

# The Objects War

SUMMARY OF THE OBJECT-RELATIONAL IMPEDANCE MISMATCH



# Historical Background

- Relational Model (1970s)
  - Data organized in tables and rows
    - Table relationships as FK – CK
  - Formal foundation on set theory
    - Algebraic and calculus-based techniques for querying
  - Operational applications (table and columns like)

But...

- New application domains and new needs
- OOPLs began to develop
  - User-defined classes, methods, encapsulation...
- Normalization affects query performance!

# Object Data Models: OODBs

- Object-Oriented Database Systems (OODBs) in the mid-1980's
  - The *Object-Oriented Database System Manifesto*
- Main claims:
  - Better support for programming languages
  - Support for complex structures / objects
    - Non-operational applications
- Main features:
  - Data objects organized as user-defined classes
    - Hierarchies
    - Inheritance (of attributes and behaviour)
  - Classes relationships by means of OIDs
  - Uniformity
    - OOPs as single language to access the database and implement database applications (no impedance mismatch)

# ODMG: Object Definition Model

- Standard Object-Oriented data model
  - *Object Model*
  - *Object Specification Languages*
    - The ODMG Object Definition Language (ODL)
    - The ODMG Object Interchange Format (OIF)
  - *Object Query Language (OQL)*

# Object Data Models: ORDBs

- Object-Relational Database Systems (ORDBMS)
  - the *Third Generation Database System Manifesto*
- Main claims:
  - Relational technology SHOULD NOT be put aside
  - Object features as an extension to relational DBMSs
- Main features:
  - User-defined data types
  - Object tables formed from user-defined types
  - Hierarchies
  - Support for OIDs (also as a mean to relate objects)

# The OODBs Manifesto

The Golden Rules		Optional	Open
OO Data Model	DBMS		
Complex Objects	Persistence	Multiple Inheritance	Programming Paradigm
Object Identity	Secondary Storage Management	Type Checking and Inferencing	Representation System
Encapsulation	Concurrency	Distribution	Type System
Types / Classes	Recovery	Design Transactions	Uniformity
Inheritance	Ad hoc Query Facility	Versions	
Overriding, Overloading, Late Binding			
Extensibility			
Computational Completeness			

# The ORDBs Manifesto

## □ 3 Tenets (Third Generation DBMSs):

- Traditional DBMS services + support for richer object structures and rules (i.e., constraints)
- Must subsume second generation DBMSs
- Must be open to other subsystems

Object and Rule Management	DBMS Function	Towards an Open System
Rich Type System	Non-procedural, high-level access language	Accessible from multiple high-level languages
Multiple Inheritance	Support collections: enumeration of members or using the query language	For better or worse: SQL
Encapsulation		Enhancement of DBMS – programming language interfaces
UIDs and PKs	Updatable views	Queries and answers as the lowest level of communication (client / server)
Rules Enforcement	Data independence	

# OODBs Vs. ORDBs

- Complex Objects, Types Classes, Extensibility, etc. Vs. Rich Type System
- Inheritance Vs. Multiple Inheritance
- Encapsulation
- Object Identity Vs. UUIDs and PKs
- DBMS main features: persistence, concurrency, recovery, etc.
- HLLs Computational Complexity Vs. SQL + constraints enforcement + updatable views + HLLs Interfaces
- Open Systems
- OODBMSs Vs. ORDBMSs



# OODBs Vs. ORDBs

- ~ • Complex Objects, Types Classes, Extensibility, etc. Vs. Rich Type System
- ~ • Inheritance Vs. Multiple Inheritance
- ✓ • Encapsulation
- ✗ • Object Identity Vs. UUIDs and PKs
- ✓ • DBMS main features: persistence, concurrency, recovery, etc.
- ✗ • HLLs Computational complexity Vs. SQL + constraints enforcement + updatable views + HLLs Interfaces
- ✓ • Open Systems
- ✗ • OODBMSs Vs. ORDBMSs

**NATIVE ACCESS Vs. IMPEDANCE MISMATCH**

# OODBs Today

- OODBs - Until 2000...
  - Most DBs have been discontinued
    - **Lack of a common data model** (each proposal relied on the HLL used to access the database structures)
    - Lack of strong theoretical framework
    - Strong experimental activity
      - Hope for a *de facto standard*!
- With the arrival of NOSQL they are back to life
  - Db4o, yet a running OODBs
    - <http://www.odbms.org/>
  - Other Current Examples
    - Objectivity/DB (Objectivity)
    - ObjectStore (Progress Software)
    - Versant's ODBMS (Versant)

# ORDBs Today

- Postgres (1987): First ORDBMS
  - Current RDBMSs support (some) object-relational features
- *Standard* object-oriented features (SQL:1999)
  - LOB (Binary Large Object) type
    - “Relational” feature to support object-oriented databases
    - Objects as serialized bytes stream
  - Row type (ROW)
    - A column containing several attributes
  - User defined types (UDTs)
    - Distinct types / Structured types (no objects!)
  - Typed tables
    - Each row as an object (object Vs. type)
  - Inheritance
    - At UDT level
    - Multiple inheritance not supported
  - REF Type
    - *Object pointer*
  - Collection type (ARRAY)
    - Multiple values