

# MOSAICO - A System for Conceptual Modeling and Rapid Prototyping of Object-Oriented Database Application

M. Missikoff and M. Toiati

IASI - CNR

Viale Manzoni 30, 00185 Rome, Italy

email: {missikoff,toiati}@iasi.rm.cnr.it

## Abstract

The system Mosaico [MT94] has been conceived to support the design, conceptual modeling, and rapid prototyping of data intensive applications based on Object-Oriented Databases (OODBs). The application is modeled through a graphical user interface and the produced model is encoded in *TQL++*, the design language on which Mosaico is based.

## 1 Architecture of Mosaico System

The functions supplied by the system can be grouped into four main subsystems.

- *OODB Application Specification* - This subsystem supports the designer in the process of constructing the specification of an application, by using the specification language *TQL++* [FM93]. An application is expressed by defining a set of types and a set of associated actions. Types can be defined from scratch or imported (and possibly adjusted) from a type library. New types are inserted in the type library for future reuse. Type and action definitions form the conceptual model of the Object-Oriented application.
- *OODB Application Verification* - The formal nature of the language *TQL++* allows the semantic verification of the specification of an application. Mosaico processes the specification aiming at finding incorrect descriptions, such as incomplete, inconsistent, or unsafe assertions. To this end, the specification is translated into an internal (logical) format and then theorem proving techniques are employed.
- *Rapid Prototyping* - This subsystem is devoted to the compilation of the conceptual model and the production of executable code. The code produced implements a prototype of the application. To actually run the prototype, it is generally required to have a sample object database (ODB). To this end, Mosaico supplies a language, Lobster, for objects definition and the initial load of the database.
- *ODB Management* - Having defined the schema and having populated the database, it is possible to

perform query functions. One key feature of the system is the possibility of querying the database using the same language conceived for the Data Definition: *TQL++*. In essence, the enquiry is performed by defining new query-types. Then, the system retrieves all the objects that satisfy the query-type and constructs the answer class.

The main innovative aspects of Mosaico are represented by the Object-Oriented conceptual language *TQL++* and by the integration of advanced functionalities offered to the designer of an Object-Oriented database application, such as formal specification and rapid prototyping. The language is characterized by all the main features of Object-Oriented technology (such as typing, inheritance, object identity, message passing), but avoids all the technical details required by existing development languages.

## 2 Implementation Issues

A first prototype of Mosaico has been developed on a SUN workstation, using BIM-Prolog. Current version generates prototypes in the form of Prolog code. A module for the generation C++ code is under development.

## References

- [FM93] A. Formica, M. Missikoff, 'Integrity Constraint Representation in Object-Oriented Databases,' *Information and Knowledge Management*, Springer-Verlag, LNCS 752, pages 69-85, 1993.
- [MT94] M. Missikoff, M. Toiati, 'Safe Rapid Prototyping of Object-Oriented Database Applications,' *Proc. of IEEE Int'l Workshop on Rapid System Prototyping*, Grenoble, June 1994.