# CS 255 Model Application Short Paper

[John Vowcicefski]

[John.Vowcicefski@snhu.edu]

Southern New Hampshire University

## Process Model Application

QUESTION: How would you apply a process model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.

A process model would be highly beneficial for designing the DriverPass system, as it focuses on the sequence of activities and the flow of data within the system (Satzinger et al., 2016, p. 154). By applying a process modeling approach, the development team can better understand and optimize the various business processes involved in the DriverPass system, such as user registration, online learning, reservation management, and reporting.

A process model could visually represent the steps a customer goes through when signing up for an account, browsing available courses, and scheduling an on-the-road training session. This model would help identify the inputs, outputs, and decision points at each step, making it easier to design a system that seamlessly guides users through the process.

Process models can be used to map out the administrative workflows, such as managing customer data, updating training materials, and generating reports. By clearly defining these processes, the development team can ensure that the system provides the necessary functionalities and interfaces to support efficient operations.

Techniques such as data flow diagrams (DFDs) or business process model and notation (BPMN) can be made to create detailed visuals of the DriverPass system's processes. These models would serve as a communication tool between the development team and stakeholders, allowing for better collaboration and requirements validation.

## Object Model Application

QUESTION: How would you apply an object model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.

An object model is essential for designing the DriverPass system's structure and defining the key objects, their attributes, methods, and relationships (Satzinger et al., 2016, p. 196). By applying an object modeling approach, the development team can create a clear and organized representation of the system's components, making it easier to understand, implement, and maintain.

For the DriverPass system, an object model would include classes such as User, Customer, Administrator, Instructor, Course, Lesson, Quiz, Reservation, Vehicle, and Report. Each class would hold the relevant data and behavior, creating a reusable design.

For example, the Customer class would have attributes like name, email, phone number, and address, along with methods for registering an account, enrolling in a course, and making a reservation. The Course class would contain attributes such as title, description, duration, and price, with methods for adding lessons, quizzes, and updating content.

The relationships between these classes, such as associations and aggregations, would also be captured in the object model. A customer could have multiple Reservations, while a Course is composed of multiple Lessons and Quizzes.

Class diagrams and entity-relationship diagrams (ERDs) can be used to visually represent the DriverPass system's object structure. These models would serve as a blueprint for the development team, guiding the addition of the system's components and their interactions.

## Process and Object Model Comparison

QUESTION: What are the advantages of each model for the DriverPass scenario? What are the disadvantages of each model for the DriverPass scenario?

A process model excels at capturing the dynamic behavior and flow of activities within the system, making it easier to understand, analyze, and optimize the business processes. It provides a clear picture of how the system should function from a user's perspective, helping to identify bottlenecks, redundancies, and improvement opportunities.

On the other hand, an object model focuses on the static structure and organization of the system's components, enabling a more modular, reusable, and maintainable design. It promotes encapsulation and abstraction, allowing for easier modification and extension of the system's function over time.

While process models are better suited for understanding and streamlining the system's workflows, they may not provide a detailed view of the system's architecture and the relationships between its components. Object models excel at representing the system's structure but may not capture the different aspects of the system's behavior as effectively as process models.

In the case of the DriverPass system, creating both process and object models would provide a comprehensive and balanced approach to system design. The process models would help optimize the user experience and ensure efficient operations, while the object models would guide the development of a robust and scalable system architecture.

By using process models to validate and refine the system's requirements and object models to design its structure, the development team can create a high-quality system that meets the needs of DriverPass and its customers. The combination of these two modeling approaches would result in a system that is both user-friendly and technically well off.

**References**

References: Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016). Systems analysis and design in a changing world (7th ed.). Cengage Learning.