modification of schema and objects. With our approach, the interpretation of functions are avoided increasing the efficiency of the system.

The main components of MOOD are as follows:

1. **Catalog management:** The MOOD catalog contains the definition of classes, types, and member functions in a structure similar to a compiler symbol table. SQL-like object-oriented query language, MOODSQL: Functionality of MOODSQL commands can be divided into two parts: Schema definition and modification commands, and data manipulation commands.

2. **A dynamic link facility:** During the interpretation of MOODSQL statements, class methods (which have been previously compiled with C++ compiler of MOOD, namely MOODCC) are dynamically linked and executed. A GUI based utility, MOODView: MOODView displays both the schema information and the query results graphically. Additionally, it is possible to update the database schema and to traverse the references in query results graphically.

3. **A C++ library interface:** This interface enables end users to develop advanced applications based on MOOD. These applications can access the MOOD kernel functions in order to define, create, retrieve and delete objects in MOOD.

MOOD source code is available from the site *http://www.srdc.metu.edu.tr* along with its related documents.

Funding: This project is supported by the Scientific and Technical Research Council of Turkey.

Some of the related publications are listed next.

1. Dogac , A., Evrendilek, C., Okay, T., Ozkan, C., "METU Object- Oriented DBMS", in Object-Oriented Database Systems, edited by Dogac, A., Ozsu, T., Biliris, A., Sellis, T., pp.172-198, Springer-Verlag, 1994.

2. Arpinar, B., Dogac, A., Evrendilek, C., "Mood-View: An Advanced Graphical User Interface for OODBMSs", in ACM SIGMOD Record, Vol.22, No.4, December 1993.

3. Koc, K., Dogac, A., Evrendilek, C., "Comparison of Clustering Algorithms in a Single User Environment through 007 Benchmark", in Proc. of East-West Database Workshop, Austria, September, 1994.

4. Aytekin, H. , Dogac, A. "A Distributed Parallel Object Manager for Smalltalk", in Proc. of Ninth Intl. Symposium on Computer and Information Sciences, Antalya, November, 1994.

5. Dogac, A., Altinel, M., Ozkan, C., Durusoy, I., Altintas, I., "METU Object- Oriented DBMS Kernel", DEXA'95, Lecture Notes in Computer Science, Springer Verlag, 1995.

6. Ozkan, C., Dogac, A., Evrendilek, C., "A Heuristic Approach for Optimization of Path Expressions in Object-Oriented Query Languages", 6th International Conference on Database and Expert Systems Applications, London, September 1995 (Lecture Notes in Computer Science, Springer Verlag 1995).

7. Ozcan, F., Nural, S., Koksal, P., Altinel, M., Dogac, A., "A Region Based Query Optimizer through Cascades Query Optimizer Generator", IEEE Data Engineering Bulletin, Special Issue on Query Optimization, Vol. 18, No. 3, September 1995.

8. Ozkan, C., Dogac, A., Altinel, M., "A Cost Model for Path Expressions in Object-Oriented Queries", Journal of Database Management, Vol.7, No.3, June 1996.

SRDC has organized a NATO Advanced Studies Institute on the advances in Object-Oriented Database Systems which took place during August 6-15, 1993 in Kusadasi, Izmir. The invited talks given during this institute is collected in the following reference:

- Dogac, A., Ozsu, T., Biliris, A., Sellis, T., Advances in Object-Oriented Database Systems, Springer-Verlag, 1994.

III. Hospital Information System

Healthcare Information systems (HIS) are well known for their complexity and thus require a lot of effort in implementation and in maintenance. A HIS is designed and developed for a private hospital. Its architecture is based on client/server model. In the realization of this system Microsoft Visual Basic is used as the Graphical User Interface (GUI) builder of the client programmes and connection to Oracle Server (Release 7.2.3) is provided by Oracle Glue. Main design goals of this project include flexibility in adoption to the new requirements, user friendly GUI, and high performance. The system is also designed to be easily portable to other hospitals. This product is being marketed and maintained by BYS Company. Detailed information can be obtained in the following reference.

- Ceken, Y., Altinel M., Guven, H., Altintas, I., Dogac, A., "Experiences in Design and Implementation of Healthcare Information System", International Oracle User Week, September, 1996, Dallas, Texas.

IV. Database Design Tools

A number of database design tools have been developed at SRDC. GESDD (Generalized Expert System for Database Design) attempts logical database design whereas MAESTRO 7 does index selection.

GESDD is a compound expert system made up of two parts: an expert system for generating methodologies for database design, called ESGM; and another expert system for database design, called ESDD. ESGM provides a tool for database design expert to specify different design methodologies or to modify existing ones. The database designer uses ESDD in conjunction with one of these methodologies to design a database starting from the requirement specification phase and producing a logical schema in one of the well-known data models. The database designer interacts with ESDD in the requirements specification phase to test the conceptual schema produced by ESDD. The system is evolutive in the sense that an existing methodology can be modified or a new methodology can be added to the existing ones. GESDD is menu driven and it is coded in Prolog.

The index selection problem is NP-Complete and therefore requires heuristic to be used. Heuristic must be based on the query optimizer principles because the indices decided should be usable by the query optimizer. It should be noted that since the query optimization principles of commercial DBMSs vary to a great extent, it is infeasible to implement a general purpose index selection tool.

MAESTRO 7 (METU Automated indEx Selection Tool foR Oracle7) which is an index selection tool for Oracle7 assists the database administrator in designing an index configuration for a given database application for Oracle7. It decides on the complete set of indices both primary and secondary by considering the index maintenance costs. MAESTRO 7 automatically derives valid SQL statements and their usage statistics during regular database sessions through SQL Trace Facility. It classifies the SQL statements that will produce the same execution plan and accumulates their weights. MAESTRO 7 also decides on plausible columns for indexing at the beginning, before the more time consuming processing on the queries start, to improve performance of the tool. The selection of primary indices cannot be done independently for each table for multi table queries. Therefore in choosing primary indices, all combinations of plausible primary indices are considered. The benefit of using the indices are calculated by obtaining the cost of executing the queries from the query optimizer. Although this tool is specific to Oracle7, a similar tool for any DBMS that has a cost based optimizer can be developed very easily following the ideas presented in MAESTRO 7.

Some of the related publications are listed next.

1. Dogac, A., "An Expert System for Physical Design of Relational Databases," Proc. of Expert Systems Workshop, Houston, Jan. 1991.

2. Dogac, A., Erisik, A., Ikinci, A., "MAESTRO 7: An Index Selection Tool for Oracle 7", Proc. of Intl. Workshop on Next Generation Database Systems, Honkong, Feb. 1994.

Finally, SRDC has undertaken the development of AWACS Man-Machine Interface Prototype for NATO Mid-Term Modernization Programme in cooperation with Gallium Software Inc.(Canada).

All the information related with these projects can be obtained from *http://www.srdc.metu.edu.tr*.