Chapter IV

Mapping Entity Relationship Diagrams to Class Diagrams

This chapter first explains why it might be preferred to first create an entity relationship diagram (ERD) and then map it to a class diagram. The chapter then describes the mapping rules, demonstrating the mapping process with several comprehensive examples.

Why Map an ERD to a Class Diagram?

We have already seen that there is a great deal of resemblance between the objects and entity relationship (ER) models. The main difference between the two is that the ER model does not deal with functionality of the system. But there are some other differences. One of the main differences is that in the objects model there are only unary and binary relationships, while in ER there are also ternary relationships. This means that a ternary relationship between objects in reality, which can be represented as such in an ERD, is represented in a class diagram as a (separate) class. Another difference is that in ER there may be weak entity types, signified by a special symbol (a dotted rectangle for the weak entity type and dotted diamonds and connection lines for the relationships to the respective “strong” entity types). In the objects model there are no “weak” classes, but there may be classes whose key includes reference attributes to other classes. Of course, there are some differences in notations; the most visible is that in the ERD a relationship is signified by a diamond, while in the class
In almost any other sense, the models are very similar.

In the previous chapter we have learned how to create a class diagram based on the users’ needs. However, there is an alternative way—to first create an ERD (based on users’ needs) and then map it into an equivalent class diagram. There are several reasons to pursue this course of action: Some analysts may prefer working with an ER model, either because of having more experience with ER or due to personal preference. Moreover, there is research which shows that an ERD is in some cases more comprehensible by users (Shoval & Frumermann, 1994), and that analysts create more correct data models when using ERDs rather than class diagrams (Shoval & Shiran, 1997). The main reason for these two phenomena is the advantage of ER in dealing with ternary relationships: The ER model is capable of representing such relationships “directly,” using the diamond symbol, while the objects model represents such relationships as “relationship classes.” Because of this it is sometimes difficult to understand whether a class represents a “simple” thing or a complex one. Indeed, a “relationship class” can be identified by looking at the key: It consists of two or three reference attributes (depending on the specific type of ternary relationship which it stands for), but this form of identification is not clear enough compared to the “direct” representation of a ternary relationship in ERD.²

Whether we agree with these claims or not, it is important to remember that there may be users who prefer ERDs or understand them better than class diagrams, and that there may be analysts/designers who prefer ERDs or produce more correct schemas when using them rather than class diagrams. Hence, given an ERD, we must provide a precise method for mapping it to an equivalent class diagram. Equivalent means that they bear the same meaning (semantics). Once the target class diagram is created, it is of course possible to add the functional dimension (i.e., to define the functions and attach them to the proper classes).

**The Mapping Rules**

We present the mapping rules according to the following categories: (1) mapping of simple entity types and their attributes; (2) mapping of relationships between simple entity types; (3) mapping of weak entity types; and (4) mapping of structural relationships.
Related Content

Data Dictionary
www.igi-global.com/chapter/data-dictionary/18739?camid=4v1a

MDA with xUML: Model Construction and Process Management
www.igi-global.com/chapter/MDA-xUML-model-construction-process/25631?camid=4v1a

Design of the Man-Machine Interface: Menus, Inputs, and Outputs
www.igi-global.com/chapter/design-man-machine-interface/18741?camid=4v1a

Agile Project Controlling and Effort Estimation
www.igi-global.com/chapter/agile-project-controlling-effort-estimation/25642?camid=4v1a