

# Modularity and Tuning Mechanisms in the O<sub>2</sub> System

Fernando Vélez  
O<sub>2</sub> Technology

The O<sub>2</sub> System is a commercial Object-Oriented Database Management System with a complete development environment and a set of user interface tools. In this presentation, we focus on the modularity and application tuning facilities of the system.

## Modularity

Class Inheritance and encapsulation are the key object-oriented concepts which have proven to help programmers in writing well structured, reusable and maintainable code. However, as soon as the number of classes and their corresponding inter-relationships increase, the information system becomes more and more complex. A new level of abstraction is needed to provide modularity.

Modularity facilities are provided in O<sub>2</sub> through the notions of *schema* and *database*. A schema is a collection of class definitions and persistent *name* declarations. A schema may *export* other classes and thereby allow other schemas to reuse (*import*) them. Details of the implementation of these exported classes are completely encapsulated inside schemas. The structure and method code of the exported classes are shared by the importing schema, therefore guaranteeing consistency. In addition, the imported classes can be re-defined locally by defining subclasses. These set of features are very important for code reusability. They allow object-oriented programming in the large.

A Database is a collection of objects whose structure and behaviour conform to the structural and behavioural definitions of the schema to which it is associated. Several different databases may be associated

to the same schema. The entry points to a database are the persistent names declared in its schema. As with schemas, databases may also be imported by importing one or more names. In this way, applications running within the context of a database may access another database by invoking a method which will run against this "remote" database. This mechanism allows data reusability, and is the basic model for communication with possibly heterogeneous databases.

## Application Tuning

Besides being able to control "classical" tuning parameters (e.g., size of the client and server caches, page fill factor, etc), O<sub>2</sub> allows a database administrator to optimize I/O operations on large databases by using clustering and indexing mechanisms.

A DBA is able to exercise control of placement of objects on disk via *cluster trees*. A cluster tree expresses the way in which a complex object and its components will be clustered together. Cluster trees may be rooted on named objects, or on classes of objects. Arcs of the tree correspond to composition links. Clustering conflicts due to sharing may arise. The DBA may want to control the way in which they are resolved, and to do so, he/she may order the cluster trees by priority.

Indexes may be defined on named collections. We preferred to index on individual collections rather than indexing on class extents (i.e., a system-defined collection gathering all members of a class) because in this way only applications using the indexed named collection bear the overhead of index maintenance. In addition, inheritance is naturally taken into account because of collections being polymorphic. Indexes may be defined on composition paths, allowing the O<sub>2</sub>SQL query language optimizer to deal efficiently with navigation-oriented accesses to objects.

Both indexes and cluster trees are data independent, "physical schema" information attached to a given database.

---

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

SIGMOD /5/93/Washington, DC,USA

© 1993 ACM 0-89791-592-5/93/0005/0440...\$1.50