Mapping Extended Entity Relationship Model to Object Modeling Technique

Joseph Fong
Department of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
email: csjfong@cityu.edu.hk

Abstract

A methodology of reengineering existing extended Entity-Relationship(EER) model to Object Modeling Technique(OMT) model is described. A set of translation rules from EER model to a generic Object-Oriented(OO) model of OMT methodology is devised. Such reengineering practices not only can provide us with significant insight to the "interoperability" between the OO and the traditional semantic modelling techniques, but also can lead us to the development of a practical design methodology for object-oriented databases(OODB).

Keyword: EER model, OMT model, reengineering, interoperability

1 Introduction

In the current software area, there is a paradigm shift to Object orientation. OO modelling is a new way of representing static and dynamic data semantics in the form of objects, links/associations, and methods[1]. Traditional record-based relational databases built by using top-down modelling techniques such as the EER model have been generally used over the past two decades[2]. Applications with such record-based databases could seek reengineering their databases into OO ones in order to capture more semantics of the application domain. The OMT model[3] can be regarded as an extension of the ER model with complete OO features, a comprehensive OODB model enhanced with advanced semantic features. The EER model can be mapped to an OMT model as a representative OODB model in forward engineering. OMT model improves EER model in the areas of expressiveness and readability. However, from the user point of view, EER model is much easier to use than OMT model because of its simplicity. It seems reasonable to follow the traditional method to design a database starting with EER model for its richness in static semantic data modelling technique, and then map it to an OMT model as part of OODB design.

2 Mapping from EER model to OMT model

The process of mapping from EER model to OMT model can lead us into transformation from traditional semantic models into OO model, and can be described in the following steps:

Step 1 Entity → class[4]

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 of OMT model as shown in Figure 1.

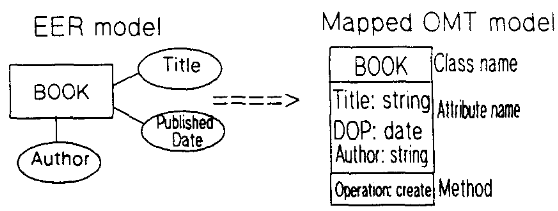

Figure 1 Map an entity to a class

Step 2 Relationship → Association[5]

In EER Model, relationships are represented as named associations among entities. In OO schema, they are links and associations between superclass(es) and subclass(es). Link is a physical conceptual connection between object instances. Association describes a group of link with common structure and semantics and pointer as an attribute in an object that combines an explicit reference to another object. The relationship in the EER model can be mapped into an association in OO schema on a 1:1 basis with its corresponding multiplicity of links and pointers. When constrained by cardinality, relationship need to be mapped according to the constraint. For example, the 1:n relationship in Figure 7 can be mapped into the OMT model where "cour-prer" is an association attribute of class Course associating with a set of another object instance of Prerequisite to comply with the 1:n relationship between entity Course and entity Prerequisite in the EER model, and its "prerequisite-of" is an inverse association class in Prerequisite. Another example of m:n relationship mapped to a link attribute can be shown in Figure 9 where "grade" is a link attribute in association Enrol. In general, a n-ary relationship in EER model can be mapped to a n-ary association in OMT model. Also the minimum and maximum occurrence in EER model can be mapped to the multiplicity of association in OMT model. Similarly, an optional relationship can be mapped to an optional association in OMT model.

Step 3 Generalisation → Method[6]

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 subclass(es). The mutually exclusive subclass(es) are called disjoint generalisation. The mutually inclusive subclasses are called overlap generalisation. For example, disjoint generalization in Figure 2 can be mapped into the OMT model where subclass(es) Contract-Staff and Permanent-Staff inherit the properties and operations of superclass Staff. The mapping of overlap generalization into the OO schema is similar to the mapping of disjoint generalization into the OO schema except that check statement is omitted and a solid triangle is used to indicate overlapped subclass(es).

