# CS 255 Model Application Short Paper

Brian Larson

brian.larson1@snhu.edu

Southern New Hampshire University

## Process Model Application

The process model is focused on how things flow. Where does the process start? What happens next? It does not rely on who is processing the data. It does keep an idea of where the data should go in an abstract sense. The process model should focus on the steps of the process and what comes after each step, including branches.

Therefore, the process model would outline the flow of data from start to finish along all possible lines of use within the system. This would include the processes for registering an account, scheduling an appointment, or even running the reports for the CEO. With this, we may be able to identify a new method of handling the process that could improve DriverPass’s business model and perhaps increase their ROI.

## Object Model Application

The Object model is focused on the “what” of the system. What are the actors? What data will be sent between them? What does this data look like? Now that we have the flow mapped out with the process model, we can start mapping the objects that need to follow that process.

As we accomplish this, we will start to see how the data relates to other data, how different roles might overlap or share functions, allowing us to better design the system to be as efficient as possible. With encapsulation and polymorphism, we can condense data to be sent, protect critical parts of the system and build base class templates for classes that share functions. But this can only be done once we understand how the different roles or types of data relate to each other.

## Process and Object Model Comparison

One of the advantages of the process model is mapping the flow of the process. When we do this, we can see the pain points, bottlenecks or other areas that can be improved. This can often be easy to visualize, which could assist in gathering data for the requirements. It helps the stakeholders focus on what the system should be doing.

A disadvantage of this would be that it is hard to imagine what kind of data needs to follow the processes as you follow the model. It might be easy to see what needs to happen next, but difficult to tell how to actually make the jump from one step to the next within the system. Another drawback is that process mapping gets harder with scale. As the system complexity increases, so too does the complexity, and therefore the confusion, of the process model.

An advantage to object modelling is clarity when it comes to thinking about the code. Many object models are designed around object-oriented conventions, and therefore converting to code is fairly straight forward. This also allows data templates to be identified and reused where possible. This can also help with modularity, not only making parts interchangeable, but also assisting with encapsulation for packaging or security.

On the other hand, however, object models are hard to follow when it comes to tracing the process. Some can find it very difficult to see what the next steps might be, or how to decide which branch to follow in a decision tree. This is not ideal when it comes to dealing with stakeholders, who need to be able to understand just what the system is doing. That way, they can describe what is missing, or what needs to be cut, in order to make the best to-be system.

But despite the strengths and weaknesses of each model, both are required to really get the best picture of the system. By showing the how and the what of a process together, one can then start to see the full system as it should be.