

An re-engineering Approach for Object-Oriented Database Design

J. Fong^a and I. Kwan^b

^aDepartment of Computer Science, City Polytechnic of Hong Kong, Kowloon, Hong Kong

^bDepartment of Computing, Hong Kong Polytechnic, Kowloon, Hong Kong

Abstract

Applications with databases built by EER model could seek to convert their databases into its OO model for capturing more semantics of application domain. To capture its dynamic semantic, we need users as domain expert for the newly developed database, as they are familiar with the application and can provide domain and relation integrity from the designed database. The objective of this paper is to present an OODB design methodology, which can be used as re-engineering procedure to upgrade database system to OODB through an Object Modelling Techniques(OMT).

Keyword Codes: D.2.10, H.2.1

Keywords: Extended Entity Relationship, Object-Oriented Database, Object Modelling Technique

1. INTRODUCTION

The development of recent object-oriented database(OODB) technology has emphasized on both static and dynamic structure and semantics. The object model to be used in this paper is based on the OMT methodology. The OMT supports the entire software development cycle by the use of its three models: Object model, Dynamic model and Functional model. This three-part model of the system is then refined and optimized to constitute a design. We believe re-engineering a database by taking advantage of new technology, in particular, OODB technology, while maintaining all the features that were proven to be useful in the original database. The translation from an EER-based database to an OODB model can contribute to the development of OODB design methodology.

2. THE HEURISTIC CONVERSION RULES FROM EER MODEL TO OODB MODEL

We will adopt the syntaxes of OODB model by John Hughes[1] using Class, Properties and Operations, instead of Class, Attributes and Methods, and OMT model by James Rumbaugh[2] in this paper.

2.1 Part I- Map EER to OODB in static semantic transformation

Step 1 Entity → class[3]

An EER Model works with entity types and their corresponding attributes. Attributes of a particular entity may be considered as instance variable of the class instance. For example, an entity type Book can be mapped into a class Book as shown in Figure 1. The followings are the mapped OODB with class BOOK where properties are attributes and operations are methods:

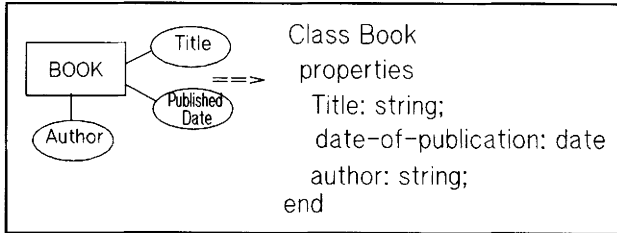


Figure 1 Map an entity to a class

Step 2 Relationship → Association[4]

In EER Model, relationships are represented as named associations among entities. In object-oriented schema, they are links and pointers between superclasses and subclasses. The relationship in the EER model can be mapped into an association in object-oriented schema on a 1:1 basis with its corresponding multiplicity of links/pointers. When constrained by cardinality, relationship need to be mapped according to the constraint. For example, the 1:n relationship in Figure 2 can be mapped into the following OODB where "has" is an association attribute of class Department associating with a set of another class Employee to comply with the 1:n relationship between entity Department and entity Employee in the EER model, and its "hired-by" is an inverse association.

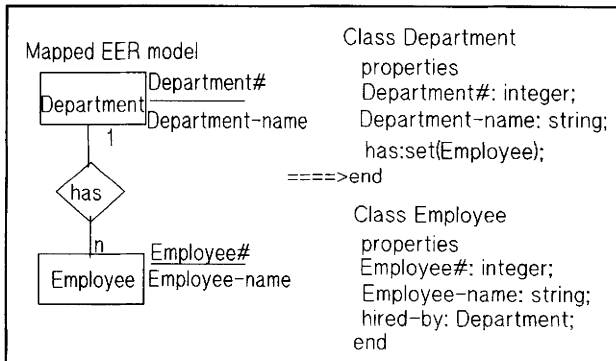


Figure 2 Map relationship to OODB

Step 3 Generalisation → Method[5]

For generalization, the variances among entities are suppressed and their commonalities are identified by generalizing them into one single class. The original entities with each of its unique differences are special subclasses. The mutually exclusive subclasses are called disjoint generalisation. The mutually inclusive subclasses are called overlap generalisation. For example, disjoint generalization in Figure 3 can be mapped into the following OODB where subclasses Contract-Staff and Permanent-Staff inherit the properties and operations of superclass Staff.

