# CS 255 Model Application Short Paper

Danny Faught

danny.faught@snhu.edu

Southern New Hampshire University

## Process Model Application

When creating a process model for an application, it should display the users (or other systems) that will be interacting with our system, AKA actors, and how each actor will use the system. This is displayed with lines that associate our actors with different use cases.

For our application we have several different actors. The most apparent ones will be the students that subscribe to our services. We will also have instructors, employees, IT, and Management actors that will need to be modeled. Another actor that we will need is our backend that will modify our databases.

In a process model we also need to list our use cases. Our students will be associated with use cases that will facilitate in creating a profile, managing their profile, managing subscriptions, and managing appointments. Our instructors will use the system to see the schedule and interact with students. The employees will modify user information and schedule or manage appointments, IT will need to modify user information, disable classes, manage appointments, and update content. Management will need the ability to run reports as well as modify databases. Our backend will need to add, update, delete, or modify databases and access updated DMV content.

## Object Model Application

When creating an object model for an application, we need to see how the objects, or data, move through the system. This is not the same as a class diagram, but instead give more of a basic overview of physical things, roles, incidents, interactions, records, catalogs, policies, and many others that all interact within a system, and then in what ways they can interact with said system. (Citation)

When creating an object model for DriverPass, we would include objects for each of the roles we have already identified (Student, Instructor, Employee, IT, Management or Admin) as well as student vehicles and schedules, user profiles, read, write, update, delete objects for modifying each database, as well as many others.

An object model shows how objects are associated with each other using numbers and lines of various types. I will not list all the possible associations in this paper, however as an example: A student object would be associated with an appointment object by drawing a line between the two objects. A student can have zero or many appointments, but an appointment cannot exist without a student. This would be shown using numbers above the line drawn.

## Process and Object Model Comparison

In my opinion, process and object models serve two different purposes. A process model will give a very general description of how a system will operate when specific users interact with it. This is a great way to convey how a system will work to non-developers or clients that will be involved with the project. It does not, however, give a good description of each of the processes that will be needed for a project.

An object model will do this a little better as it can break down a system into various objects and show how each will be associated with each other in a system. I find this more suited for the team working on the project rather than a client because while someone could hypothetically determine how a user would interact with a system based off the object model, it is far more complicated and is better suited for displaying how data flows through a system.

## References

Dennis, Alan et al (2012). [*Systems Analysis and Design with UML, 4th Edition*](https://learning.oreilly.com/library/view/systems-analysis-and/9781118037423/) Wiley