MOODS: A MODULAR, OBJECT-ORIENTED DESIGN DATABASE SYSTEM*

Ivan L. M. RICARTE†
Department of Computer Engineering and Industrial Automation
State University of Campinas — UNICAMP
Campinas, Brazil

Abstract

MOODS is a database management system that can control multiple physical databases integrated through the object oriented paradigm. In addition to enabling the representation of design data, this system can represent data associated to the design process and to integrate designers’ supporting tools in a uniform way. Hence, MOODS can support a multimedia database, with adequate features not only for engineering design but also for a broader class of non-conventional applications.

1 Introduction

Using database systems to support engineering applications has been largely justified. Engineering design applications (or CAD — Computer Aided Design — applications) involve a large amount of data, and database systems can manipulate large volumes of data with data independence (not depending on the internal data structures). Because of this feature, databases have been proposed as a mean of data integration in CAD systems.

The early proposals of database systems for CAD applications have attempted to adapt commercial systems. Since commercial databases do not support the features required for engineering applications, many extensions were proposed, mainly to the relational approach. Among the required features are: support of dynamic structures, representation of objects composed by other objects and definition of unusual data types.

Generally, the extensions proposed at this time lead to the support of Abstract Data Types (ADT). A system supporting ADT can offer more data types, defined over the basic

†Author partially supported by CNPq — Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil
data types and with specific operations for each new type. ADT includes the concept of encapsulation (hiding of internal information), solving partially the problem of unusual data types for engineering applications.

Another approach for developing engineering databases is the proposal of new data modeling tools, mainly with the use of semantic data models. These data models incorporate structures derived from Artificial Intelligence, specially from semantic networks such as the hierarchies for generalization and specialization (arcs of the type *is-a*) and aggregation (*is-part-of*). These representation structures solve the problem of complex objects and include the early concepts of inheritance through the specialization hierarchy.

Object Oriented Database Systems (OODBS) are now emerging as a new hope in the engineering databases field. These systems integrate concepts of both previous proposals, providing encapsulation (internal structures of objects are accessed only through defined methods) and specialization hierarchies (new objects may be defined from already existing ones, with the inheritance of associated methods).

The origin of object oriented systems comes from the programming languages field; some OODBS are defined through the inclusion of database features to object oriented programming languages. Another approach is to provide an object-like view to database systems (mainly relational systems). However, the best (and probably the most difficult to implement) option seems to be the definition of new OODBS from scratch. This is the best way to achieve the full power of the object orientated approach (behavioral object orientation, according to Dittrich [3]).

MOODS is a database system which aims to integrate features from object orientation and semantic data modeling. This proposal differs from many others mainly because MOODS does not intend to concentrate but to integrate all the data related to the engineering application. In the next section the MOODS environment will be described. In addition, in order to provide a behavioral object orientation, a core system for manipulating objects is also proposed. This system is UniCOSMOS, also described in this paper.

## 2 MOODS

MOODS is an acronym standing for Modular, Object Oriented Design Database System. Its main goals are to provide the facilities needed for engineering data modeling and to integrate CAD software tools with the database through the use of uniform mechanisms. The object orientated paradigm was chosen to provide a natural and uniform view to users.

Object orientation offers a good representation model for CAD applications. The design process may be viewed as a process of refinement, evolution and integration of design objects. With object orientation, these design objects may be represented directly by objects in the data model. In other approaches, this representation is not so natural.

In MOODS, everything is an object. Objects may be grouped into classes (into more
than one class, if needed), which can be viewed as single objects by classes of higher order. This concept of ordering, as many others in the MOODS and UniCOSMOS modeling approach, is derived from the Theory of Types [5].

Domains of values are also classes. Thus, objects may be described uniformly in terms of values and of other objects, which may be of the same class of the described object. This is a natural way to define complex objects, classes and domains.

The main concept of MOODS is the node object type. Each object of this type may be viewed as an isolation unit. At the physical level, these isolation units may represent distinct and isolated database units. This approach enables the definition of a segmented database, an important feature for design databases [4]. Using node objects, it is possible to associate design sub-objects with isolated databases, enabling the structuring and supporting of the design process modeling.