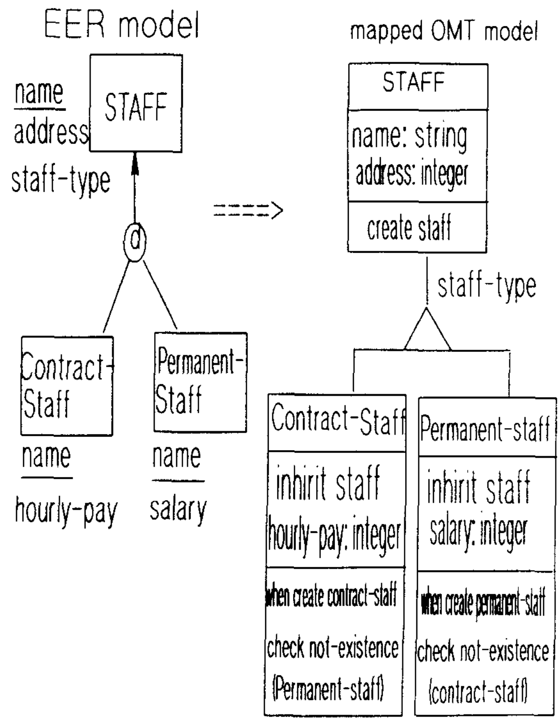

Figure 2 Map generalization to OMT

Step 4 Categorisation → "Multiple" inheritance[3][7]

A categorisation is derived by mapping isa relationships and their record types to superclass/subclass such that a set of superclass(es) can be united together to form a superclass to a subclass. All these superclass(es) may have different key attribute as they are originally independent classes. For example, the categorization in Figure 3 can be mapped into the following OMT model where subclass Car-owner inherits by associating with only one of the three superclass(es): Bank, Person or Company.

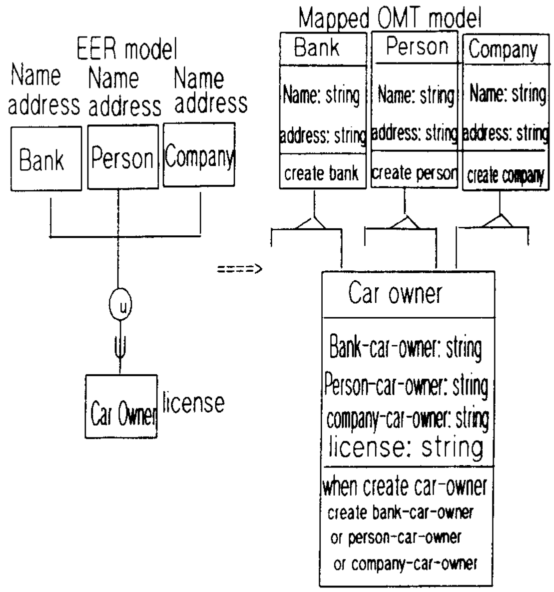

Figure 3 Map categorization to OMT model

Step 5 Isa → Inheritance[2]

The concept of inheritance associated with generalization (isa) relationship in OO schema permits classes to be organized in which specialized class(es) inherit the properties and operations of more generalized class. Class carries common properties while deriving specialized subclass. For example, the isa relationship in Figure 4 can be mapped into the following OMT model where subclass Car inherits the properties of its superclass Vehicle.

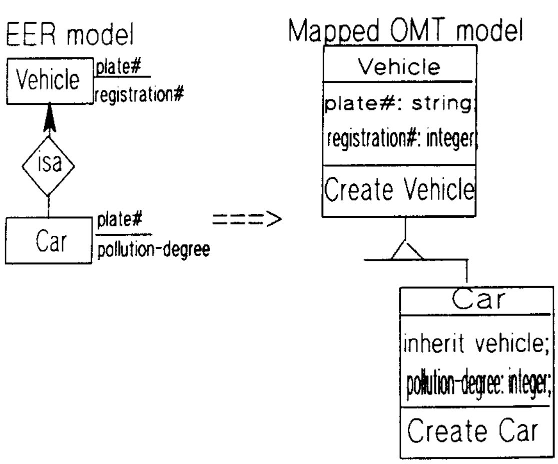

Figure 4 Map isa relationship to OODB

Step 6 Weak entity → Component class[2]

The existence of weak entity in EER model depends on its owner entity. For example, the weak entity Instructor in Figure 8 can be mapped into the following OMT model where class Department is a composite object class which owns a component class Instructor. The own statement implies an existence dependence of component class Instructor such that if an instance of class Department is deleted, its corresponding component class Instructor instances are also deleted.

Step 7 Aggregation → Composite object[8]

The entities and their relationship in EER model can be aggregated to form an entity. In OO model, it permits the combination of class(es) that are related into a higher level of a composite class. For example, the aggregation in Figure 9 can be mapped into the OO schema where class Section is an aggregation with two component classes: class Instructor and class Course.

3 Case study of mapping an EER model to an OMT model

Figure 5 is an EER model for an application Enrolment, which will be mapped to OMT model.

