Object-Oriented Database: How Wide is the Application Range?

The information systems field is a dynamic one in which we are constantly moving the state-of-the-art forward. Database management systems are very much a part of this movement. We have never been satisfied with the current DBMS paradigm, always looking for a new and better approach. Navigational DBMS was followed by Relational DBMS and now the latest embodiment of this trend is Object–Oriented Database Management Systems (OODBMS).

The OODBMS paradigm is striking in that it includes several truly advanced features including abstract data types, encapsulated code, and inheritance hierarchies, plus, of course, the concept of the object itself. Quite naturally, the direction that applying the new OODBMS concept has taken is in the exploitation of the new features in application areas that did not lend themselves well to the relational or navigational models. Thus, we have seen OODBMS applications in rather complex data storage applications such as electric power grid management, computer-aided design for VLSI circuits, computer-aided software engineering, geographic information systems, medical decision support, and document browsers, among others.

While these are interesting and important applications, I cannot help but wonder if on a broader scale we are missing the point and missing the boat. Database management, as a technology, has been successful at least in part because of its very broad applicability. While in the past it has perhaps not been applied to the relatively exotic applications mentioned above, it has been applied to the wide range of common, “bread-and-butter” applications in accounting, inventory control, customer account management, and so forth, in virtually every conceivable industry. In fact, it was so successful in the navigational era, that when the relational era emerged, there was widespread resistance—and there still is—to converting the existing data and programs to relational technology because of the huge investment that had been made originally in, what are now, smoothly running applications. In most companies today, new applications are based on relational technology and there certainly has been activity in converting some navigational applications to relational technology, but there has also been a great deal of effort expended in devising ways to continue to use the so-called “legacy” systems in the modern environment. Nevertheless, it is clear that relational technology, barring the factors just mentioned, could replace navigational technology. Yes, there are some comparative advantages and disadvantages (the ease-of-use of the relational query interface) and some lingering controversies (relational versus navigational performance in some applications) but overall, everyone recognizes that relational technology is capable of handling common applications.

The question now is whether object-oriented