Dylan Dunagan

1 June 2025

CS-255 System Analysis and Design

Module 4 assignment Evaluate an Object Model

Interpret the object model

1. What are the different functions of the online storefront? How are they represented in this type of model?

The different functions of the online storefront are customer account management, product browsing and shopping cart management, order processing, shipping, notifications, and administrative functions. The functions that are in the UML diagram are register(), login(), updateProfile(), verifyLogin(), addCartItem(), viewCartDetails(), checkOut(), updateCatalog(), calcPrice(), updateShippingInfo(), and placeOrder(). They are represented this way because when it is time to produce the actual code, these will be the names of each individual function, and it already gives the programmer a full idea of what the program needs to be capable of.

1. What are the different classes of “users” represented by this object model? What are the associations between these classes?

There are only two classes of “users” in the UML diagram, the administrator and the customer. They both inherit from the “User” class to allow the system to verify what kind of user is logging in. Although they inherit the same info and functions of the “User” class, they both also have their own details and functions that the other doesn’t. For example, “Administrator” had the “updateCatalog()” function and “Customer” has “shippingInfo” and “accountBalance” elements, as well as the “register”, “login”, and “updateProfile” functions. This will be important when it comes to giving permissions to the various people that use the website

1. How would the objects “use” their respective variables and functions?

Each object within the UML diagram has its own variables and functions associated with them. Like a process model DFD, the UML diagram starts with the user class and works its way through the flow of functions until it reaches the end. The “User” class would ask for the variables of “userID” and “password” and then call the “verifyLogin()” function. This would then incorporate the “registerDate” and “loginStatus” variables. Depending on the user that just logged in, it would flow either to the “Customer” or the “Administrator” class and the process would start over. If it moved to the “Customer” class, it would have their information stored in each variable and would allow for each function to be called. Essentially, each object uses its variables to store the data that is required for each function to work properly.

1. Does this object model capture all of Hamp Crafts’ desired functionality? Why or why not?

This object model does not capture all the desired functionality that Hamp Craft wanted. First, Hamp Craft wanted both the customer and the store themselves to receive a notification whenever orders were placed. Hamp Craft also wanted the administrators to have customer support functionality. Unfortunately, neither of those functions are shown in the object model. On the positive side, it is an easy problem to find and fix. Other than those two functions, the model does show the other functions that Hamp Craft requested.

1. The above diagram uses a solid diamond shape to represent a form of aggregation. What type of aggregation does this represent? What does it imply about the relationship between the classes? Why is a solid diamond the appropriate choice here?

The solid diamond represents the composition relationship form of aggregation. Basically, this means that the child object cannot exist without its parent object. For example, if “Customer” was deleted, the “Shopping Cart” and the “Order” objects would no longer exist. Because the “Order” object would not exist anymore, the “Shipping Info” and “Order Details” objects would also cease to exist. The solid diamond and this form of aggregation is the appropriate choice because these objects should be tied to the “Customer” object alone. Whenever Hamp Craft adds the administrator’s ability to have customer support, those same objects would still be accessed through the customer’s account.

Compare these models

1. How well do you think a process model describes the system? What information does it make easier to understand? What aspects of the system are more difficult to understand or are not represented?

I think both models have their appropriate places in understanding systems. I think a process model is more appropriate in a business environment where they are explaining what they are wanting to people who are not programmers. I say this because a process model does not easily show data structures or the relationships between system components. It is also harder to show encapsulated logic. The process model does properly show how data flows through a system, making it easier to visualize business processes and workflow. It also helps with identifying bottlenecks.

1. How well do you think an object model describes the system? What information does it make easier to understand? What aspects of the system are more difficult to understand or are not represented?

The object model is built for programmers. I have used UML diagrams many times throughout my computer science studies and it makes the most sense to me. It completely shows the structure of the system and makes sure to incorporate what each variable and function that is required. Because of the way each object is visualized, it makes data encapsulation very obvious and really highlights how modular the system can be. The downfalls of an object model are that although I find it very easy to read, it could be confusing to people compared to the process model. Not everyone needs to know each variable. UML diagrams might show every function, but it does not show the sequence of operations, or at least not as clearly as the process model.