Brian Engel

Modular Four Object Models

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

The functions are on the bottom of the objects in the model. They are represented by +functionName() return type. The + at the beginning just means that the function is public. If there was a parameter for the function the type and name would be in the parenthesis. In the UML for the online storefront there are the functions verifyLogin, updateCatalog, register, login, updateProfile, addCartItem, updateQuantity, viewCartDetails, checkOut, placeOrder, updateShippingInfo, and calcPrice.

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

User is the super class and has the variable and functions that are shared by the classes that inherit from it. Administrator and Customer both inherit from the User class and have the variables and functions of the User class, but Administrator also adds the variables adminName and email, and the function updateCatalog. Customer adds the variables customerName, address, email, creditCardInfo, shippingInfo, and accountBalance, and the functions register, login, updateProfile. It seems to me that the variable email could have been put in the User class, since it is used in both Customer and Administrator.

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

In order to use the variables and functions of the object, you would need to create a instance of the object. After that is done, you can access it by calling the object name followed by the variable or function you want to use. Example: objectInstance.variableName or objectInstance.function().

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

The UML does not address if the billing address is different from the shipping or contact address. A separate class for billing should be added. The shopping cart is a little unclear, but if the checkOut function is the link to the established credit card vendor it would satisfy the payment portion, but it should return something to let us know the payment went through, and send something to the customer as well. Besides that I think that everything else has been covered.

* 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 is composition, which means that the class can’t exist without the class before it. I think it is appropriate in all the instances it is used in here. There is no shopping cart without a customer. There is no order without a customer. There are no order details without an order. There is no shipping info without an order.

* 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?

The process model seems to do a good job of visualizing the flow of the system and the sources and sinks or outputs of the system. Basically, it is really good at showing the functions portion of an object model. The downfall of the process model is it doesn’t have much detail on the objects themselves, such as variables, and I think the object model does a better job of showing dependencies and inheritance.

* 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?

I think the object model does a better job in describing the system because it seems to contain more information about the system. It might just be because I have used object models more. They hold information on variables, functions, dependencies, and inheritance. The object model is not good at showing some aspects of the system though, such as non-functional requirements like performance, security, and usability.