Evaluate an Object Model

Nicholle Caudy

CS255

9/22/2023

**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: addCartItem(), updateQuantity(), viewCartDetails(), checkOut(), register(), login(), updateProfile(), placeOrder(), verifyLogin(), updateShippingInfo(), updateCatalog(), and calcPrice(). They are all represented in the bottom half of each class rectangle. Each function with a “+” indicates that its attributes can be accessed outside of that class.

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

The different classes of “users” represented by the object model are Customer and Administration. The associates between these classes are userId, password, loginStatus, and registerDate. The Customer “user” has more properties to account for customer information, address, credit card information, and account balance. Whereas the Administrator “user” properties include the administrator's name and email address.

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

The class or object can access its functions available in that class. In doing so the object can perform those functions within the website environment. The variables of each object can be accessed through the functions, but only by using an instance of the object itself. However, accessing a private variable from outside the class cannot happen. For instance, in the class Shipping Info the function updateShippingInfo() can be accessed but the shippingId, shippingType, shippingCost, and shippingRegionalId cannot be accessed outside of that class.

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

For the object model to capture all of Hamp Crafts’ desired functionality there are a few things that need to be added. For instance, a function to send billing to an outside party should be added for their desire to use an established credit card vendor service. The object model seems to also be missing a notification interface for their desire to have their customers notified of the order status and confirmation. A function to access and view order history should also be added to the object model.

**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 use of the solid diamond shape indicates a form of aggregation called composition. This indicates that a part of that object belongs to only that other whole object. This is appropriate because the Shopping Cart class would not exist if the Customer class were deleted (the part of one creates the other). This holds true for the Order class in relation to the Customer class as well as the Shipping Info and Order Details classes to the Order class.

**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 describes the system very well. It does a great job of demonstrating the data flow and provides a big-picture approach for all involved. The process model for Hamp Crafts showed the flow of data between the stores and processes from customer to shipper and all the steps in between and gave a good base for the construction of the system. Some of the aspects of the system that are more difficult to understand or are not represented are the details that allow the system to function, this is where the object model comes into play.

**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 for Hamp Crafts expansion shows a clear picture of each class and what each will perform for their tasks. It illustrates the private attributes that cannot be accessed outside of that class as well as the public attributes that can be used and shared outside the classes. It makes each action easy to follow and each class connects to each other, including the form of aggregation called composition. As stated above a few things were missing such as the notification to the customer of order status and order history.