# CS 255 Model Application Short Paper

Nicholas Kreuziger

nicholas.kreuziger@snhu.edu

Southern New Hampshire University

## Process Model Application

The process model approach will require gathering the system requirements and analyzing what is already in place. DriverPass is a new business so we will not have an existing system workflow to consider. With the interview transcript in hand, I would then spend time with my team drafting up a data flow diagram in conjunction with user stories.

The data flow diagram will be centered around processes that relate to the DriverPass business. The DriverPass business is primarily gathering data from customer info, scheduling, driving assessments, test results and their internal employees. That data is maintained by Customers giving their information, customers or employees scheduling appointments, teachers making assessments, and students taking assessments. The data from these efforts is sent into databases as sinks, the how and what of these databases will have to be determined. Mapping this entire process would ensure a detailed data flow diagram.

The user stories will center around a few key players. The customer, the employee and the teacher. These three players are the orchestration behind the business, as such their vantage points and interactions with the system will have to be roleplayed to net all of the requirements. Ideally the key players are interviewed, but it is not outright required.

## Object Model Application

An Object Model provides a concise summary of the objects within a system, their relationships with other objects, their attributes, and their methods. In the context of the DriverPass scenario, an object model could be used to define the system thoroughly and facilitate smoother software development.

The relationships between objects are an important aspect of the object model, and could be used to represent different types of relationships between users and the system. For example, a user class could be used to define traits that both the customer and teacher possess through inheritance, while encapsulation could be used to protect sensitive information like passwords and payment details.

Polymorphism could also be used to demonstrate different types of assessments or lessons, allowing for greater flexibility in the system. For example, an assessment object could take on different forms, such as a written test, driver test, or other types of evaluations.

Attributes of the customer, employee, and teacher objects, as well as their relationships to scheduling and assessment upkeep, could be thoroughly outlined by an object model. Additionally, the methods and functions of these objects could be defined, such as a customer having the ability to create an account or schedule a session.

Overall, the initial mock-up of the software could be based upon the object model, providing a clear and comprehensive guide for the development team.

## Process and Object Model Comparison

Process modeling has several advantages such as providing a visual representation of the system's workflow and data flow, which helps to understand how different parts of the system interact with each other. It also helps to identify potential bottlenecks or inefficiencies in the system's processes and provides a clear guide for development teams to follow when building the system. However, a process model may not capture all the system's complexities or nuances, especially when it comes to the interactions between different objects or entities. Additionally, it can be time-consuming and resource-intensive to create and maintain, especially if the system undergoes significant changes.

On the other hand, an object model provides a concise summary of the objects within the system and their relationships with each other, making it easier to understand the system's structure. It facilitates smoother software development by providing a clear guide for the development team to follow and can help identify potential issues with the system's design early on in the development process. However, an object model may not capture all the system's processes or workflows, especially when it comes to the interactions between different objects or entities. It can also be more difficult to create and maintain than a process model, especially if the system is complex and involves many different types of objects or entities.

In designing a system like DriverPass, both process and object models have their advantages and disadvantages. A process model may be more useful for capturing the system's workflows and data flow, while an object model may be more useful for understanding the system's structure and relationships between objects. Ultimately, the choice of model will hinge upon the stakeholder needs as the project ensues.