Pasco Logan

09/21/2022

CS 255

4-2 Evaluate an Object Model

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

Visitors to the site can sign up for an account by providing personal details like name, address, and email, as well as payment and shipping information like credit card details and preferred mailing location. Customers can now add items to an online shopping cart, complete and modify their orders, and examine their order histories; administrators have exclusive access to these features. The model categorizes certain e-commerce website features as either private or public, such as the following: addCartItem(), updateQuantity(),viewCartDetails(), checkout(), register(), login(), updateProfile(), verifyLogin(),updateCatalog(),updateShippinginfo()

placeOrder(), and calcPrice().

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

"Customer" and "Administrator" are two kinds of "users" that are modeled by this object's customer and administrator classes, respectively. The associations that exist between these classes are all represented as strings, and these strings include the name, the address, and the email address.

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

Each "Customer" and "Administrator" object has a login string where the user's name, email address, and other credentials are saved. The "Administrator" role will expand to include the ability to manage the online store's product catalog. Compositions of "shopping cart" and "order" objects will be available within the customer class. Also, the "Shopping cart" will allow for the addition of items, the modification of quantities, the inspection of cart details, and the completion of the checkout process, while the "Order" will allow for the placement of orders. The "Order" also incorporates the "Shipping Info" and "Order Details" composites. The shippingId(), shippingCost(), and shippingRegionId() variables are all integers, whereas the shippingType() variable is a string, and can all be added by the customer in the "Shipping Info" section. Additionally, it contains features such as updateShippingInfo (). For comparison, the "Order Details" section uses numeric orderId(), productId(), quantity(), unitCost(), and subtotal() variables while storing the text productName(). To determine the total cost of an order, use the calcPrice() function, which can be found in the "Order Details" section.

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

The administrator will not be informed of any user actions. However, the administrator has access to user accounts and can make contact with clients to offer assistance. While credit card information is collected, no actual payment processing takes place because there is no corresponding class.

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?

A black diamond indicates a more powerful composition than an aggregate because other classes cannot composite a composite class. The customer and order is a "has-a" composition relationship because if the customer ceases to exist, so does his/her order. Therefore, there is a life cycle dependency between the container (customer) and the instances within that container (order). When the container ceases to exist, so do all its instances.

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?

Process models give a planned and executed sequence of activities over time and yield a predetermined outcome. So while we don't see the fine details of how the system's duties are carried out, we see the overall process and intended functionality flow. This diagram serves as a template for all subsequent UML-required structural diagrams. Useful not only for coders but also for team members. Class diagrams allow Business Analysts to model systems from a business point of view. Some aspects of a poorly drawn system can be overcomplicated, and an overwhelming diagram is not helpful to software developers. There may be times when developers are frustrated by the structure of the class diagrams. Mapping out every possible scenario could make the diagram cluttered and challenging to work with.

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?

To someone who isn't versed in programming, I imagine UML diagrams to be confusing and even frustrating. A coder could make good use of this, but it's more involved and hence less intuitive for the layperson to understand where variables are created and/or changed.