Nicholas LaMantia

Professor Anderson

CS-255

15 October 2023

Evaluating an Object Model

The different functions of the online storefront are the ability to create an account using the user’s shipping, billing, and contact information. As well, transactions using debit and credit cards and notifications based on order status and confirmation are also functions. The actual functions are listed below the variables for each object. That is how the functions are represented; for example, placeOrder() is a function of the Order object. The different classes of “users” in this object model are the shopping cart, the order, the customer, the user, the shipping info, the administrator, and the order details. All these classes have an association with one another based on the object model. This association is either one to one or one to many. The objects would use their respective variables and functions to create a runnable program that executes as expected. For example, the verifyLogin() function will be used by the User class to verify if the user is able to access the account they are trying to log into. If the function returns true, then they have access. If it returns false, then vice versa. This object model captures everything it should as each function has a purpose.

The filled in diamond represents ownership of one class over the other. For example, the Order class has ownership over the shipping info. This implies that the relationship between classes with the diamond and those without are exclusive. The solid diamond is the appropriate choice here because those classes need to be exclusively owned. If Owner Details were also owned by the User class, then the user can modify the price of products, which would be a fatal flaw.

A process model does not explain the system well. It does well in illustrating the main points of the system and where the start and end points are, though in terms of abstraction, it is very high-level. This means that things like functions and the “how” of the system isn’t represented, which may or may not be something important to someone looking at these models. An object model better represents the system at hand. This is because with an object model, you’re able to see each function and variable tied with every class in the system. Those classes link to each other through relationships which makes readability improve. In all, the process model was easy to read and understand, however, the object model gave us a better understanding of the processes and objects we’re dealing with in this hypothetical scenario.