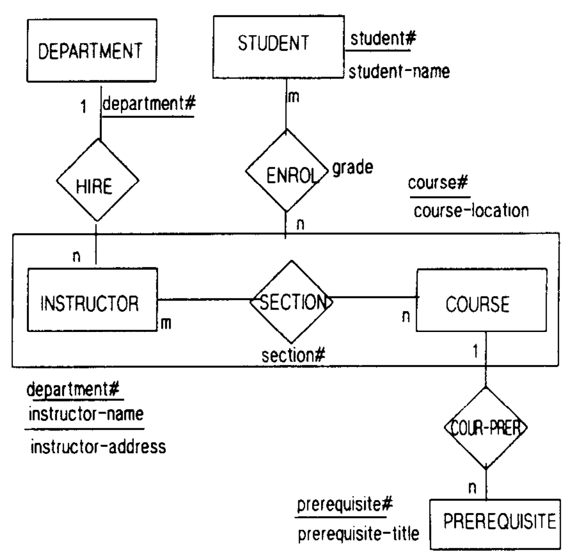

Figure 5 An EER model for enrolment

Step 1 Entity → class

Each entity in the EER model will be mapped to a class in the OMT model as shown in Figure 6.

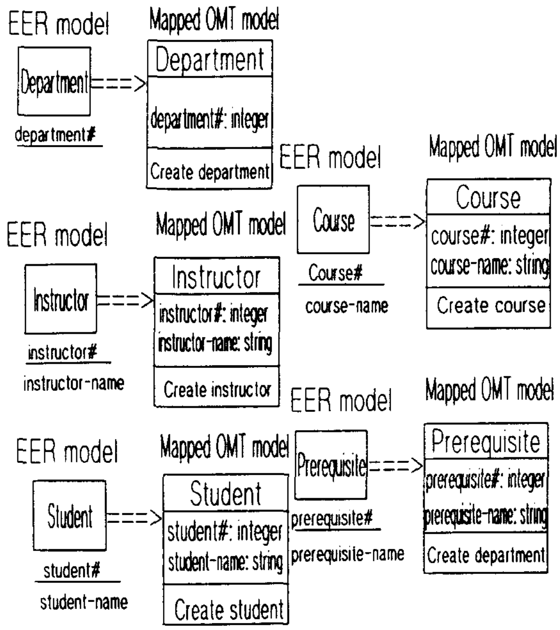

Figure 6 Map each entity to a class in OMT

Step 2 Relationship → Association

Map each cardinality to an association in OMT, with the exception of a weak entity (which will be mapped in step 6) as shown in Figure 7.

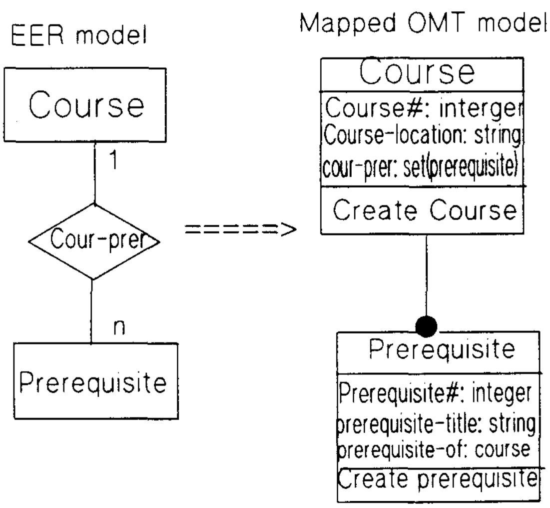

Figure 7 Map an relationship to an association in OMT model

Step 3, 4 and 5 are not applied.

Step 6 Weak entity → Component class

Map each weak entity to a component class such that there are existence dependency between two classes as shown in Figure 8.

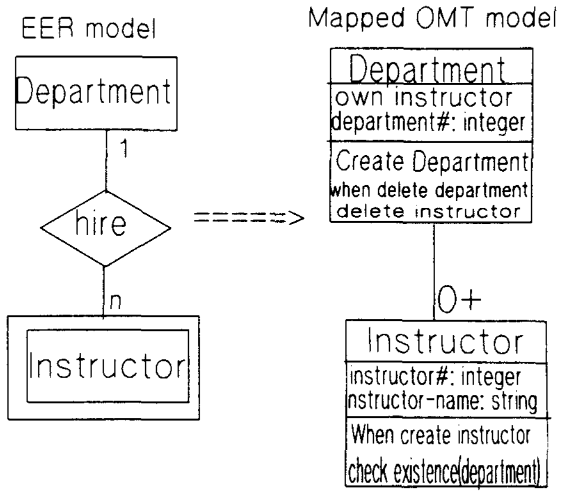

Figure 8 Map an weak entity to a dependent class in OMT model

Step 7 Aggregation → Composite object

Each aggregation in the EER model can be mapped to an aggregation (or composite object) in the OMT model as shown in Figure 9.

