

Looking for the Objects in Object-Relational DBMSs

Panel Session

Moderator: Lougie Anderson

Panelists: Mike Carey, Ken Jacobs, Erin Kinikin, David Maier

Abstract: The Relational Model first came into vogue in the early 1980's. It was based on the simplifying idea that all data could be modeled as mathematical relations (tables in "normal form"). Permissible operations on this table data structure were specified by the relation algebra and calculus. The Relational Model led to years of research in areas such as query languages, query optimization, transaction models, and database design methodologies. This research has dominated DBMS conferences for the last fifteen years and has also lead to major products offerings in wide-spread use in the computer industry today.

While the Relational revolution was happening in the DBMS community there was a minority opinion emerging from the object community. This minority opinion surfaced in heated debates and panels at DBMS conferences where it was often pitted against "relational purists." The object proponents wanted to discuss storing and retrieving complex objects and relationships in databases while the relational purists insisted on maintaining "mathematical purity and simplicity". These panels were often some of the most acrimonious (and entertaining) at these conferences and many thought there was no way to bridge the chasm between the two schools of thought.

However times changes and so do the realities of the commercial world. Object languages such as C++, Smalltalk, and Java have become *de facto* standards. The Internet and the PC have increased the demand for complex data types. And the relational model is evolving to accommodate these realities.

All of the major RDBMS vendors have announced plans for, or are already shipping, Object-Relational DBMS (ORDBMS)

products. A natural question for the object community is how well these new products will address the well-known "impedance mismatch" between a pure object model and the relational model. For example, how does one make Java, Smalltalk, or C++ objects persist using an ORDBMS? Can one search for these objects in the database using their methods?

The vendors for ORDBMS are claiming well-known OO features in their implementations, such as extensible data types, inheritance, object identity, and object language bindings. This panel will allow the major vendors to explain how those features match up with the kinds of object models that OO developers are accustomed to. The panel will consist of representatives from three of the major ORDBMS vendors as well as a representative of "pure object think". The vendor representatives will each present a brief overview of the object-related features of their products and will explain why OO programmers are going to have an easier time with these ORDBMSs. This will be followed by counterpoint discussion from the pure object thinker.

Lougie Anderson is employed by Sequent Computer Systems.

Mike Carey is a Research Staff Member at the IBM Almaden Research Center.

Ken Jacobs is Vice President of Product Strategy at Oracle.

Erin Kinikin is Executive Director of Product Management at Informix.

David Maier is a Professor at the Oregon Graduate Institute.