# CS 255 Model Application Short Paper

[Christopher Harvestine]

[christopher.harvestine@snhu.edu

]

Southern New Hampshire University

[**Note:** Complete each section below. Each response should be a few paragraphs in length, and your paper should be between 3 to 5 pages. Use 12-point Times New Roman font, double spacing, and one-inch margins. Be sure to use proper grammar and APA format to cite any sources you use, including the textbook. Remove this note before you submit your paper.]

## Process Model Application

[How would you apply a process model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.]

I would apply a process model to a design with a flow diagram. I would have at least two flow diagrams. The first one would be from the perspective of the student, it would start with validation of credentials. The flow diagram will include the multiple failure sign in process. If the user fails to sign in 10 times, the account will lock and redirect the user to their email for password reset and account unlock. Different options will have branching paths to explain all of the functionality. For instance if the user signs on they will have multiple options, they can edit their profile information, view reading material, take quizzes and tests, view on the road instructor feedback. Weekly progress will also be shown, as will any submitted tests and quizzes. If a student clicks on this week's study material, once it has been reviewed they may attempt a quiz. After passing that quiz an overall progress bar will increase and new weeks may be attempted ahead of time. The administrator's flow chart will be different. Once the administrator has verified credentials they will be able to access a program that updates the UI of the program. The administrator will also have the option to edit,modify users as needed. The flow diagram will explain what happens when they make a selection and what options they will have from there. The entire process can be represented with a flow chart and should help the programmers understand the system so they can just focus on coding.

## Object Model Application

[How would you apply an object model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.]

I would apply an object model to a design for DriverPass through a UML diagram. I would first create a base class called user. The user base class will have something every user will need, username, password and email. The student object will inherit from the userbase class retaining the fields and adding more such as student specific fields and functions. Some fields to add would be student information, name, address, phone number. The functions added will be specific to the students actions, take tests, view material, schedule an appointment. The administrator object will also inherit from the user base class and have its own fields and functions to add on top. The administrator will have a UIEditor object which allows them to eidt the DriverPass homepage, this will be useful if there is new information such as laws, or information pertaining to the course. The administrator will also have an editStudent object which allows them to edit students, take them out of the system if they passed. The editStudent object will also allow them to fix any issues if there are bugs in the system, like if a student passed a test but it didn’t show. Although no code will be on the UML diagram, it should give the team members a pretty good indication of each user’s role and functionality, especially if the object's fields and functions are named properly.

## Process and Object Model Comparison

[What are the advantages of each model for the DriverPass scenario? What are the disadvantages of each model for the DriverPass scenario?]

A positive advantage of the DriverPass scenario in the process model is it does a good job outlining how the system is going to work without it getting too technical. The process model does a good job showing how each option is connected and where it leads. A disadvantage of a process model would have to be that although you get an understanding of the flow of the system there aren’t variables, functions, those aren’t included, only how to navigate the system.

A positive advantage of the object model within the DriverPass scenario would be the finer details of specific objects. Details such as base objects, fields and functions combined with inheritance to differentiate between object purposes. This gives a programmer a good idea of how each object functions on it’s own separate from the overall program. A disadvantage of using the object model with the DriverPass is it would be hard to communicate how the objects relate to the overall program. The programmer knows how to code the objects and has a basic idea of the functionality of each object, but what action leads to where, this is something they would not know.

It seems as though both models are needed in order to get a full understanding of the program. I would see process modeling being done first as it lays out the overall flow of the program. Process models will help put team members on the same page and allow for better communication. Once the process model has been completed the team can plan out the individual classes, their fields, functions which is object modeling. When I created my video game for the Xbox 360 I went straight to object modeling not knowing what it was. I did not use process modeling and I had to recode my objects to add functionality to something that I didn’t plan for. If I were to create my game now I would use a process model first and then an object model.