# CS 255 Model Application Short Paper

Sarah Dowd

sarah.dowd1@snhu.edu

Southern New Hampshire University

## Process Model Application

A process modeling approach for the DriverPass system would focus on mapping out the sequence of activities and data flows between different system components and users. This approach would be particularly valuable for capturing the complex interactions in DriverPass's reservation and training processes.

For the reservation system, process modeling would detail the steps from initial customer contact through lesson completion. This would include mapping how reservation requests flow through the system - whether initiated online by customers or entered by the secretary - and how these requests trigger subsequent processes like instructor assignment, vehicle allocation, and payment processing. The model would show how data moves between different system components, such as when customer information is collected, validated, and stored in the database.

Process modeling would be especially effective for representing DriverPass's training package workflow. It would illustrate how students progress through their chosen package, from initial enrollment through completion of online tests and in-person driving sessions. The model would capture important decision points, such as package selection, lesson scheduling, and test completion status updates. It would also show how the system handles DMV updates, incorporating them into training materials and notifying relevant users.

For administrative functions, process modeling would map out workflows for user management, report generation, and system maintenance. This would include processes for password resets, account modifications, and activity tracking, clearly showing how different user roles interact with the system and how their actions trigger various system responses.

## Object Model Application

An object-oriented modeling approach for DriverPass would focus on identifying the key entities in the system and their relationships. This approach would be particularly useful for organizing the system's complex data structure and business rules.

The object model would define core classes like User (with subclasses for Admin, Secretary, Instructor, and Customer), Package (containing attributes for different training options), Vehicle, and Reservation. Each class would encapsulate related data and behaviors - for example, the Package class would maintain information about included services, pricing, and duration, while implementing methods for enrollment and progress tracking.

For the training component, object modeling would define classes for Course Materials, Practice Tests, and Test Results, establishing relationships between these objects and the User class. This would effectively represent how students interact with learning materials and how their progress is tracked. The model would also include objects for managing DMV updates and their integration into the training content.

The reservation system would be modeled through interrelated objects representing appointments, vehicles, instructors, and locations. This approach would naturally enforce business rules, such as preventing double-booking of instructors or vehicles, through object relationships and constraints.

## Process and Object Model Comparison

The process modeling approach offers several advantages for the DriverPass scenario. It excels at representing the dynamic flow of activities, making it easier to understand and optimize the reservation and training workflows. This approach is particularly strong at showing how different system components interact over time, which is crucial for complex processes like lesson scheduling and package management. It also helps identify potential bottlenecks and inefficiencies in the system's workflows.

However, process modeling has some disadvantages for DriverPass. It may not effectively capture the static structure of the system's data and relationships. This could make it harder to understand how different entities like users, packages, and reservations are related. Process models might also become overly complex when trying to represent all possible variations in the system's workflows.

The object-oriented approach offers different advantages. It provides a clear structure for organizing the system's data and functionality, making it easier to maintain and modify the system over time. This is particularly valuable for DriverPass's requirement to potentially add or modify training packages in the future. The object model also naturally supports the role-based access control requirements through inheritance and encapsulation.

However, the object model has its own limitations for DriverPass. It may not as clearly represent the sequential flow of activities, making it harder to understand and optimize complex workflows like the reservation process. Additionally, object modeling might make it more challenging to visualize how system components interact in real-time, which is important for features like concurrent reservation management.

The optimal approach for DriverPass would likely involve using both modeling techniques in combination, leveraging the process model's strength in representing workflows while using the object model to establish a solid structural foundation for the system.