# CS 255 Model Application Short Paper

Guilherme Pereira

guilherme.pereira@snhu.edu

Southern New Hampshire University

## Process Model Application

DriverPass has a system in place that wants to focus on student drivers being able to access online resources to prepare for an exam, while also allowing them to schedule and modify their appointments with real life driving instructors. While there are multiples ways a student can go about fulfilling their agenda, a process model would be a good idea to show how exactly a student or user is intended to navigate through the system. Process models depict how a business system operates, or how a system is intended to function under normal circumstances. A process model would help demonstrate how data flows from start to finish and how the user can go from logging in to setting up an appointment.

There are certain methods that are considered good practice when developing a process model. This ranges from being realist, agile (flexible), all the models should collaborate with one another, and that the model is an abstraction of reality. Initially, the first step when creating a process model for DrivePass is identifying who the actors are and creating an initial node (starting point) for them. These initial nodes will then flow into the appropriate object node, or actor, and show what steps they need to perform in the system. The next step is detailing what actions and activities each actor can perform. Following this are decision branches that can connect to one another or creation a control flow that shows how one action leads to the next. Finally, a final-flow node must be shown in the model to show where all the data will end up going to in the end. These diagrams also have certain restrictions such as having one initial node, having this node placed at the top left of the diagram, and following a down-right path from start to finish to show the flow of information.

For DrivePass, the focus is on the student users who want to access their information online. Now that the student has been identified as the initial node of the process model, we must consider what the student can do. According to Liam, the head of DrivePass, student users will be able to: take online classes, practice tests, on-the-road training, schedule appointments, view current progress, modify personal information, and add any special needs if needed. For a student user to perform any of these actions and activities, they must first be registered on the system. The initial node will go into the user object node and the first action would be to connect to the website. Secondly, the user must create an account to have their information saved onto the system and use those services. A create account activity is followed up next and at the same time, an object flow will go to the data base object node to demonstrate that the user’s information is being stored there. The user will then need to fill in the required information: first and last name, address, city, state, zip, phone, and email to create their account and have access to the site. Next, we come to a decision branch that can go in multiple ways. At the home page, the student can schedule appointments, modify their settings, take an online class or practice test, upload a student photo, and view their online test progress along with driver notes.

If the student were to schedule an appointment, they would need to fill a form online or call the secretary to have the appointment set up by them. In a process model, this would show that a student needs to fill in a form and this form is then sent to the secretary object node. The secretary will then approve and decline this appointment and provide feedback to the user. The online test progress would show the user their current progress and notes from the driver; this information will be retrieved from the DrivePass data base with current information if the user is connected online. If a student wanted to modify their settings or information, they would need to go to change settings and then confirm. This information is passed from the website to the database and then back to the website. The final-flow node could end anywhere, but the goal is for the student have completed their driving classes and tests. Therefore, we begin with the student logging online, creating a DrivePass account, providing detailed information on the account, taking online classes, taking online tests, taking on-the-road driving exams, and finally completing all these benchmarks so that they are ready to take the DMV exam.

## Object Model Application

While process models focus on how information flows through the system, object models focus on the “what” of the system. Specifically, what each object in the system can do and what their attributes detail. While there is no specific flow for object models, it is connected by different forms of relationships such as association, aggregation, generalization, and aspects of polymorphism are also shown. For object modeling to work with DrivePass, we need to determine what the class object are (actors), what attributes and functions each of these objects are capable of or hold, and what functions they interact with in the system.

To begin, we look at the different actors in our system: student user, secretary user, IT user, and admin user. The baseline for all these actors is that each has the same attributes which we would list in an entity class that they inherit from. The attributes they hold would include personal information, such as first and last name, address, city, and all the previously mentioned attributes. For student users, they can schedule an appointment, taking online classes and tests, and viewing their progress. For a secretary class, they can confirm or decline appointments as well as create or delete student accounts at the request of the student or owner. The IT class can modify the status of the website, which means temporarily disabling parts of the system until they are fixed or retrieving information from the database that was previously lost (such as a forgotten password). Finally, the admin class would be able to perform all of these functions and handle the status of each user by reviewing a student user’s profile or deleting them from the system.

Since each actor now have their own defined attributes and functions, we can interconnect them among one another to show how these interactions occur. For instance, a student user would have a 1 to 1 association with the secretary while the secretary has a 1 to many with the students. The student can schedule an appointment with the secretary while the secretary can confirm or deny this appointment. These interactions are known to be part of a structural model, which helps show the structure of the system and the purpose of each object.

In contrast to the structural model, object models can also showcase different object behaviors. While this was lightly touched before, the state of an object can change during the process of completing the DrivePass course (from not completed to completed). Behavioral modeling shows the case by case of an object interacting with another object, such as a student creating an appointment, an IT member retrieving a lost password, or the admin creating a detailed report from a student’s profile.

## Process and Object Model Comparison

Process and object models share two different perspectives of the DrivePass system: the flow of information and the capability of the system’s objects. While object modeling is more commonly used today and provide further detail as to how an object can interact and behave within the system, process modeling is just as important to show how data should flow through system from start to finish. The strength of a process model is showing how information flows from one point to another. While this seems simple, this passage of information can be further broken down to smaller diagrams, each with more detail as to how information can be accomplished. Granted, process models should not be broken down all the way to make an extremely complex diagram but should provide enough information to understand how data travels from one point to another. The biggest weakness to process modeling is that it shows the general scenario of how the system should be interacted with. Some users may handle the system differently and never reach the final-flow node of the system, opting to use the on-the-road service only for experience instead of taking the full course. Additionally, detailed information regarding what each object-node can do is lacking in this model.

Object modeling covers some of the weaknesses of process modeling, in that it provides attributes and functions for what each object is and can do. The biggest strength of object modeling is sharing how each object can interact with one another in a system, what is reachable to that object, and how the states of an object could change in a system. Object modeling can show both the structure and behavior of the system. The one weakness object models have is that they cannot show the process in which data flows through the system. There might be a general idea as to how the objects interact, but there is no set flow for them to follow. This can become an issue from a business perspective because a system without a purpose or flow is useless to a customer who has a need to be fulfilled.

In conclusion, both process and object models provide strengths that help detail a system’s purpose while also helping to cover one another’s weakness. It is to the benefit of DrivePass that both models are to be created to explain the different perspectives of the system. The process model would help give stakeholders a general idea of how data flows through the system while object models give developers and clients a more thorough explanation as to how each object interacts in the system.