An Object Model for Engineering Design

G.T Nguyen, D. Rieu, J. Escamilla

IRIMAG
Laboratoire de Génie Informatique
IMAG-Campus
BP 53 X
38041 GRENOBLE Cedex
France
Tel : (33) 76.51.45.75
e-mail : nguyen@imag.fr
fax : (33) 76.44.66.75

Abstract. Applications requiring sophisticated modeling techniques raise challenging issues to software designers. CAD/CAM and genetics are example of applications that call for powerful modeling techniques. Existing approaches seem limited in their ability to support their demands. Relational database systems for example support only simple tables. The need to enhance their capabilities led to non-normalized relational data models. Object-oriented programming languages and databases propose new solutions to the problems of complex and composite object modeling and manipulation. Yet, severe shortcomings impede their practicability, e.g., their inability to model multiple object representations and complex semantic relationships.

This paper is an informal overview of a data model called SHOOD based on object-oriented concepts and frame-based knowledge representation. SHOOD implements sophisticated features, such as: • object persistence, multi-methods along a specific specialization hierarchy (which is independent of the class hierarchy), • sophisticated semantic relationships, e.g., dependency relationships between objects (which are totally independent of the composition relationship), • multiple object representations, allowing the users to manipulate the objects from several points of views simultaneously, • the systematic use of a powerful meta-object kernel, which is used to implement a reflexive architecture. The paper focuses on the last two issues.

1 Introduction

SHOOD is an object-oriented data model designed to support highly dynamic applications. Design, genetics and weather forecast are examples of such applications. The challenge there is to merge partial amounts of information into a consistent and flexible computerized model.

Yet, SHOOD is not just another exotic data model. It includes features that are seldom found simultaneously in existing data models. Some of its salient characteristics are:
• the support for evolving data, both in their definitions and values,
• the support for flexible user defined semantic relationships between objects, e.g., composition relationships, specific dependency relationships (existential, exclusive, etc.),
the support for multiple object representations, which can be concurrently defined by different users on the same objects, e.g., hydraulic and electric points of view for an engine,

- object persistence, although SHOOD does not currently provide all the functionalities of a complete database management system, e.g., concurrency control and restart facilities,

- multiple inferencing capabilities, allowing for several alternatives to be defined for attribute computations, e.g., depending on the arguments' availability.

From a historical perspective, SHOOD was designed to support the requirements of mechanical and VLSI CAD/CAM applications [22, 28]. It elaborates on knowledge representation techniques found in AI and data models found in the database area [19]. It also imports recent advances in object-oriented databases and programming languages [10, 15]. It includes such notions as meta-classes, methods, inferences, and dependencies. They are merged in a powerful and flexible data model that is currently tested on full size applications in mechanical engineering design. SHOOD is implemented in Le_Lisp™ on Sun SPARCstation 2™. A user-friendly interface is built on top of the system to provide an easy access to its functionalities. It is based on a menu-driven windowing system implemented in Aida™, a graphic interface development tool running under X-Window.

The paper is organized as follows. Section 2 is an in-depth analysis of the design requirements for SHOOD. Section 3 describes the meta-object kernel. It includes the meta-classes that are used to implement the concepts in the model. Section 4 is an overview of the multiple representations supported for the user objects, with indications on the classification mechanism available. Section 5 discusses research issues. Section 6 is a conclusion.

2 Rationale for the design of SHOOD

2.1 Extensibility

Among the issues raised by design applications, the evolution of objects values and structure - grossly speaking database evolution - is probably the most challenging. It questions one of the most securing aspects of data storage and manipulation, i.e., object stability. Because design applications are constructive, they require the definitions of the objects to evolve over time. This departs dramatically from business applications that manipulate large amounts of information with somewhat few data definitions. Design in contrast manipulate smaller amounts of data because the goal is to define artifacts rather than query the database.

The need for flexible data models is therefore fundamental. The commonly accepted relational model of data is for this matter all but flexible. User relations flatten object attributes that are distributed in multiple pieces among relations without explicit semantic relationships.

Recent advances in object-oriented data modeling have opened new perspectives for the definition and manipulation of complex composite objects. Besides their ability to encapsulate and reuse existing definitions - that are of first importance in design - they benefit from research in the field of object-oriented database systems [30, 31]. Yet, severe shortcomings limit their use in design applications. First there is a crucial lack of design methodologies for object-oriented systems' applications. Also, they suffer from inherent limitations in their ability to model user defined semantic relationships [8, 11]. Specific dependency relationships need to be defined between parts of a composite object, e.g., there is an existential dependency between the airframe and an aircraft built around it,