# CS 255 Model Application Short Paper

Daniel Schween

[Daniel.schween@snhu.edu](mailto:Daniel.schween@snhu.edu)

Southern New Hampshire University

## Process Model Application

Process modeling is a business-driven diagram that helps the analysis team, the client, and other engaging parties organize the system’s requirements. The client has needs and specifications for their system, and to fulfill those requests in a timely manner, a model is necessary. The client, The IT department, administration, owners, and developers share the process model to communicate their goals with one another.

Process modeling describes the data flow between sources and sinks, processes, and data stores. In the case of DriverPass, the client, the user, the administration, the IT officer, and the developer can be classified as sinks or sources. These actors relay information to and from the processes modeled in the diagram. For example, the IT officer needs full access to user accounts so they can delete infrequent users or reset the passwords of accounts that forgot their password, or if their data has been breached. The owner must have full access to the system so he can delete packages that are offered to user that they no longer want. Additionally, the owner must have access to the system and user’s accounts so they can download reports including test scores, scheduling, and conducted road tests. The owner must analyze the numbers so they can make improvements as needed.

Data stores are mostly used in the back end or non-functional requirements. DriverPass must have data stores to support different platforms, such as Windows or Mac. Users with different platforms accessing the system must be accounted for, and databases help support each platform. Process models help identify how each of the subjects work together through data flow. Process modeling also break down the processes into smaller steps that help detail the requirements better.

## Object Model Application

Object modeling focuses on the “players” at work to get the process done. An object model highlights the class/object, its attributes, and its functions that are used to meet the requirements of the project. Some objects for DriverPass would consist of the user, the secretary, the system, the drivers, and the back end. These objects can now be broken down into their functionality and attributes. For example, the secretary has the function to schedule appointments, and the attributes would be gathering user information. The owners have a function to modify packages or accounts by using attributes of the user.

Object models break down the processes that meet the client’s requirements into classes. These classes have a function on the overall system to contribute to the end goal. With these functions, attributes are necessary because there is relationship or instance between each object.

The relationship between objects is represented through different annotation, such as a solid diamond to represent aggregate composition. In the case of DriverPass, “driver notes” cannot exist without the object, the “driver”. Inheritance is represented as a hollow arrow. For example, the “IT Officer” inherits attributes and functions from the parent class “User”. An object model describes relationships and methods between objects that are used in programming code.

## Process and Object Model Comparison

The process model aids the entire business, including the user, the client, the system analysis, and the developer. Process models help improve efficiency by mapping out what they know and what they need to improve on or add. Process models also enhance communication because it is a visual language that is easy to understand, and all the parties involved can interpret the model. Process models also break down larger processes into smaller steps; this break down allows the business to analyze costs in an efficient manner because the more steps needed, the more steps and time must be allocated to address the requirement.

An object model breaks down the processes into classes that a developer can understand. Identifying the class, attributes, and methods are key to writing a program. Understanding how each object interacts with one another will allow the developers to write a more concise and efficient program. Relationships between objects are very important and it is a necessity to identify before writing a script.

The difference between object models and process models is how they are used. Process models support communication between the different teams working on the project. It also helps identify key requirements that were left out. A process model can be updated to include the missing requirements. Process models are used to help all parties of the project understand how the system works. It is a language most people can understand. On the other hand, object models are tailored for the system analysis and developers to implement the requirements of a project.

Both models have their purpose and time to be used. I believe which model should to select depends on who is viewing it.

In conclusion, both models are very useful to fulfill a client’s request, and both models should be implemented before working on a large project with a team. For the case of DriverPass, I believe both models should be used to illustrate the different functions and data flow between objects and processes. Using both models will ensure all requirements are met and the system being built is efficient and user-friendly.