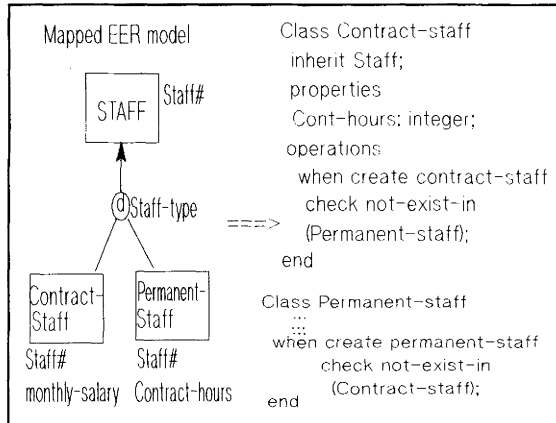


Figure 3 Map generalization to OODB

The mapping of overlap generalization into the object-oriented schema is similar to the mapping of disjoint generalization into the object-oriented schema except that check statement is omitted.

Step 4 Categorisation → Method

A categorisation is derived by mapping isa relationships and their record types to superclass/subclass such that a set of superclass can be united together to form a superclass to a subclass. All these superclasses may have different key attribute as they are originally independent classes. For example, the categorization in Figure 4 can be mapped into the following OODB where subclass Car-owner inherits by associating with only one of the three superclasses: Bank, Person or Company.

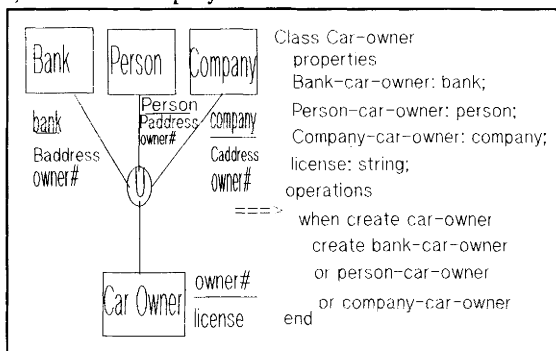


Figure 4 Map categorization to OODB

Step 5 Isa → Inheritance[6]

The concept of inheritance associated with generalization/specification (isa) relationship in object-oriented schema permits classes to be organized in which specialized classes inherit the properties and operations of more generalized class. Class carries common properties while deriving specialized subclass. For example, the isa relationship in Figure 5 can be mapped into the following OODB where subclass Car inherits the properties of its superclass Vehicle.

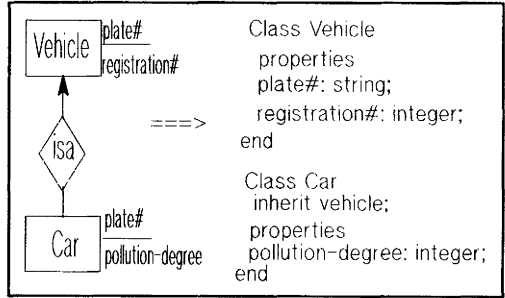


Figure 5 Map isa relationship to OODB

Step 6 Weak entity → Component class

The existence of weak entity in EER model depends on its owner entity. For example, the weak entity Dependent in Figure 6 can be mapped into the following OODB where class Employee is a composite object class which owns a component class Dependent. The own statement implies an existence dependence of component class Dependent such that if an instance of class Employee is deleted, its corresponding component class Dependent instances are also deleted.

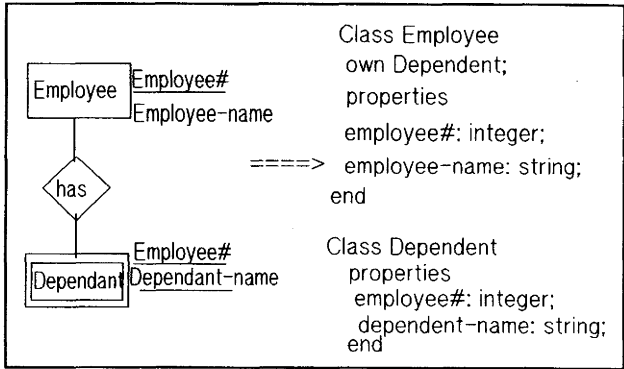


Figure 6 Map categorization to OODB

Step 7 Aggregation → Composite object

The entities and their relationship in EER model can be aggregated to form an entity. In object-oriented model, it permits the combination of classes that are related into a higher level of a composite class. For example, the aggregation in Figure 7 can be mapped into the object-oriented schema in the followings where class Section is an aggregation with two component classes: class Instructor and class Course.

