# CS 255 Model Application Short Paper

Michael Savage

michael.savage1@snhu.edu

Southern New Hampshire University

## Process Model Application

When applying a process model for the design in DriverPass, I would start by implementing a flow diagram with user stories from the customer side of the process. Getting the idea of the customer in the planning process is critical in the long-term development process for the application. A process model is designed to provide a clear explanation on how data is used, how it is stored, or how it is changed or modified in between phases for each class in the system.

Starting off with the customer, I would move the flow diagram to provide the customer with the capabilities to access the system, log in to the system, allow the customer to change password, or access a method to reset their password in case they forgot. I would also implement a branch that allows the customer to update their contact information, schedule appointments, change payment information or check their course progression. When scheduling appointment I would also provide a branch that allows the customer to have a secretary do it for them, in order to assist with their process.

When considering the multitude of different branches, we would have to do in this method I find the a flow diagram is the best way to approach this model for our client DriverPass. The flow diagram provides a smooth “flow” form one action to the next and gives a clear understanding of how this method will work. Flow diagrams provide one of the better options when it comes to branching from different steps in the process as well.

## Object Model Application

When applying the object model, I would consider using a UML diagram to efficiently explain the system as an object model. An object model is used primarily for data orientation in system design. For this case I will use the UML diagram to describe the system through a chart showing the different methods, classes, and the attributes associated with each class.

When designing the UML diagram, I would first determine the classes that will be implemented into the system, examples of this would be customers, secretary, IT admin etc. For example, when designing the object model for the customer class we would add attributes and methods for this class as well. Methods we could implement would be creating, deleting, and modifying the customers information. We could also add a method that allows the customer to book appointments, modify appointments, cancel appointments, take a practice exam, view driver, take a practice class, or view total progress. The secretary would be another class within the system. The secretary could have the potential to modify, delete or cancel customer appointments as well as updating contact records for customers. The next class would be the IT admin class. This class would have the capabilities to have access to both the secretary and the customer classes. This would also allow the IT admin to fix bugs and update the systems as required. The final class would be the System Owner. The system owner would have all the methods of the previous classes, as well as the method that allows the owner to download data form the system.

When designing this Object Model, I found that the UML diagram was the best option. The UML diagram presents a clear representation of what all users on the system can do as well as providing a visual “roadmap” for how the development team will develop this system.

## 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?]

In comparison, I found that both the Process and Object Models provide their own advantages and disadvantages associated with each of them. One major advantage that I would consider for the object model is its ability to convey the behavior of a system. The object model has many different processes such as Gantt charts and UML diagrams. Then Gantt chart provides great detail for a project’s overall analysis. It also provides a visual aide for the development team so that they can view how each class and program is intended to function.

However, one major disadvantage that the Object Model faces is how it is understood by a client. You must investigate an Object Model with a technical perspective otherwise understanding the model can become a bit difficult. Presenting an Object Model to a client with no technical experience is not ideal as you may need to explain the system more than would be necessary for a process model.

An advantage that we see with the process model is how easily it can be presented and understood by a client. It provides the scope of how the system is intended to run by showing the client a step-by-step roadmap of the project.

A major disadvantage that the Process Model faces that it does not provide a clear explanation of how the system should behave. It leaves many programmers confused on how they will begin to develop the project, as the Process Model only shows branches of what is the most important process of the system.

## References

Dr. Dobbs. (1994). *Dr. Dobbs Journal and Microsoft Corporation.*

https://www.cs.umd.edu/~pugh/com/