Open OODB: A Modular Object-Oriented DBMS

Steve Ford José A. Blakeley Thomas J. Bannon
Texas Instruments Incorporated

Open Object-Oriented Database: The Open OODB project, part of the DARPA Persistent Object Base (POB) Program, is an effort to build an open, extensible object-oriented database management system (OODB) in which database functionality can be tailored for particular applications within an incrementally improvable framework. The system is designed to serve both as a platform for research and as a testbed that can meet the needs of demanding, next generation database applications. The Open OODB project goals are to describe the design space of OODBs; build an architectural framework that enables configuring independently useful modules to form an OODB; verify the suitability of this open approach by implementing an OODB to these specifications; and identify opportunities for building consensus that can lead to much-needed OODB standards. The motivating factors in this approach were that our previous experience in object-oriented databases had convinced us that different applications have differing requirements, and that a monolithic system is unlikely to meet the needs of many demanding kinds of applications.

The Open OODB project is currently entering its fourth phase, with continued effort planned. In the first phase, requirements were gathered from a wide cross-section of application domains. In the second phase, architecture and module interface specifications were produced by TI, and reviewed by leading researchers and developers of OO systems from academia, industry, standards organizations, and government in working groups and workshops. The third phase produced an alpha release in December 1992 and will produce two subsequent releases in 1993. The fourth phase will generate detailed design specifications that can be used by external system developers to add or modify Open OODB modules to increase or alter functionality.

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 0-89791-592-5/93/0005/0552...$1.50

The Open OODB Architecture is shown in Figure 1. The system uses an Object Services Architecture which consists of a software backplane (the Meta Architecture), an extensible collection of optional "services," and support modules. The Alpha Release of the DARPA Open OODB Toolkit provides a subset of the full Open OODB functionality. It includes the following modules:

- The *C/C++ Preprocessor (ppCC)* is a generic C/C++ preprocessor that translates C++ to C++, and in the process generates sentries and translation routines; collects type information necessary for persistence and other behavioral extensions, and a data dictionary; and parses object queries.
- The *C++ Meta Architecture* provides event (sentry) management which allows behavioral extensions to objects, including persistence.
- The *Persistent C/C++ Service* allows applications to make C and C++ types and classes persistent.
- The *C/C++ Object Translation Service (OTS)* copies C/C++ objects within and between address spaces.
- The *Persistent Address Space Manager* provides robust storage of persistent objects. The current implementation is based on the Wisconsin's Exodus Storage Manager.
- The *Namespace Service* allows applications to name persistent objects.
- The *C/C++ Data Dictionary* allows C and C++ programs to access type/class information at run-time.
- The *Query Service* includes an object query language called ZQL[C++], an object query optimizer built using the Volcano Optimizer Generator, and a code generation component that can target various Persistent C++ languages.

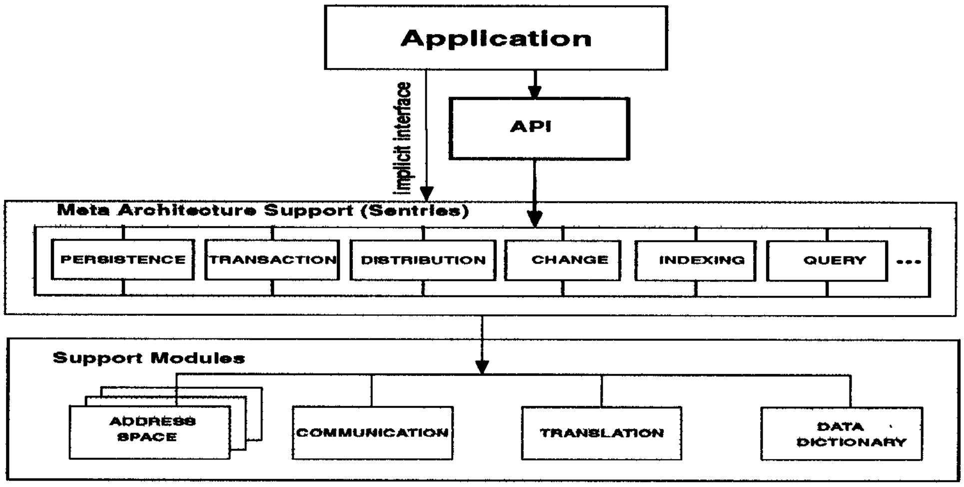

Figure 1: Open OODB Architecture.

The April Release of the DARPA Open OODB Toolkit will add Persistent[Common Lisp], versions, support for adding new object services to the framework, persistence of third party libraries (e.g., NIHCL), and other performance and portability improvements. We will present an overview of the Open OODB Architecture; describe Persistent[C++], Persistent[Common Lisp], and ZQL[C++]; and demonstrate the steps for developing an Open OODB application.

Visual Memory Prototype: *Visual Memory* is an architecture to interface future real-time image understanding systems with applications requiring access to information about 3-D objects and their environment. A prototype visual memory has been developed which uses and extends the Open OODB with the addition of several spatial and temporal indexing mechanisms to speed object retrieval. *Spatial indexing* provides fast, efficient answers to questions such as, "Is anyone in area X?" Spatial indices typically provide conservative approximate answers, allowing false positives but not false negatives, and rely on further filtering for exact results. Several different spatial indices are available, catering to different types of questions. For example, grids and point quad-trees are good for determining objects near a given point, while interval trees are better for finding object intersections. *Temporal indices* provide means for accessing object histories by efficiently determining an object's state at a given time and how objects changed over a given interval of time. They help answer questions such as "Was anyone in hall X during the night?" and "Where was object Y at 10:00am?". The prototype is implemented in C++ using a data model based on an image understanding environment class hierarchy. The laboratory demonstration uses a real-time camera monitoring task to exercise the visual memory prototype and to determine future requirements. The SIGMOD demonstration will use video taped scenes.

Our image understanding algorithms report where people are walking in the field of view of a camera. The algorithms use image motion estimation techniques and knowledge of scene geometry to estimate the positions and heights of people in the scene. These objects are stored in the visual memory. A user interface provides interactive access to visual memory data via historical and real-time queries. To make historical queries, the user specifies periods of time, regions of space, and object types, and the system retrieves the corresponding objects. To convey real-time information, locations of people are indicated on a floor plan display and changed dynamically as the visual memory is updated.

Acknowledgements: Open OODB is sponsored in part by the Defense Advanced Research Projects Agency under contract DAAB 07-90-C-B920.

The *Open OODB* is currently the work of José Blakeley, Steve Ford, Moira Mallison, Edward Perez, Aditya Srivastava, Craig Thompson, and David Wells. Many others have significantly contributed in the past. The *Visual Memory* application is the work of Tom Bannon, Bruce Flinchbaugh, Steve Ford, Chris Kellogg, and Tom O'Donnell.
Authors' address: Texas Instruments, P.O. Box 655474, MS 238, Dallas, Texas 75265. email: lastname@csc.ti.com.