# CS 255 Model Application Short Paper

Winnie Kwong

winnie.kwong@snhu.edu

Southern New Hampshire University

## Process Model Application

The process model is a visual representation that describes and captures the workflow of all activities that contribute to the goal set based on a person’s vision. Utilizing the process model builds a roadmap for engineers to understand key concepts of what, when, and why things happen while identifying possible issues. The client, DriverPass, has a vision of a new system to help reduce the number of people that fail the driving test. Applying the data flow diagram (DFD) would help establish the process outline within the DriverPass scenario.

When looking at the DriverPass scenario breakdown, the DFD focus should identify the flow of activities and information between various actors such as customers, instructors, and the system administrator. The customer, the central factor of each company, will start by creating an account that will redirect the user to the homepage. Once the registration is complete, the data flow of the customer will transition to allow the user to make appointments, cancel and modify appointments, take practice tests, and edit their account information. After customers have taken practice tests, the instructor will be introduced to the DFD to provide user feedback. Allowing instructors to provide clear and high-quality feedback allows the customers to focus on areas of improvement and brings them closer to obtaining a driver’s license. If the customer has any appointment inquiries, the system administrator will assist. The system administrator will also be responsible for managing user accounts and resetting user passwords when requested.

Without grasping the complexity that intertwines the customer, instructor, and system administrator, the DFD would not be descriptive, prescriptive, and explanatory during the growth of an application that defines the DriverPass scenario. DFD must be comprehensible for the client’s goal to be achieved and reduce the number of failures of the driving test.

## Object Model Application

The object model visualizes the software or system that includes objects, attributes, actions, and relationships. Using an object model helps show how things behave and work together in the assigned tasks. The object model best suited for the DriverPass scenario would be a Unified Modeling Language (UML) diagram defining the four elements of object-oriented programming (OOPs): inheritance, abstraction, encapsulation, and polymorphism.

The structural UML diagram would include various classes, objects, packages, and components in the system and the relationship between the elements. The step is to identify the model's key classes, including user, customer, instructor, driving lessons, package, and administrator. From there, it determines the methods that specify the class's operations and attributes. For example, the class would have the package name and price as attributes, the methods of the package's accessor and mutators (getter and setter), and the package price. Once classes have been established with correct attributes and methods, the UML diagram will include connectors such as inheritance, association, aggregation, and composition. Examples would show the customer and administrator using a connector to inherit all attributes and methods from a user. In contrast, package and driving lessons would use a composition connection to show those classes cannot exist without a customer. Effectively inputting the system’s classes and connectors based on DriverPass enables the design to describe the system based on the objects and classes.

Object models give a different perspective to understanding the decomposition of how a real-world process function. By understand the UML diagram’s objects, functions, and relations, it allows a dynamic type to check the web application.

## Process and Object Model Comparison

Knowing the advantages and disadvantages allows the client to see different perspectives that may occur during decision-making. However, working with models can determine if the system will become efficient or viable for the business.

The advantage of the process model is that it can help provide clarity, coordination, and quality control to the DriverPass scenario. Project models visually represent the workflow by defining the activity phases within the DriverPass scenario. By successfully mapping the roadmap with explicit knowledge of activities and relationships, it provides a clear and transparent workflow to streamline and automate tasks that increase the overall efficiency of DriverPass. Also, coordination between employees will improve by tracking the performance and productivity of the process model. Working together efficiently helps identify any issues, such as bottlenecks within a process and prevent other inefficiencies. Furthermore, process models can enhance quality control by providing consistent guidelines on handling the process. Having quality control reduces the number of reworks and saves the business valuable time and resources. With process models, DriverPass could grasp the internal procedures to manage the application.

Unfortunately, with all advantages, there are also downsides to using the process modeling, such as limited innovation, misunderstandings, and wasting valuable assets. Instead of being open to finding new ways of thinking, the strict framework of process models hinders innovation within the company. It causes uncertainty because of its inability to adapt to change. Additionally, with a rigid framework and lack of innovation, process models can lead to over-focus on the data flow, oversimplifying the context. If the workflow becomes oversimplified, critical details can be left out, leading to misunderstandings or misinterpretations, causing employees to feel disengaged and perform poor-quality work. Failing to provide a well-presented process model costs the company time, money, and other valuable resources while damaging customer experience. Suppose the company cannot understand the phases of the workflow. In that case, customers that will use the application will have the same effect, negatively impacting customer satisfaction, resulting in poor public-profile impacts. Using process models, DriverPass can risk inadequate innovation, misconceptions, and misuse of valued assets.

The benefits of utilizing an object model help to sustain context, improve maintainability, and increase flexibility. Object models offer modularity and reusability by using inheritance and polymorphism that simplifies the development and management of complex data structures. Reusing a concept allows the company to customize and adapt the components to an individual’s needs. Also, using object models helps improve maintainability because it shows efficient representation and management of complex data structures and relationships. Object models are handy for applications that require integrating different data types and sources, such as the DriverPass scenario. Lastly, object models have more wiggle room to adapt to changing requirements. Since object models can be altered, it provides a practical basis for computer implementation. Without understanding the benefits of an object model’s ability to reuse, maintain, and be flexible, it would be challenging to demonstrate object behaviors and how objects would perform real-world tasks.

There are disadvantages of using object models that include risking complexity, insufficient allocation of time, and compatibility errors. As projects being to grow and become more complex, the object models may become more difficult in capturing all the behaviors and interactions of an object. If the object model is complete incorrectly, it becomes a learning curve and technicians may not be familiar with the process causing more issues. Also, with engineers becoming unfamiliar with the system, it can lead to an increase in development time and wasting valuable resources. Furthermore, not all processes are compatible with the object model. Compatibility issues with other database models, which can limit their interoperability and integration with other systems and applications. Unsuccessfully creating an object model leads to complex situations, improper resource allocation, and incompatibility issues.

When knowing the pros and cons of a situation can help to assess the performance and workability of the model. Furthermore, once the base of the model is established, the model can suggest ways to improve the system as well as enhancing the operation for a business to evolve.