

# Object Oriented Database Schema Design

B. B. Meshram  
Dept.of Computer Technology.  
V. J. T. I. Matunga,  
Mumbai –400019  
Maharashtra State(INDIA)  
bbmeshram@roltanet.com

T. R. Sontakke  
S.G.G.S. C. E & T  
Vishnupuri.,  
Nanded –402 602  
Maharashtra State(INDIA)  
trsontakke@yahoo.com

## Abstract

To provide a rich variety of modeling constructs, which can model most of the situations that may arise in advance database application domains, there is a need of systematic integration of analysis design methods. To take the goodies of famous methods integrating OMT, Booch, Ivar Jacobson & Shlaer Mellor is a challenging universal truth of requirements engineering. Consequently, we attempt to borrow useful ideas and notations from these methods for Object Oriented Analysis and design. We have exploited this for object oriented database schema design. For this we have drawn an Extended Entity Relationship Diagram Information Table (EERDIT) to overcome the disadvantages of Entity Relationship Diagram and mapped it to Class Composition Diagram (CCD). We have mapped CCD to object schema using an appropriate mapping algorithm; then we have augmented the OO schema with the necessary behavioral construction.

**Keywords:** object oriented analysis and design (OOA& D), OODBMS, EERDIT, CCD, IOOAM, Use Cases.

## 1 Introduction

Relational Database Management systems (RDBMS) have been very successful, but their success is limited to certain types of applications. As business users expand to newer types of applications and grow older ones, their attempts to use RDBMS encounter the “Relational Wall”, where RDBMS technology no longer provides the performance and functionality needed.

Despite the large number of methods that has been introduced “no silver bullet” has been found. Each method has its particular strengths and weaknesses and is more suitable for certain problem areas than for others. Formal methods suffer from a large gap between the models and reality as remarked in [10]. How can we bring together and extend the best of this work for use in designing object oriented database systems? Addressing this topic is an arduous task, and we do not pretend to have all of the answers, but we succinctly provide our contribution in the form of some specific model transformations and algorithms/steps for object oriented database schema design and provided systematic strategy to include method signature generation and in the form of several specific adjustments based on insights about application characteristics and theoretical anomalies. We proceed with the development of our approach as. In section2, we examine the models that are fundamental to an object approach to system analysis: the models that describe the relations between objects (the data model), the way objects cooperate (the interaction

model) and the way objects behave (the behavior model). Certain models like OMT, Booch, Ivar Jacobson & Shlaer Mellor are integrated together for object oriented analysis. In section 3, we give our basic object oriented database design steps by integrating the OO analysis and design models.

## **2 THE INTEGRATED OBJECT ORIENTED ANALYSIS MODEL( IOOAM )**

The limitations of the existing record oriented data models have long been observed [2,37]. To alleviate these limitations, several object oriented and semantic data models have been introduced as the potential alternatives for modeling many advanced database applications to capture much more of the semantics of these application domains in “natural” way; but to provide a rich variety of modeling constructs, which can model most of the situations that may arise in such application domains, there is a need of systematic integration of analysis design methods. To take the goodies of famous methods OMT, Booch, Ivar Jacobson & Shlaer Mellor is a challenging universal truth of requirements engineering because we often find that the notation of one method is not rich enough to express semantic concepts, consequently, we attempt to borrow useful ideas and notations from other methods. The Booch OOA method reference: class diagram, object diagrams, state event diagrams and interaction diagrams[16]. The OMT method reference: object modeling, function modeling and dynamic modeling [24]. Ivar Jacobson's Object Oriented Software Engineering approach is a use case driven approach [21] Shlaer Mellor OOA models are made up of three separate, but integrated parts: Object Information model, state models & action specification (Action dataflow diagrams or a well designed action language)[40]. This is only a overview ! We recommend to read [16,21,24,40].

### **2.1 Entity-Relationship Diagram**

ER diagrams (ERDs) were introduced by Chenn(1973) to represent the conceptual structure of a data in a database system; but following are the drawbacks that we have observed in the Extended ER Diagrams ( EERD's)

- all information about entity and its attributes cannot be detailed in the EER diagram i.e. whether the entity is structured or unstructured image.
- A relationship mode in an EERD's can represent a communication link, but it may also represent a visibility link, a permission, a part of link and so on.
- >From the start of their introduction by Chenn, EERD's have been ambiguous, representing components in the environments of the software product or representing
- conceptual software components. . This ambiguity arises from the fact that in database systems, where EERD's are mainly used the conceptual structure of the data must correspond to the conceptual structure of the part of the world represented by the data base.
- In the control intensive systems, an EERD can represent the part of the world controlled by the system or it can represent conceptual object inside the system. One cannot tell by looking at the EERD whether the particular entity represents real entity in the environment or software objects in the control system.