The Design and Implementation of $O_2$, an Object-Oriented Database System

Francois Bancilhon, Gilles Barbedette, Véronique Benzaken, Claude Delobel, Sophie Gamerman, Christophe Lécruise, Patrick Pfeffer, Philippe Richard and Fernando Velez

Abstract

This paper describes the design and implementation of the first version of the $O_2$ object-oriented database system. The Altaïr project is currently designing and implementing a complete environment for developing data intensive applications. As a first step, we have implemented a quick prototype of an object-oriented DBMS. We report here on this implementation. The first part of the paper describes the system interface as seen by the application programmer, the second part describes the architecture of the system.

1 Objectives and assumptions

One of the major objectives of Altaïr is to prototype the DBMS of the 1990's. More precisely, we want to build a complete development environment for data intensive applications. Thus, the functionalities of the system should include those of a DBMS, those of a programming language and those of a programming environment. In this paper, we specifically report on the merge of programming language technology and database technology. Other reports [Cazalens et al 88] cover the user interface part of our effort.

The target applications for our system are (i) traditional applications such as business and transactional (excluding however very high performance transaction processing systems), (ii) office automation applications and (iii) spatial data management (such as geographic data management). Note that, at this stage of the game, no specific emphasis is given to CAD/CAM, CASE or knowledge base applications (whatever these may be),

$^1$Altaïr is a consortium funded by IN2 (the computer subsidiary of the Intertechnique group), INRIA (the French National Institute of Research in Computer and Control) and LRI (the Computer Science Laboratory of the University of Paris XI). The consortium started in September of 86 and is created for 5 years.
but we believe that, in a later stage, the system could be enhanced to serve also these applications.

The target customers of our system are (i) application programmers and (ii) end users. Our main interest is in application programmers, since we consider the major problem is to increase programmer productivity. We are not interested in the so-called naïve user and we assume that both end users and application programmers have a reasonable training with the system and use it on a regular basis.

The hardware configuration we are targeting at, consists of a single server connected to a set of workstations. Three facts are important: (i) computing power is more and more in the workstation rather than in the server, thus the ultimate workload left on the server will be archiving and data sharing (ii) workstations are chosen by the customer and thus the set of workstations will be heterogeneous and (iii) we believe that restricting ourselves to the single server case makes sense. This does not preclude the server from talking to another server to get its data, it justs imposes that, in the context of one application, each workstation sees only one server which is in charge of archiving and sharing data.

From a software point of view, our objective was to use, as much as possible existing products. The project was started with the idea of developing the server software around the Pick operating system and Pick file management system. We also had planned on using Score, a transaction management software developed by IN2 on top on Pick. The first prototype was built with these ideas in mind; emphasis has now been shifted differently and we are examining alternate solutions based on Unix \(^\text{2}\) for the next prototype.

We made the decision, to meet all these requirements, to build an object-oriented database system, named \(O_2\) and its programming environment. Our motivations for this choice are the following:

- We do believe that the main bottleneck to the productivity of the application programmer is the impedance mismatch between the programming language and the query language. This impedance mismatch cannot be solved by redefining the database box (i.e. by changing the frontier between the programming language and the database system) but by mixing database technology and programming language technology to build a complete system which will have the functionalities of a DBMS and of a programming language.

- We do believe that, among the available technologies produced by programming language people and among the possible approaches, the object-oriented approach is the best one to mix with database technology. This is due both to the intrinsic characteristics of the approach and to the appeal this paradigm has to programmers.

The next choice concerned the programming language of the system. Among the possible solutions (extend an existing language, design a new language or try to be language independent), we have chosen the last one mainly for marketing reasons (from a pure technical point of view the second was probably the best). Thus the system is viewed by the user as consisting of a Data Definition Language by which the user can declare the type structure and hierarchy. The user can attach methods to types and to objects by

\(^2\) Unix is a trademark of AT&T Bell laboratories