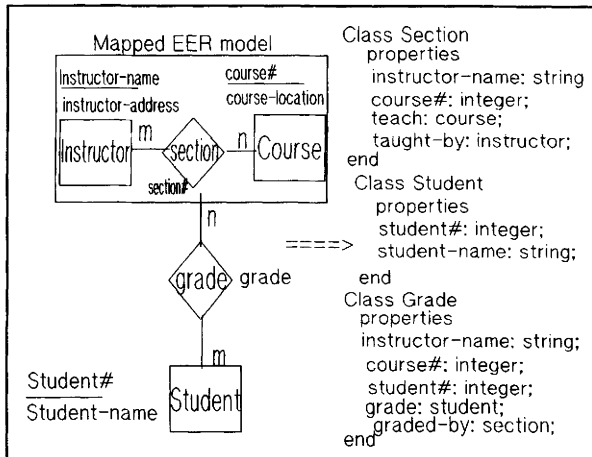


Figure 7 Map aggregation to OODB

2.2 Part II- Knowledge acquisition to obtain the dynamic behaviour semantic of data

Logical schema express data structure with little semantics. To translate an EER schema to OO schema, a direct translation can only map the data structure and relationships but not the extra semantic not represented in original EER and the dynamic semantic recovery. To recapture these dynamic semantics and extra semantic, a knowledge-based approach is therefore useful as users views relies mainly on judgement instead of mechanistic algorithms. The Expert system could then fill in the semantics needed in schema translation by leading users to make knowledgeable decision whenever semantics judgment is required[7].

Object-oriented database development is primarily a data abstraction technique. The extra semantic of the object-oriented data model as compared with EER data model are :Schema Evolution and Object Behaviour

The dynamic data modelling which is not catered in EER model would need to be recovered via a knowledge-based approach to elicit the control information of EER model and represent them in OODB model.

The OMT dynamic model which consists of state diagrams and global event flow diagram, specifies allowable sequences of changes to objects from the static object model. Thus, it forms a good representation of changes towards objects and their relationships over time. A set of steps with questions could be asked by the knowledge-based system upon the completion of schema translation process after each schema's properties and relationships in the system have been formed and the resultant object model is created.

These may also be considered and derived in the mapping procedure. This could be best accomplished with the translation rules PLUS users assistance with a knowledge-based approach.

Step 8 : Draw an intermediate OMT Object model(optional)

Operations and state properties are not defined at this stage. They are integrated into the refined Object model after step 8, step 9 and step 10 are processed.

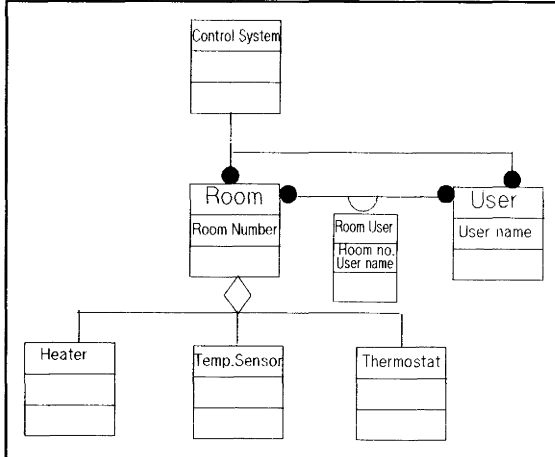


Figure 8 An intermediate corresponding OMT model of a heating system

Step 9 : Derive Dynamic Data Semantic

Q. 1 In light of providing overall picture of the system, what are the expected 'Normal' behaviour in the system?

Answer: A Heater System Scenario

Set a Heater desired temperature by Thermostat, sense temperature by Temp.sensor
 If temperature > desired temperature, switch heater off (event 1)
 If temperature < desired temperature, switch heater on (event 2)
 If temperature = desired temperature, no action (event 3)

Q. 2 What are the causing and resulting event of change of state in each object class ?
 (for example : heater)

Answer: On -> Temp>Desired Temp -> Off -> Temp<Desired Temp -> On
 event-1 event-2

Otherwise,
 no change of state value when Temp=Desired Temperature (event-3)

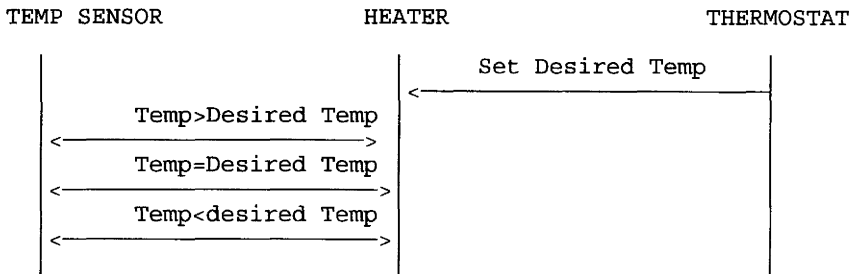


Figure 9 Event-trace diagram for HEATER active object between its related active objects TEMP SENSOR and THERMOSTAT

Q. 3 What is the typical changing sequence for Heater object ?

(ask one object class at a time, for example: heater)

Answer: On->Off->On subject to event that stimuli the object state.

