

Object-oriented databases: systems and standards

Byung S. Lee, University of St. Thomas

Since the mid-eighties when Gemstone was introduced as the first object-oriented database management system (ODBMS), a dozen other commercial ODBMSs have joined the fierce competition in the market. Although we call them all ODBMSs, they differ in their system concepts and data management standards. Below, I discuss the past evolution and future prospects of those ODBMSs.

Evolution of systems

There have been three approaches to building an ODBMS: extending an object-oriented programming language (OOPL), extending a relational DBMS, and starting from the ground up. The first approach realizes an ODBMS by adding to an OOPL persistent storage for multiple concurrent accesses with transaction support. This extension has the advantage of reusing the type system of a programming language as a data model and thus achieves a seamless integration between programming language and database manipulation language. This approach has become the most popular in the commercial world so far and is represented by commercial ODBMSs such as ObjectStore, Versant, Objectivity, and O2.

In the second, extended relational approach, an ODBMS is built by enhancing an existing relational DBMS with object-oriented features such as classes and inheritances, methods and encapsulations, and complex objects. Exemplary systems are the research prototypes Postgres95 and Starburst, which were incorporated into commercial products Illustra and DB2/6000 V2, respectively.

The third approach is revolutionary in the sense an ODBMS is built from the ground up, as represented by UniSQL and OpenODB. Research prototypes like Orion and Zeitgeist belong to this category as well. These systems provide their own proprietary data models and data manipulation languages.

Lately, a new paradigm of ODBMS, called object-relational DBMS (ORDBMS), has begun to draw increasing attention. The objective of an ORDBMS is to support both relational and object-oriented database applications. Systems in the category of ORDBMSs at the time of this writing are extended relational DBMSs such as Illustra and DB2/6000 and ground-up DBMSs such as OpenODB and UniSQL.

Evolution of standards

In September 1991, the Object-Oriented Database Task Group of the ANSI Database Systems Study Group published its final report for establishing "a framework for future standards activities in the object information management area." Subsequently, two ongoing efforts have begun -- one by the Object Data Management Group (ODMG) and the other by the American National Standards Institute (ANSI) SQL3 (also called Object SQL) committee. In October 1993, ODMG published its first version of standard, ODMG-93, which defines the data model, query language, and language bindings with which all commercial ODBMS systems are advised to comply. The ANSI SQL3 committee has not yet finished developing a full-fledged standard of SQL3. Their target publication years are 1997 for some major components and 1998 for the other components. Both standards are still evolving and in some sense competing with each other. This lack of a solid, unified standard is a barrier to be overcome as soon as possible.

Future of systems

There is no doubt that ODBMSs will gain increasing market share. Particularly ORDBMSs will gain more popularity because of their dual support for relational and object-oriented data management capabilities. Virtually all commercial ODBMSs will move in that direction. One interesting question in this regard is who will arrive there first. Right now, Illustra, UniSQL, OpenODB, and DB2/6000 V2 are considered the first ORDBMSs. Other ODBMSs will follow the same track. On the other hand, conventional relational DBMSs including Informix, Ingres, Oracle, and Sybase may gradually be turning into ORDBMSs. However, their current object-oriented extension is limited to providing SQL3 wrappers and storing complex data as binary large objects (BLOBs) and thus entails poor query processing and optimization. Nonetheless, some users may be willing to wait for relational DBMS vendors to release full-fledged OO extensions.

ODBMSs will not replace relational DBMSs in conventional database markets such as inventory management, airline reservation, finance and investment management. Rather, the use of ODBMSs will be restricted to complex applications such as design engineering and network management. Geographic information systems and electronic book technology are emerging as new areas of complex applications. Ironically, hierarchical DBMSs, such as IMS, will still dominate in terms of deployed data volume for at least another decade.

Future of standards

The ODMG standard will soon emerge as the strongest standard in commercial ODBMS community and influence current commercial ODBMSs. A few commercial products in compliance with ODMG-93 are soon to be released, and a revised version of the ODMG standard, ODMG-95, is on its way. ANSI SQL3 covers more comprehensive features such as rules and triggers in addition to object-oriented features but its current progress is slow. SQL3 will be the primary target standard for extending conventional relational DBMSs such as Informix, Oracle, and Sybase with OO features. Both ODMG-95 and ANSI SQL3 will add impetus to the ORDBMS trend.

We may see the recurrent phenomenon of vendors running ahead of ANSI committees, which was evident with the Internet protocol -- for example, TCP/IP against ANSI X.25. Eventually, however, it is likely (and desirable) that the two streams of standards, ODMG and SQL3, will influence each other or even merge.

References

1. E. Fong, W. Kent, K. Moore, and C. Thompson (eds.), "X3/SPARC/DBSSG/OOBTG Final Report," September 1991.
2. J. Blakeley, "Object-oriented database management systems," Tutorial at SIGMOD, Minneapolis, MN, May 1994.
3. R. Cattell (ed.), "The Object Database Standard: ODMG-93," Morgan Kaufmann, 1994.
4. J. Melton and N. Mattos, "Database Language SQL," Tutorial at SIGMOD, San Jose, CA, May 1995.

Byung S. Lee is an Assistant Professor of Graduate Programs in Software at the University of St. Thomas in St. Paul, Minnesota. E-mail: bslee@stthomas.edu.