Domain Model using UML

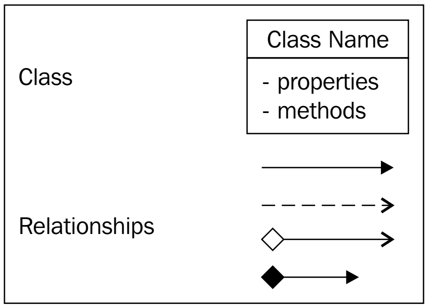
The domain model is a more object-oriented way of indicating the relationships between different objects in the context of the business logic of the application. It is similar to the ER diagram. But instead of merely showing the relationships between the entities involved, it graphically reflects how these entities relate to each other in an object-oriented fashion. On the other hand, an ER diagram is only focused from a relational perspective.

Unified Modeling Language, or UML in short, is a graphical language used to describe object-oriented designs for software systems. UML is quite a vast language, but we will focus more on class diagrams and UML relationships to represent our domain model. Class diagrams are widely used in every object-oriented system to describe the different types of internal relationships between the different business entities.

Before going for a 3-layer object-oriented system, we need to create a domain model of the system. So we need to put all of the business code into separate logical structures and start creating a domain model, in order to understand the different business entities involved.

For this, we need to "organize" the code by breaking it down into logical entities, which we call objects, and create relationships between them. The resulting set of objects with relationships defined between them would be known as the domain model of the application. It is so called because this model illustrates how different entities in the application domain would interact with each other.

We use the following shapes in our class diagram:

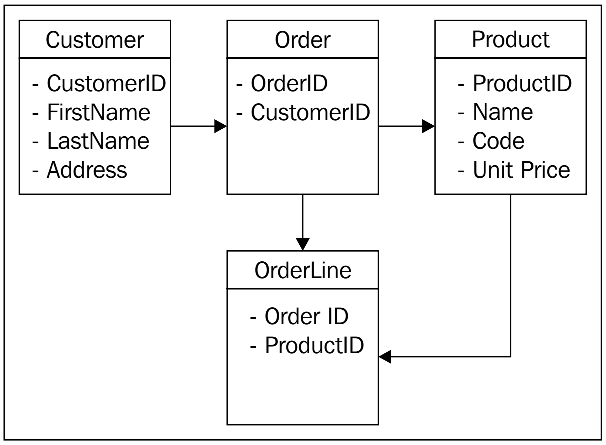


We will learn in detail what each figure represents, and how to create a domain model using them.

Class Diagram

A class diagram simply represents how different entities are related to each other in an object-oriented system. Class diagrams are different from ER diagrams because class diagrams deal with relationships in an object-oriented manner, showing inheritance, interfaces and so on, whereas an ER diagram can only depict relational models (for Relational Database Management Systems, or RDBMSs).

In order to create a class diagram for our OMS, let us highlight the major entities in our OMS in terms of domain classes:



The rectangular boxes denote the entities (or classes) with the class name in the header and the attributes (or fields) below it. The arrows define relationships between entities. These relationships can be of different types and are depicted differently using different arrow styles.

Broadly speaking, we can place class relationships into these categories:

- Dependency relationship
- Association
- Generalization
- Realization

Let's explore each of these UML relationships in detail.