# CS 255 Model Application Short Paper

Dhiraj Gurung

Dhiraj.gurung@snhu.edu

Southern New Hampshire University

## Process Model Application

A process model is a type of development tool used to show “an abstract representation of the development process” (Sulemani, 2024, para. 2). Process models help provide a high-level outline for the things that need to be done in a software development cycle. They help to provide developers with information on how their workflow and schedule will likely look during the development process. In order to apply the process model to the DriverPass scenario, we need to first analyze the information given in the project. We would then need to identify the different processes, data stores, and data sinks. Once we can identify these components within the system, we can place them and map them out in a Data Flow Diagram (DFD) and then use arrows to show the direction in which the data flows. I would use a DFD to show how the booking process of the system will work. There needs to be different process diagrams to handle the following processes: receiving customer orders for different packages, checking out at the payment page, notifying the customer of sale confirmation, and scheduling different trainings. In the following processes, the main data source will be the customer. The flow of data will start from the customer and flow forward to the first process of receiving customer sales, then the checkout process, then the email notification process for sale confirmation, and lastly the scheduling process. The schedule information should be sent to a data store that has information on both driver availability so that the customer can better choose which dates to conduct their training. The data will flow from the schedule process to the scheduling database and back.

## Object Model Application

Object models are frameworks used to show the different types of Object-Oriented Programming (OOP) concepts like “abstraction, encapsulation, modularity, hierarchy, typing, concurrency, and persistence” (Booch et al., 2011, Chapter 2, para. 1). Applying these OOP concepts can help programmers write code that is more modular and organized allowing for more sustainable coding (GeeksforGeeks, 2025). To implement the object model in the DriverPass scenario, we would first need to figure out the different objects, like functions, that we need to implement for the system. Next, we would list each element and attribute associated with the different functions. We would also use the appropriate Unified Modeling Language (UML) notations and draw the relationship symbols between different objects to specify the type of relationship. I would take a look at the different functions that would be needed, like a function to help with scheduling trainings, to verify customer login information, to handle customer bookings, to help the administrator customize and edit bookings, a function to handle password resets, and a function to notify DriverPass of any changes in the DMV database. Afterwards, I would use these functions to create an object diagram. In the object diagram, I would show the different attributes of each function and then use different object diagram notations to show the type of relationship between the diagrams.

## Process and Object Model Comparison

The advantage of the process model in the DriverPass scenario is that it helps to show the sequence of events that occur. The process model will help us visualize a high-level outline of which processes occur when and which way the data will be sent and received (Sulemani, 2024). This means from the start of the booking process to the end, we will get to see the data flow and the sequence in which events work in order to allow customers to book and schedule their individual trainings. The advantage of having an object model is that we get to see the structure of the system. In this model, we get to see which attributes and information belong to each class and function. We also get to see how the different objects are related to one another, and there are different notations to denote different types of relationships. There are also different notations to show if an attribute is public or private. A plus sign denotes that a function or attribute is public, while a minus sign indicates that they are private. Object models help make it easier to understand how each function and class within the system work because of how much detail is added about each of the objects associated with the system. For example, the object model will show the different attributes required for the login information and verification of the login information. There will also be information on all the other functions, like the function that allows the administrator to customize and edit the packages.

There are also disadvantages to using each of the different models. For example, when using the process model, we do not get a detailed view of the data and types of information involved with each process. For example, in the booking process, we will not see what package the customer chose or when the customer scheduled their training. We only get to see a high-level view of how the sequence of events will unfold when a customer wants to book different trainings with DriverPass. On the other hand, when using the object model, we get to see more detail and the structure of how the system will work. However, there is no information on which event will occur first, or information on the sequence in which each function or process will occur.

## References

Dennis, A., Wixom, B., & Tegarden, D. (2012). *Systems analysis and design with UML (*4th ed.). John Wiley & Sons.

GeeksforGeeks. (2025, July 23). Object model: Object oriented analysis & design. GeeksforGeeks. https://www.geeksforgeeks.org/system-design/object-model-object-oriented-analysis-design/#benefits-of-object-model

Greg. (2021, March 2). UML class diagram arrow types: Explanations and examples. Gleek. https://www.gleek.io/blog/class-diagram-arrows

Sulemani, M. (2024, May 27). What is a software process model? Top 7 models explained. Educative. https://www.educative.io/blog/software-process-model-types