4-2 Assignment: Evaluate an Object Model

Jorgo Qendro

Southern New Hampshire University

CS-255: System Analysis and Design

Professor Trajkovski

September 28th,2024

Evaluate an Object Model

Introduction

Here are my responses for this assignment. I am interpreting the object model based on the company's needs for customer account creation, order management, payment processing, and administrative functions.

Evaluation of Object Model

1. Functions of the Online Storefront

The UML object model outlines the key functions of the online storefront:

- Customer Account Management: Customers can create an account, log in, and update their profiles using the `Customer` class. Methods such as `register()`, `login()`, and `updateProfile()` facilitate these functions.

- Shopping Cart Management: The `Shopping Cart` class enables users to add items, update quantities, view cart details, and proceed to checkout. These actions are captured by methods like `addCartItem()`, `updateQuantity()`, and `checkOut()`.

- Order Management: Customers can place orders, check the status, and review order details using the `Order` and `Order Details` classes, where functions like `placeOrder()` and `calcPrice()` operate.

- Administrative Tasks: The `Administrator` class allows updates to the product catalog using the `updateCatalog()` method.

2. User Classes and Associations

The object model presents two primary user classes:

- `Customer`: The class represents individuals who interact with the online store by creating accounts, placing orders, and managing their profiles.

- `Administrator`: This class represents users who manage the backend operations, particularly product catalog updates.

Customers are associated with `Shopping Cart` and `Order` objects, enabling them to make purchases and track order history.

3. Object Variables and Functions

Each object in the system makes use of specific variables and functions to fulfill its role:

- The `Customer` object uses variables like `customerName`, `email`, and `creditCardInfo` to facilitate actions such as registering and logging in.

- The `Shopping Cart` object handles `productId`, `quantity`, and `cartId` to manage items in the cart and proceed to checkout.

- The `Order` object stores attributes like `orderId`, `status`, and `customerName` to enable order tracking and status updates.

- The `Administrator` object uses the `updateCatalog()` method to manage product listings.

4. Missing or Incomplete Features

The object model captures most of Hamp Crafts' required functionality, including account creation, shopping cart management, and order tracking. However, it does not explicitly show the integration with third-party payment vendors or the notification system for order status updates. These features could be enhanced by adding a `PaymentGateway` class for third-party transactions and a notification system.

5. Aggregation and Composition

The solid diamond in the UML diagram represents composition, implying that the contained objects (like `Order` and `Order Details`) are part of a whole and cannot exist independently of their parent objects. This is appropriate for scenarios where the destruction of the parent would result in the destruction of the associated objects, such as an `Order` and its corresponding `Order Details`.

Comparison of Process and Object Models

1. Process Model Description

A process model focuses on the flow of activities within a system, making it easier to understand the sequence of operations and data flow. However, it may not capture the detailed interactions between entities or the structure of the system, which can be important for understanding relationships between data.

2. Object Model Description

An object model, like the UML diagram for Hamp Crafts, excels at capturing the system's structure and showing relationships between objects. It makes it easier to understand how data entities, such as customers, orders, and shopping carts, interact. However, it may not clearly represent the dynamic flow of processes or interactions over time.