Of Objects and Databases: A Decade of Turmoil

Carey, M.J.; DeWitt, D.J. (1996)

Presented by:
Ricardo Pedrosa
Jian Xu

Areas of research

**Extended relational database systems**
- Allow the addition of new, user-defined abstract data types (ADTs).
  - ADTs are implemented in an external language.
  - After being registered with the database, ADT's functions can be used in queries.
- Projects:
  - Ingres
  - Postgres
    - Query optimizers with ADT's properties and functions awareness.
    - Support for storing and querying complex data types.

**Persistent Programming Languages**
- Add data persistency and atomic program execution to traditional object-oriented programming languages.
- Problems addressed:
  - Orthogonality.
  - Persistence models.
  - Binding and namespace management for persistent roots.
  - Type systems and type safety.
  - Alternative implementation techniques for supporting transparent navigation, maintenance, and garbage collection of persistent data structures.

**Object-Oriented Database Systems**
- Combine all of the features of a modern database system with those of an object-oriented programming language, yielding an object-oriented database (OODB) system.
- Focused on:
  - Support for querying, indexing and navigation.
  - Addressing version management needs of engineering apps.
- Projects:
  - Gemstone (Smalltalk).
  - Vbase (CLU-like language).
  - Orion (CLOS).

**Database system toolkits/components**
- Provide a DBMS that can be extended at almost any level, using mostly kernel facilities plus additional tools that help building domain-appropriate DBMS.
- Projects:
  - EXODUS:
    - Storage manager for objects
    - E: a persistent Prog. Language.
    - Query optimizer generator.
  - Starburst:
    - Clean architectural model that facilitates storage and indexing extensions.
    - Rule-based extensible query subsystem.
What happened?

- System toolkits & persistent programming languages:
  - In spite of some interesting results these were a failure from a commercial point of view.
- OO database systems:
  - Many results from the academic point of view. Not expanded commercially as expected by its developers.
- Language-specific object wrappers for relational databases:
  - New approach that appears to be important for building OO, client side apps.
- Extended relational DBS:
  - Renamed as Object-Relational DBMS. Appears to be settling in terms of providing objects for enterprise DB apps.


The Database Toolkit approach problem.

- Require a lot of expertise.
- End up in being inflexible, awkward or incomplete.
- As OO and O-Relational database systems provide enough extensibility, it’s not worthy to start from scratch even given a toolkit to help in the process.

Why EXODUS failed?

- The client/server architecture introduced an unwanted level of indirection when users tried to use EXODUS to implement their own object servers.
- E programming language: Too general for skilled database implementors and too low-level for application-oriented programmers.
- The query optimizer was inefficient and hard to use.

Was all that bad after all?

- Interesting research by-products relevant to OODBMS and ORDBMS.

Discussion

Q1: Do you believe there will be an OODB that overcomes the shortcomings of the above causalities? Give reasons.
Object-Oriented Database Systems (OODBMS)

What went wrong with OODMS?
- Lack of standards.
- OODBMS products are behind RDBMS in some terms (e.g., no view facilities).
- Painful schema evolution.
- Tight coupling between an OODBMS and its application programming Language.
- Low availability of application development tools.

Object-Relational Database Systems (ORDBMS)

Main tenets for ORDBMS
- Provide support for richer object structures.
- Subsume RDBMS.
- Be open to other subsystems (tools and multidatabase middleware products).

What ORDBMS should provide?
- A rich type system, inheritance, functions and encapsulation, optional unique ids and rules/triggers.
- A high-level query based interface, stored and virtual collections, updatable views and separation of data model and performance features.
- Accessibility from multiple languages, layered persistence-oriented language bindings, SQL support and a query-shipping client/server interface.

A vision from 1996 of databases in 2006

Fully integrated solution
Object relational servers will provide:
- Support for OO ADTs.
- Inheritance among ADTs.
- ADT implementation in various programming languages.
- Full OO support for row types.
- Support for middle-tier and desktop applications.
  - Provide a development environment where the same object model will describe the DB in all levels, both for querying and navigational programming.
  - Methods and queries will be run on cached data on servers or clients depending on where's faster.

Research Challenges
- Server functionality and performance
- Client integration
- Parallelization
- Legacy data sources
- Standards

Discussion
- Based on your knowledge of DB’s, how much of their vision came true? (If the reality differs from vision, how?)
- Why does this differ from their vision?
  - list facts that pushes (or will push) development
  - list facts that stunts development
- Predict the future: What do you expect OODB and ORDB to be in year 2016