# CS 255 Model Application Short Paper

Cooper Adams

cooper.adams@snhu.edu

Southern New Hampshire University

## Process Model Application

A process model provides a graphical representation of the workflow occurring in a space or system. For the DriverPass scenario, I would create a data flow diagram (DFD) to represent the procedural flow of using the website. In fact, multiple DFDs could be created, as the user interface and the options within would differ for customers and employees of DriverPass.

Every diagram would begin with a process of signing in/up, but the options diverge from there. A customer can register for courses, see their assigned driver, access practice materials and tests, and much more. An employee could update their schedule or profile, view their course materials, and review student progress. The IT administrator can handle support, reset passwords, and restrict account access. Additionally, the owner of DriverPass can view and download reports about the site or financial information. Nearly every action a user could take when on the site can be listed as a process in a DFD, and the only true order within all of it is that the user would have to login first. That is the process model is important, as it will help the design of the system by providing easy to read diagrams of all the potential processes occurring in the final system.

## Object Model Application

The object model provides a visual representation of a systems’ actions, attributes, objects, and relationships. For the DriverPass scenario, this is an interesting application as the system itself is accessed through a webpage made with HTML and JavaScript (and CSS, for styling). JavaScript can technically be considered “classless”, but it does display prototypal inheritance as all of its’ objects are derived from the Object object, so there are literal objects that can be diagrammed in an object model. For DriverPass, this would be the users of the system, as each “type” of user (customer, employee, IT admin, owner) has unique data and things to access on the site, which can be shown in an object model (likely a UML diagram) as object specific attributes and methods. Other components of the system can be modeled too, of course, but the user modeling is a fantastic example as it displays multiple object-oriented principles:

* Data Abstraction – The user only sees data relevant to their “role” in the system
* Encapsulation – Attributes and methods are packed into a single component (the user object)
* Polymorphism – Each type of user can make use of the same function (signing in or out, updating their profile, etc.)
* Inheritance – Each unique user inherits from the original user object

The object model is clearly beneficial to designing the system, as it not only displays relationships, properties, and methods, but also object-oriented principles, which are assuredly helpful when the system is heavily reliant on an object-oriented language such as JavaScript.

## Process and Object Model Comparison

Object models can better capture the relationships and properties of objects within the system, which allows the design team to reuse code and concepts throughout the project, while also providing structure to build with, bettering the system’s maintainability. Object models are also typically fast to develop, as one would have to consider only relationships and parts of an object, as opposed to an entire system’s behavior. Object modeling may lack cohesion though, as its’ structuring doesn’t outright define any activities, which could lead to missing or misinterpreted behavior.

Process models give a holistic view of the system in terms of its’ processes, which can lead to better analysis of the system’s behavior. Defining the processes and sub-processes of an entire system could be quite time consuming, though, which is possibly a negative for DriverPass considering the few months the design team has been given to deliver the system.

Clearly, both models have pros and cons relevant to system design, but they tend to make up for each other’s weaknesses, so in my opinion, the design team should make use of both the object and process model when developing DriverPass’s system as it should lead to a product that is both logical in its workflow and informative in its object representation.