ODMG-93: A Standard for Object-Oriented DBMSs

R. G. G. Cattell (ODMG Chair)
SunSoft, Inc, Mountain View, California. USA

The Object Database Management Group (ODMG) is a consortium of object-oriented DBMS vendors that have developed a standard interface for their products, ODMG-93. The standard includes a common architecture and definition for an object-oriented DBMS, a common object model with an object definition language, a common object query language, and standardized programming language bindings, currently for C++ and Smalltalk. An object-oriented DBMS (by ODMG's definition) provides programming language bindings with direct, transparent persistence for data structures, in contrast to the embedded language bindings used in most DBMSs. The common object model allows data to be shared across programming languages. This model incorporates object IDs, encapsulation, methods, first-class types, multiple inheritance, relationships, lists, sets, bags, arrays, and many other features.

Since there was no standard to use as a starting point in the area of object databases (as there was with SQL for early relational products), substantial creative effort was required to arrive at ODMG-93. ODMG was formed because other standards efforts are working on a different kind of data management problem, and are doing so with an organizational methodology that was not well-suited to the cooperative creative effort required to converge on a proposal in a timely fashion.

ODMG currently has 13 members. Several of the members already have partial implementations of ODMG-93, and all of the vendor members have committed to support it by early 1995, so it is expected to become a de facto standard for this industry. Hopefully this standard will help to focus new research and development in object databases, and will enable new work such as application development tools that work with more than one product. ODMG is now proceeding with future enhancements.

ODMG-93 differs from current ANSI SQL3 work on several important dimensions. Architecturally, ODMG-93 is based on extending existing programming languages to support persistent data and database functionality. In a SQL-based system, the programmer writes embedded statements to convert programming language data structures into database data structures and back again. ODMG does define programming language-independent data definition and query languages based on an object model shared across languages, but programmers have the option to use equivalent syntax based on their favorite programming language.

Unlike SQL3, ODMG chose not to extend SQL to create another programming language, because many programmers prefer an established programming language and development environment. ODMG instead chose to combine SQL syntax with OMG object model extensions to allow declarative queries based on a clean, more powerful data model. For example, the result of a query (and the value of any named entity) in the current SQL3 must be a collection of rows, while in ODMG they may be of any type.

ODMG members do hope to work with the SQL3 group now that ODMG's more immediate needs have been met. Many of the ODMG vendors already support SQL2 in their products, and ODMG can easily fit SQL3 into their architecture by making SQL3 another language binding, with interoperation and sharing with the C++, Smalltalk, and other language bindings. In this way an object database customer could choose between using database functionality integrated into a programming language, or using SQL3 embedded in the traditional way.

ODMG is affiliated with OMG and is based on OMG standards for the object model and definition language. The ODMG members are also members of OMG task forces, and OMG has adopted the ODMG-93 interface as part of the Persistence Service, which is one of OMG's Object Services. OMG Object Services go somewhat beyond ODMG's goals in that OMG is defining mix-and-match standards by which a customer acquires relationships, transactions, persistence, queries, and other functionality from different vendors. This architecture provides more flexibility than ODMG's at the price of some trade-off in complexity. Fortunately, OMG's interfaces can nicely be accommodated in ODMG's architecture by making OMG IDL-based interfaces be another language binding for the ODMG framework, alongside C++, Smalltalk, SQL3, and other bindings ODMG may define. ODMG's architecture allows a "best of both" architecture where each binding is well-integrated with its host programming environment, yet databases and the data model can be shared across languages and environments.