4-2 Assignment: Evaluate an Object Model

Armon Wilson

SNHU

CS-255 System Analysis & Design

**Interpretation of Object Model for Online Storefront**

1. **Functions of the Online Storefront:**
   * **User Management:** The model supports user registration, login, profile updates, and login verification. This is essential for both customers and administrators.
   * **Product Browsing and Shopping Cart Management:** Although not explicitly shown, it's implied through the relationship with "Shopping Cart" that users can browse products, add them to their cart, update quantities, view cart details, and proceed to checkout.
   * **Order Placement and Management:** The model includes functionalities for placing orders, updating shipping information, and calculating prices. It also stores order details like date, customer information, status, and shipping details.
   * **Administrative Functions:** Administrators can update the product catalog, likely adding, removing, or modifying product listings.
2. **Classes of "Users" and Their Associations:**
   * **Customer:** This class represents the shoppers who interact with the storefront. It inherits properties from the "User" class and has specific attributes like customer name, address, credit card information, shipping info, and account balance.
   * **Administrator:** This class represents the store's administrators. It also inherits from "User" and has specific attributes like admin name and email.
   * **Associations:**
     + A customer can have zero or many shopping carts (0.\* relationship).
     + A shopping cart belongs to one customer (1 relationship).
     + An order is associated with one customer (1 relationship).
     + An order can have one shipping info (1 relationship).
     + An order can have multiple order details (1.\* relationship).
3. **How Objects Use Variables and Functions:**
   * **Customer Object:**
     + **Variables:** The Customer object stores information like customerName, address, creditCardInfo, shippingInfo, and accountBalance. These variables hold the customer's data, allowing the system to identify and track individual customers.
     + **Functions:** The register(), login(), and updateProfile() functions directly manipulate the customer's stored data. For instance, register() would take input values and assign them to the respective variables, creating a new customer record. updateProfile() would modify existing variable values.
   * **Order Object:**
     + **Variables:** This object uses variables like orderDate, orderStatus, totalPrice, and shippingDate to maintain a record of each order's details. These variables are essential for tracking an order's progress and history.
     + **Functions:** The placeOrder() function combines information from the customer's profile and shopping cart to create a new order and assign values to its variables. The updateShippingInfo() function would specifically modify the relevant shipping-related variables.
   * **Shopping Cart Object:**
     + **Variables:** Variables like cartID, quantity, and totalPrice store information specific to a customer's shopping session. These variables are updated as the customer adds or removes items from their cart.
     + **Functions:** Functions like addToCart(), updateQuantity(), and viewCartDetails() directly manipulate the shopping cart's contents. For example, addToCart() would add a new item to the cart and update the quantity and totalPrice variables accordingly.
   * **Other Objects:**
     + Similar principles apply to other objects like Administrator, Shipping Info, and Order Details. Their variables store relevant data, and their functions provide ways to create, modify, or access that data as needed.
4. **Does the Model Capture All Desired Functionality?**

The model captures most of Hamp Crafts' desired functionality, including user management, order placement, and basic administrative functions. It does not explicitly represent product browsing, catalog display, payment processing, or customer notifications. These could be added as additional classes and relationships.

1. **Aggregation (Solid Diamond):**
   * **Type:** The solid diamond represents composition aggregation, a form of aggregation where the lifecycle of the parts (Order Details) is tied to the whole (Order). If an order is deleted, its order details are also deleted.
   * **Implication:** This implies that order details cannot exist without an order, and they are an integral part of the order.
   * **Why Appropriate:** A solid diamond is the right choice here because it accurately reflects the dependent relationship between orders and their details.

**Comparison Between Process and Object Models**

Both models are valuable tools for understanding a system from different perspectives. A process model is better for understanding the "how" of the system, while an object model is better for understanding the "what."

* **Process Model:** Good for understanding the flow of activities and interactions between users and the system. It may not represent the system's structure and data as clearly.
* **Object Model:** Excellent for representing the system's structure, objects, classes, and relationships. It makes it easier to understand the system's data and how objects interact. It may not show the flow of activities as intuitively as a process model.