# CS 255 Model Application Short Paper

Euntak Jang

euntak.jang@snhu.edu

Southern New Hampshire University

## Process Model Application

Using a process model in the context of DriverPass would include specifying the sequential actions that the system must take to achieve its goals. This might include the steps a user must take to sign up, log in, retrieve lost credentials, change personal information, engage with the system, and log out.

Understanding may be acquired by modeling these processes using tools like Data Flow Diagrams (DFDs) or flowcharts. These diagrams depict the process flow visually, making it simpler to identify steps that may need additional optimization or have room for improvement.

After establishing a complete process model, it may be reviewed to discover inefficiencies, redundancies, or bottlenecks that can be eliminated or simplified. For example, if the sign-up procedure is very complex, it might be streamlined to improve user conversions.

Communication is the process model acts as a link between technical and non-technical stakeholders, making it easier for all parties to comprehend and agree on how the system will operate.

A process model can offer a baseline against which improvement attempts can be assessed by providing a tangible description of how the system now functions. This might be especially advantageous for DriverPass if the system changes to accommodate changing user demands or company objectives.

## Object Model Application

## An object model, on the other hand, would entail describing the system in terms of distinct, self-contained items. In DriverPass, there might be items such as User, Vehicle, Admin, and so on.

## Modularity is when each object would contain its own data and activity, emphasizing concern separation and modularity. User-related processes, for example, would be encapsulated within the User object, irrespective of the Vehicle or Admin objects.

## Objects enable the management of system complexity through abstraction. This means that, while each object is complicated in and of itself, it can be understood in terms of its interfaces - what it can do and what it can be done to - without having to comprehend its inner workings.

## Reusability is when objects may be made reusable. A Vehicle object, for example, might be utilized in many situations throughout the system, maintaining consistency and avoiding code duplication.

## System maintenance is typically simplified by object-oriented architecture. Changes to the implementation of one object do not necessarily impact others, as long as the object's interfaces stay unchanged.

## Process and Object Model Comparison

**Advantages and Disadvantages of Process Model:**

**Advantage:** The process model depicts how the system works in a straightforward and comprehensible manner. This can assist all stakeholders, including those with no technical expertise, in comprehending the system's workflow and interdependencies. This might be especially useful in a system like as DriverPass, where effective communication between technical developers and non-technical stakeholders is critical.

**Disadvantage:** However, as the system evolves, process models may become complicated and unmanageable. If the DriverPass system includes more characteristics or processes, the model may become difficult to manage and analyze.

**Advantages and Disadvantages of Object Model**

**Advantage**: Because the object model encourages modularity, abstraction, and reusability, it is perfect for sophisticated, scalable systems such as DriverPass. Because each component can be created and tested individually, the whole development process may be made more efficient.

**Disadvantage**: The fundamental downside of the object model is that it may be difficult for non-technical stakeholders to understand. Understanding concepts like encapsulation, inheritance, and polymorphism can be difficult for people without an object-oriented programming background. As a result, it may be necessary to expend more work in order to properly convey the system's design to all stakeholders.

## References

Valacich, J., & George, J. (2020). Modern Systems Analysis and Design (9th ed.). Pearson