Q. 4 What are the valid state of each object class ?

(ask one object class after the other, for example: heater)

Answer: On or Off, 2 state values only.

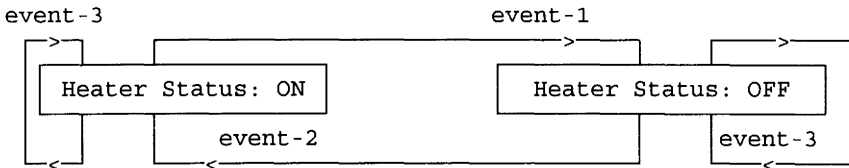


Figure 10 A state diagram of heater

Step 10: Check for consistency and completeness of events between objects and shared among the state diagrams

Q. 5 Are there any state defined in an object class not invoked by a defined event of any object in the event flow diagram ?

Answer: No, (for example, heater object) On -> Temp>Desired Temperature -> Off -> Temp<Desired Temperature -> On

Q. 6 Are there any event causing or resulting a state of a class not being defined in its corresponding state diagram ?

Answer: No, (for example, heater object) On -> Temp>Desired Temperature -> Off -> Temp<Desired Temperature -> On

Action: Put correspond state variable and operation event in each of the related active object class in the object model as in figure 14.

Step 11 : Check consistency between object model and dynamic model

Q. 7 For each state diagram in the dynamic model, is there a corresponding state attribute definition in each active object class in the object model ?

Answer: Yes, Heater status in Heater, Temperature-reading in Temp Sensor and Required Temperature in Thermostat.

Q. 8 For each event causing or resulting the change of state value of an object class, is there a corresponding operation definition in the object class in the object model ?

Answer: Yes, Switch On/Off in Heater, Read Temperature in Temp Sensor and Set Required Temperature in Thermostat.

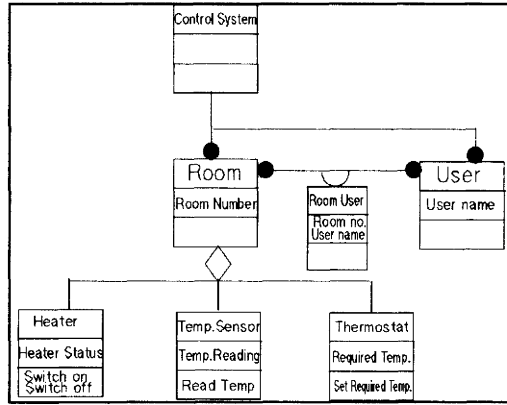


Figure 11 Final OMT model

3. CONCLUSION

This paper proposes a methodology to transform different semantic represented in EER model to OODB schema.

The proposed schema translation procedure from EER to OODB with capturing capabilities of both static data modelling and dynamic data modelling could be semi-automated, tested and verified with a sophisticated case study. The refined methodology, procedures, system/user dialogue and graphical user interface could be automated by implementing it with an appropriate Expert system shell and software, which would certainly open enhancement. The future enhancement of this research is to investigate the knowledge-based approach of capturing dynamic behaviour data semantics and to build CASE tools functionality for OODB design[8].

REFERENCES

1. Hughes J. 'Object-Oriented Databases', *Prentice Hall*, 1991, pp102-106.
2. Rumbaugh J. et al. 'Object-Oriented Modelling and Design', *Prentice Hall*, 1991, pp183-185.
3. Getta J. 'Translation of Extended Entity-Relationship Database Model into Object-oriented Database', *Interoperable Database Systems(DS-5)*, 1993, pp87-100.
4. Il-Yeol Song 'A Survey of Object-Oriented Database Design Methodologies', *Proceedings of International Conference on Information and Knowledge Management'92*, 1992, pp52-59
5. Chan, J. 'Comparison between OODBMS and RDBMS: A Case Study Approach', *M.Sc Project report of Computer Science, Department at City Polytechnic of Hong Kong* 1994
6. Narasimhan B., Navathe S., Jayaraman S. 'On Mapping ER and Relational into OO Schemas' *Proceedings of the 12th International Conference on Entity-Relationship Approach*, 1993, pp397-692.
7. Fong J., Ho M. 'Knowledge-Based Approach For Abstracting Hierarchical and Network Schema Semantics', *Proceedings of the 12th International Conference on Entity-Relationship Approach*, 1993, pp498-509.
8. Fong J., Kwan I. 'Application of CASE Tool in Database Design By Use Of Semantic Association Model' *Proceedings of the SCS Silver Jubilee Conference on Software Engineering : New Technologies & Business Payoffs*, 1992, pp277-290.