# CS 255 Model Application Short Paper

Alexander Flood

alexander.flood@snhu.edu

Southern New Hampshire University

## Process Model Application

To design the DriverPass system, an effective process model focuses on defining the workflows and behavior of the system. “Process modeling involves graphically representing the functions, or processes, that capture, manipulate, store, and distribute data between a system and its environment and between components within a system” (Valacich & George, 2024). For example, the registration process involves users entering personal information, selecting services, and making payments. Each step is mapped out in a process flow diagram or data flow diagram, illustrating the sequence of actions and decision points for the given case.

In the DriverPass system, key processes include user registration, lesson scheduling, compliance updates, system updates, and basic administration operations. Each process is broken down into detailed flows. The user registration process starts with a new user entering their details, followed by email verification, selection of services, payment processing, and confirmation. Similarly, the lesson scheduling process includes checking availability, booking a slot, sending confirmation, and updating schedules.

These workflows are documented using flowcharts and process diagrams, ensuring that every action and possible outcome are defined. This highlights the behavior of the system and how it handles various scenarios, such as successful user registration, payment failures, and making changes to scheduled lessons. Each process is linked to specific system functions, ensuring that the system meets the client's needs.

To apply the process model to the design of DriverPass, the system architecture, each module should correspond to each major function. For example, a user registration module should handle user verification, password changes, and resets. The payment module should handle client payments, refunds, and transfers to external accounts. The scheduling module should manage lesson bookings, cancellations, and modifications to scheduled lessons. Each module interacts with other modules to form a cohesive and robust system.

By focusing on the flows and behavior of a system, the process model ensures that the DriverPass system is organized, user-friendly, and meets the client and business needs.

## Object Model Application

An effective object model is critical to designing the DriverPass system, as it helps structure the system using real world entities and their interactions. Key objects such as User, Employee, Admin, Schedule, Lesson, Payment, and others are identified. Each object has attributes and functions that belong to it. For example, the user object will have attributes such as username, email, and password with functions such as registerUser() or resetPassword() that support basic user functionality like creating a new user or resetting their password. Relationships between objects are established, such as a user being able to book multiple lessons, or an admin who manages multiple users or scheduled lessons, or the relationship between the parent user class and the employee and admin sub classes that extend the user functionality. These relationships help inform developers as they program and code the system modules.

Class diagrams can be used to visualize these objects, their interactions, and relation to each other, showing how each object has its own scope and responsibility. This also helps promote object-oriented design by allowing us to see relationships and design our objects as efficiently as possible. For example, using an object-oriented approach, we can create the employee and admin objects as sub classes of the user object, extending its basic functionality by reusing code we have already written. Using a detailed object model can help keep the system aligned with client needs and promote maintainability while providing excellent technical documentation to support the development of the DriverPass system.

## Process and Object Model Comparison

Process models and object models each offer different advantages when it comes to the development of the DriverPass system. Process models provide a clear overview of workflows and behaviors, giving us a consistent view of what our overall system does. It helps us map out sequences of actions and key decisions that help inform the project timeline and scope. To provide a detailed process model, significant effort is needed in planning, documenting, and requirement gathering to develop an accurate process model. Teams may research other systems, domain experts, and conduct user interviews to gather accurate requirements and user stories that can help inform the model, which can take some time.

In contrast, the object model is excellent at providing a technical overview of the structure, objects, and relationship between these objects that make up a system. When a system's objects are designed properly, the system is easier to understand, maintain, and scale. Often, the objects we see in this model map to the real-world entities. For example, the user class represents a real-world user. When this user logs into the system, their user object is used to create schedule and lesson objects, that relate to what they plan to learn in the real world. The primary disadvantage to the object model is the complexity and technical knowledge required to create them and accurately interpret them. In particularly complex systems, relationships can grow to be complicated and time consuming to properly identify and represent.

The true strength of these models is often found in combining them. Leveraging the insights into the process flows, interactions, and behaviors represented in the process model, and combining it with the structure and modular design of the object model, helps ensure we have an organized, maintainable, and scalable DriverPass system. Using both models to inform development ensures we have a clear and well documented architecture that also meets the client's needs.

## References

Valacich, J. S., & George, J. F. (2024). Modern Systems Analysis and Design (10th ed.). Pearson Education (US). <https://mbsdirect.vitalsource.com/books/9780138180294>