# CS 255 Model Application Short Paper

Kennedy Uzoho

Kennedy.uzoho@snhu.edu

Southern New Hampshire University

2/15/2022

## Process Model Application

Let’s begin by defining what a process model is. Assuming a process model models a system and its processes, wouldn’t it be easier to analyze the system using a process model since it is a representation of that system? In system design, a process model is used to perform an action on data, so the data can be either transformed, stored, or distributed in a certain way.

The process model is applied to understand how data flows, how it is stored, how it is distributed, and how it is modified during an active process. I will be leveraging the process model application and use-case diagram to model DriverPass system for a better understanding of the system by analysts, developers, product managers, and other specialists involved. The system model will be broken down into different parts, such as actions, functions, behaviors, and classes. This is to reduce the complexity of the system, which will leave us with a simple and comprehendible developing system. Process models facilitate solutions to solve a real-world problem. For example, DriverPass system process model will be facilitating a system that will prepare student drivers to understand the driving concept, so they can pass their DMV driving test.

Starting with a use-case scenario, assuming it is a customer’s/guest’s first time using the system, the system would prompt the guest to answer a list of questionnaire questions with all information about the guest needed to create a customer’s profile. With a confirmed profile, the user now a customer can be asked to “Select Package”, “Reset Password”, “Purchase Package”, or perform other functions that can be added. Then, the user can checkout after purchasing a package. The system will recognize the purchase by its identification number, then the system will now assign the same id with the purchased package to an available driver as scheduled. The driver, customer, car, date, time, pick up location, drop off location will all be recognized according to their functions and data flows. The system will provide a space for “driver’s comment”. It will display the lesson in progress, the driver and customer involved, the scheduler information (date, time, time used, etc.) on the system application interface.

Security can be handled by the head of the security department/Administrators, and they would need to include a “security requirement” for the system. The administrators will be able to create an administrative profile that will allow them to perform some specific tasks. The secretory will have permission to schedule, add, delete, or modify a customer’s profile information including their courses, training, and test schedules. All users, attributes, modules, objects mentioned so far have a unique process for completing their designated task, these tasks are performed with the help of a process, thus process model will then entail what is exactly happening as the users, attributes, or other items in the system performs their tasks.

## Object Model Application

Object modeling really focuses on real-world modeling systems. If a system designer or programmer is asked to design something that is objective, the programmer needs to reference the model as it is in real life. For the most part, almost everything we see and know is based on objects oriented, that is why an object-oriented modeling system is best for any objective system design and analysis. The diagramming tool (UML diagram) is an objective graphical notation, it is used to construct and visualize object-oriented systems. There is also a class UML diagramming system. When describing object programming, it is as important to also describe “class” because “class” makes up an object in programming. For example, “human being” is a class, an object would be “Kennedy, me” because I have unique variables driven from a base class “human being”, so somebody can describe me using my skin color, height, voice, and other variables and attributes a “human being” should have.

To analyze DriverPass using object modeling applications, we will first consider the different users. Initially, we considered the users of the system to be guest/customer, secretary, owner/big boss, and IT officer. Each of these users will need a “username”, “password”, “user-id number (if needed)”, “email address”, and “phone number” to register on the system. We can call these dependencies user variables because these variables differentiate each user from another user. Some users might need a special variable declaration. With users and variables declared, we can now add some methods and functions to our user variables, for example, we can take the object/user “customer” then add some functions such as “login”, “reset password”, “verify user”, “update password”, “update user information”, “select package”, and “checkout”. Another example, the administrator can have functions like “update system”, “design layout”, “print system report”, and “add or delete user acc”. Each of these functions and variables could also be decomposed or composed into another thing or object, like embedded system arrangement, however, after the initial declaration of functions and variables, all the components will be needed for the system to function perfectly.

The system security could also be managed in the same manner. For example, the security officers could decide to add more security user objects, we already have a “username” and “email” for identification then the password for security, the IT officers can decide to add additional objects, modules, or variables like “security user number”, “security credentials”, or “authentication code”

## Process and Object Model Comparison

In comparison to process and object models, both models provide crucial information about a system. However, each of them provides unique and specific system information in a systematic way. For example, the process model deals more with the important process that a system follows to perform a specific task. The process model will not explain the behavior or method of a system and that is where the objective model kicks in. The object model explains the behavioral aspects of a system, objects in a system, modules in a system, variables in a system, class in a system, and the object model entails how all these aspects are put together from an attribute to a full-functioning system. During project development, I would love to use both models because each of them will provide a different view of a developing system. The choice of using each model will depend on the type of data or information that is being worked on. However, the objective model gains more attention because it also describes some processes as a bypass product. For example, the object model will break things down and describe variables, methods, and how they cooperate with various parts of a whole system, and they describe the entire system in terms of its object types. So, the objects model includes processes by using fluctuating definitions of variables. For example, declaring variables requires to break down of a given system process to know what and how to declare the variable, so the variable will have a unique and fitting definition.

## References

Valacich, J. S., & George, J. F. (2018). *Structuring System - Process Requirements.* In Modern

systems analysis and design (9th ed., pp. 179-201). Boston, MA: Pearson.

Redding, G., Dumas, M., Hofstede, A., & Iordachescu, A. (2007, September 24). *Transforming*

*Object-Oriented Models to Process-Oriented Models*. Retrieved August 2, 2020, from

<https://link.springer.com/chapter/10.1007/978-3-540-78238-4_15>

J., Braz. (2006, August 12*). A Case Study Applying Process and Project alignment Methodology*. scielo. br

https://www.scielo.br/scielo.php?script=sci\_arttext&pid=S0104-65002006000400006