# CS 255 Model Application Short Paper

Darrell Walker II

darrell.walker@snhu.edu

Southern New Hampshire University

## For this DriverPass project, two types of modeling approaches can be used: process modeling and object modeling. Each approach has its own strengths and weaknesses, and it’s important to understand how they apply to the project and the differences between them.

## Process Model Application

In a process modeling approach, we would focus on breaking down the tasks DriverPass needs into clear, step-by-step workflows. Each process would show the flow of actions a user takes to accomplish a goal, such as booking a driving lesson or taking an online practice test. For instance, if a student wanted to schedule a driving lesson, the process model would map out each step involved: logging in, selecting a lesson package, choosing an available date and time, confirming the booking, and receiving a confirmation message. Each action would be represented in a flowchart that shows how the system guides the user through these tasks.

The process model could also help us identify potential problems in the workflow, such as bottlenecks or inefficiencies. For example, if several students try to book lessons at the same time, we need to ensure the system can handle that traffic without crashing or double-booking instructors. The model can help identify areas where we need to improve the system's performance or efficiency. It also allows us to define business rules, such as ensuring that students cannot book a lesson if there are no available cars or instructors.

Additionally, the process model would make it easier to explain how the system works to stakeholders. By showing the flow of actions, we can communicate more effectively with DriverPass management or other non-technical staff. They can easily understand how a student’s journey through the system should look, which is important for ensuring the system meets their business needs. Process models are particularly useful for managing the user experience and ensuring that every process is completed smoothly and without errors.

## Object Model Application

On the other hand, an object modeling approach focuses on the things, or "objects," in the system and how they relate to each other. Objects are things like "Customer," "Lesson," and "Instructor." Each object has its own properties (like a customer’s name or a lesson’s time) and behaviors (like scheduling a lesson). For DriverPass, an object model would be used to show how these different objects interact. For example, the "Customer" object might book a "Lesson" object, which would have its own set of details like date, time, and instructor. This approach makes it easier to break the system into smaller, reusable parts, and changes can be made to one part of the system without affecting others. This is especially useful for larger, more complex systems where different parts need to work together.

Additionally, object modeling promotes reusability, as each object can be utilized in various contexts throughout the system. For instance, the "Lesson" object could be reused across different types of driving packages, whether it’s a basic package with three lessons or an advanced package with additional in-person instruction. This modularity makes it easier to add new features to the system, like a new type of driving package or an upgraded payment method, without needing to redesign the entire structure. Object modeling also emphasizes encapsulation, which keeps data secure within the objects themselves. For example, customer data, like their payment information, would be safely stored within the "Customer" object and only accessible through specific methods, helping to protect sensitive information. This methodical approach ensures that the system remains organized and scalable, making it ideal for long-term growth and maintenance.

## Process and Object Model Comparison

When comparing the two approaches, each has its advantages and disadvantages. Process modeling is simpler and easier to understand, especially for non-technical people like managers or customers. It shows the flow of activities in a clear way, which makes it easy to see how tasks are completed. This helps to ensure that the system will support the business needs. However, it’s less flexible because it focuses more on how tasks are done rather than how data is handled. If there are changes to the system, like adding new features, the entire process flow might need to be redesigned.

In contrast, object modeling is more flexible and focused on data. It allows for easier changes because each object works independently. For instance, if DriverPass wanted to add a new type of driving package, they could add a new object without having to change the entire system. Object models are also great for systems that need to grow or change over time. However, they can be more difficult to design and understand, especially for people who aren’t familiar with programming. This can make it harder to explain how the system works to non-technical stakeholders.

In conclusion, both process modeling and object modeling have their benefits for a project like DriverPass. Process modeling is easier to understand and helps ensure the tasks are completed in the right order, but it can be rigid and harder to adapt. Object modeling, while more complex, offers flexibility and is better suited for systems that need to grow or change in the future. Choosing the right model depends on the priorities of the system, and in some cases, a combination of both models could provide the best solution.

**References**

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). Systems Analysis and Design: An Object-Oriented Approach with UML. Wiley.

Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.