# CS 255 Model Application Short Paper

Garret Gherardini

Garret.gherardini@snhu.edu

Southern New Hampshire University

## Process Model Application

A process model allows users to visualize the comprehensive workflow of a current system in place or shows what is anticipated for how a workflow should be managed during the creation of a new system. Our client, DriverPass, has an existing system for their brick-and-mortar business that needs to be translated to a virtual system. Utilizing their current system, many steps that can be taken to break down the workflow, transitioning into a functional virtual system by adding or removing elements that may be necessary to fulfill the desired tasks.

To apply a process model to design an appropriate system for our client, there are a number of factors that should be taken into consideration. A thorough evaluation of the existing system for any inefficiencies can provide clear guidelines for how the workflow should function in the new system. This can eliminate redundant steps, allow for automation of processes, and optimize efficiency to better serve their customers. Following evaluation, clearly defining roles and expectations for each task is necessary. This provides the system designers with a clear expectation for the duration of time that should be spent on each task throughout the course of the design process and provides system customers with an anticipated date to receive the final product.

When implementing a process model for a business, like DriverPass, it is imperative to keep in mind how much “wiggle room” there can be with the system. When DriverPass presented to our team they only had ten cars and one instructor per vehicle, it was clear they were still in the early stages of their business. The implementation of their virtual system should support any growth in their company, so scalability should be considered when designing a process model. Flexibility in requirement changes over the course of their business’ growth should be kept in mind when designing their system. This can include adaptation and expansion from a procedural standpoint, like the number of drivers/vehicles available and designing a system to support these projected changes, to designing a framework that is scalable. This would enable their system to be adaptable to their growing business and the process model should support anticipated changes in exponential growth.

## Object Model Application

An object model enables designers to emphasize the primary objects and relationships between each component of a system. Object models are heavily focused on these relationships as they showcase how one object relates to another or a group of other objects. Breaking down an object model into categories known as classes is a common method for showcasing these relationships. With every class that is represented by an object, it demonstrates how the functionality between other classes, or objects, should be implemented during the design process.

Breaking down these relationships into broad interactions clearly defines the system’s intended functionality. For example, our client, DriverPass, has the intention of wanting to provide their customers with the option to purchase a specific package that allows the customer to reserve specific components of their driving lessons. They should be able to reserve their lessons at a designated time with a designated instructor and vehicle. With that aspect being the primary focus of our client’s requirements for our system, it is imperative to show in our object model how this will be implemented. The object model can be expected to show a specific class for reserving these in-person driving lessons and how that reservation interacts and functions with other scheduling options such as updating the duration of time a customer wishes to reserve for instruction or wanting to reschedule a lesson.

To further break down the components we anticipate seeing in our object model, defining all object attributes should be clearly made prior to the development process. An object, such as a “user” (in one instance the customer) should be able to customize their login requirements, to some degree. The object model should reflect all anticipated functions of the user such as entering their name, email, and creating a unique password upon creating their account. It should also show what options the customer is provided to choose from when selecting a specific driving lesson package, viewing resources provided by the company, and progress reports. This should also be the case for any administrator; although they will have a different type of access, the object model should clearly define how the object’s functionality is inherently different and should be iterated in a secure manner under a separate object or class.

## Process and Object Model Comparison

Both process models and object models have their own unique approach for displaying the functionality or workflow of a system. At each model’s core, they should clearly define the system’s overview making it easy for a developer and a customer to see its intended purpose. Both models should be able to analyze the existing system (in the case of DriverPass a brick-and-mortar system) and define characteristics that prove to be more efficient by means of implementing a virtual system. Providing these models enables a clear picture of how the anticipated new system can solve any inefficiencies in the existing system and potentially reduce costs and improve workflow efficiency for the business, DriverPass.

While a process model is excellent for defining the broad workflow of a system, it does severely lack the functional details provided by an object model. The process models allow for easy identification of the system’s workflow, but do not show developers how each object and class should be interacting with one another to make these workflows efficient. An object model is capable of showing the workflow of a system but at the same time provides insight into how the system is capable of managing the data and object behaviors.

Object models approach a system’s workflow by means of defining relationships that should exist within the system to make them more efficient at storing and organizing data. It is clear the practices of object-oriented programming are kept in mind when designing an object model, especially when it comes to scalability (as mentioned previously). With a client like DriverPass, their system should be capable of supporting growth as the company expands. Showing room for expansion in an object model allows developers insight to appropriately accommodate any new behaviors or objects that may be necessary to implement as the company grows. A major disadvantage that exists for object modeling is the extended length of time spent designing the system to support the unknown or anticipated expansion of the business. The creation of the system could potentially take longer than anticipated based upon pure speculation that the business, in the near future, may require additional objects or classes to support their growing list of requirements.

Generally, both process and object models can be used to appropriately lay the framework for a new system as required by our client DriverPass. Both models have an array of advantages and disadvantages in providing our client with a clear picture of what to anticipate with the new system being created. Dependent upon our client and the depth at which they anticipate knowing about the system prior to its development is at their discretion. Personally, I believe utilizing both models can be beneficial to our client as they both can provide all encompassing information that is representative of the workflow and functionality of the desired system. However, I believe a process model is probably the best route as developers and customer alike can see how the relationships exist within the system and provide insight into how different actions can be taken to perform a variety of functions while still showcasing the relationships and general workflow of the system.

## References

Valacich, J. S., George, J. F., & Hoffer, J. A. (2019). Modern Systems Analysis and Design (9th ed.). Pearson Education (US). Retrieved from <https://mbsdirect.vitalsource.com/books/9780135172827>