# CS 255 Model Application Short Paper

Dylan Drake

dylan.drake@snhu.edu

Southern New Hampshire University

## Process Model Application

A process model defines the step-by-step tasks involved in using a system. For DriverPass, this type of modeling is helpful for mapping out how students and staff will interact with the platform. It would show how a student registers, chooses a training package, schedules lessons, and completes online practice tests. Each user role would have a separate process flow. For instance, secretaries would have steps to register students by phone and book appointments, while IT staff would manage user access and account resets.

Using a process model would allow the development team to clearly understand how each user interacts with the system. It would also show dependencies between tasks. For example, a lesson cannot be booked without first selecting a package. Visual tools like activity diagrams and flowcharts can help identify potential delays or inefficiencies in the system's workflow before development begins.

## Object Model Application

An object model is focused on identifying the major components of a system and how they relate to one another. In the DriverPass scenario, key objects include Student, Appointment, Driver, Car, Package, Test Result, and User Account. These objects each have properties and behaviors. A Student has a name, contact information, and selected package. A Driver has a schedule and is assigned to specific appointments. These objects interact with one another to form the full system.

Object modeling makes it easier to manage complex relationships and allows for better code reusability. For instance, if DriverPass later adds a new type of training package, only the Package object needs to be updated. This modeling approach supports flexibility and growth, which is important for future system updates and enhancements.

## Process and Object Model Comparison

A process model provides a clear view of the sequence of actions users take when using the system. It is useful for mapping out how tasks are completed and how users interact with each part of the system. It helps stakeholders understand the workflows and is helpful during early system planning. However, process models are not as strong when it comes to defining data structure or planning for long-term growth.

An object model focuses on the system's components and their relationships. It supports modular design and makes future system changes easier to implement. While it may be harder for non-technical stakeholders to understand at first, it helps developers plan more scalable and efficient code. This model is particularly useful for applications like DriverPass where multiple user types and resources need to be managed over time.

Both modeling approaches are valuable. A process model helps define how the system is used day to day, while an object model helps define how the system is built and maintained. For DriverPass, using both together ensures that user needs are met while also allowing the system to grow and adapt in the future.