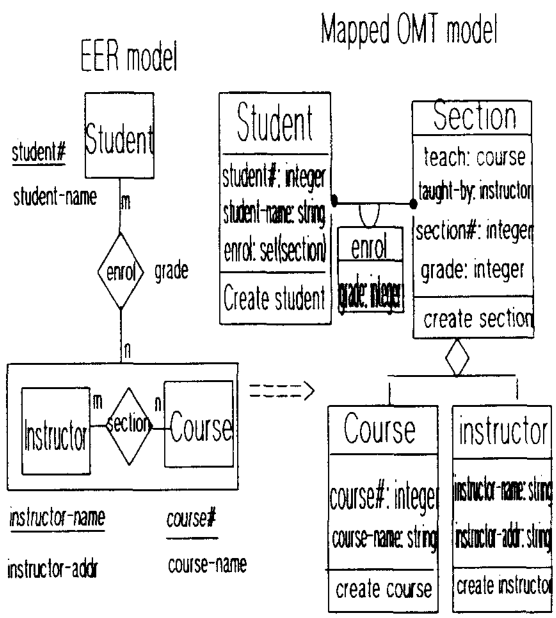

Figure 9 Map aggregation from EER to OMT

Step 8 Draw OMT model

Draw a mapped OMT in Figure 10 as a result of the previous steps.

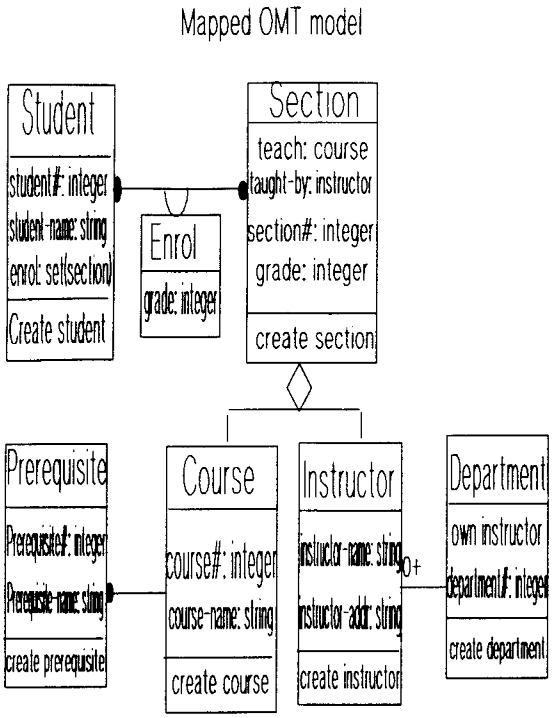

Figure 10 A mapped OMT model for the enrolment

4 Conclusion

This paper proposes a methodology to transform an EER model to an OMT model for the purpose of OODB design and also for reengineering traditional non-OODB into OODB. A schema translation procedure with steps and mapping rules that consider the vital features of the EER models and the fundamental data abstraction concepts of OO data models are proposed. The steps and its corresponding mapping rules have been illustrated with simple examples and a case study. The next logical enhancement of this research is to capture the dynamic behavior of the OODB as an enhancement of this methodology.

Reference

[1] Hughes, J. "Object-Oriented Database" *Prentice Hall* 1991, pp102-106.

[2] Elmasri, R. and Navathe, S. "Fundamentals of Database Systems" *Benjamin/Cummings* 1989 pp410-427.

[3] Rumbaugh J. et al. "Object-Oriented Modelling and Design" *Prentice Hall* 1991, pp183-185.

[4] Narasimhan, B., Navathe, S. and Jayaraman, S. "On Mapping ER and Relational Models into OO Schemas" *Proceedings of 12th International Conference on the Entity-Relationship Approach (published by Springer-Verlag)* 1993, pp402-413.

[5] Chan, J. "Comparison between OODBMS and RDBMS: A Case Study Approach" *M. Sc. Dissertation of Computer Science at City University of Hong Kong* 1994.

[6] Fong J. and Kwan I. "A Re-engineering Approach for Object-Oriented Database Design" *Proceedings of First IFIP/SQI International Conference on Software Quality and Productivity (published by Chapman and Hall)* 1994, pp139-147.

[7] Batini, C., Ceri, S. and Navathe, S. "Conceptual Database Design" An Entity Relationship Approach" *Benjamin/Cummings* 1992.

[8] Smith, J. and Smith, D. "Database Abstraction: Aggregation and Generalization" *ACM Transactions on Database Systems* Vol 2, No 2 June 1977 pp105-133.