The name node resembles hypertext systems [1]; however, in MOODS the node concept has a broader sense. As every MOOODS object, a node object may be part of the description of other objects. Hence, node objects may be used not only as mechanisms for design organization but also as a mechanism for the integration of databases with tools specific for CAD and supporting tools (e.g., graphical systems, word processors, spread sheets). In this case, the node objects refer to units that are not other databases, but files generated by these softwares, which complement an object description.

The MOODS environment can be seen in Figure 1. As shown there, a design database is composed not only of object oriented databases (generated by UniCOSMOS core system) but also of the files generated by the supporting software. Methods associated to node objects classes define the interface between MOODS and these tools.

UniCOSMOS represents an important piece of this environment, supporting logically and physically the definition of objects in each one of the database units in MOODS. This core system will be described in the next section.
3 UniCOSMOS

UniCOSMOS (acronym for University of Campinas Object Storage Management System) is the core system which provides the object orientation support for one database. The modeling features of UniCOSMOS include most of the semantic aspects supported by MOODS.

As already mentioned in the description of MOODS, every object, class or domain has uniform treatment. In UniCOSMOS, this feature is achieved through an object identification that is independent of values (as the concept of surrogate). Although values are not stored as objects, they receive similar identifications. Thus, it is possible to treat descriptions of objects uniformly, with domain-value pairs and with class-object pairs.

Domains are defined as classes, with some of them predefined to express the basic data types (integer, float, string). From these, new classes can be defined (as in ADT).

Classes may be defined through operations over other classes. These operations may be functionally defined through definition of new methods. In the definition of new classes, these new operations or some of the implicit ones (as sum, product and inclusion of classes) may be used.

Restriction rules may be modeled through methods associated directly to a class definition or indirectly through auxiliary classes, which may be associated through a higher order class. These kinds of objects may be viewed as rule object types.

In the definition of classes that can express properties involving the whole of other classes (i.e., classes of higher order), the internal class Cls must be used as domain. This class has as members every class defined in the database.

Internally, each UniCOSMOS database is composed of three logical structures. The object area contains descriptions of objects, classes and domains. The values area stores representations associated with elementary and complex data type values. At last, the access area enables access to values, objects, classes and domains, independently of internal identifiers.

There is a physical file associated to each one of these areas. This set of files, in addition to one auxiliary file describing the areas occupation, forms a physical UniCOSMOS database. This database enables the permanent storage of objects, and it is referred to by a name which identifies the associated set of files.

4 Conclusions

MOODS is a system which offers a uniform environment integrating powerful object oriented modeling tools and supporting software. The global design database is segmented into units composed of UniCOSMOS databases and files generated by other software.

The same mechanism which supports this segmentation, node objects, offers natural ways to support other features: multi-user access (with each user working in a distinct
node); multi-process support (activation of a node as a process activation); distributed systems; and access control. Through node objects it is also possible to think of MOODS as a good environment to define multimedia databases.

It is also possible to use MOODS by limiting the existing types of classes. In this way users can have a more usual data view, as the relational view (defining a object type relation with methods corresponding to relational operations) or the entity-relationship view (as the view offered by GERPAC, an entity-relationship database system for CAD applications [2]).

The early prototypes of MOODS and UniCOSMOS are being implemented with the C language. The use of object oriented languages, such as C++, is under evaluation. In this case, the object oriented constructs will not be used to implement directly the MOODS and UniCOSMOS concepts in order to avoid language limitations.

Acknowledgments

The colleagues from the group of databases for CAD at UNICAMP, specially Léo Magalhães, gave valuable contribution. The concept of node object is an extension of schema, proposed by Armando Delgado to the GERPAC system.

The first version of this work was written at the T.H.Darmstadt (West Germany), during a period of cooperation work.

Wu Shin Ting and Mario Jino gave valuable suggestions which improved the readability of this paper.

References


