**Project One: Model Application – DriverPass**  
*Leonard Foy*  
*Southern New Hampshire University*  
*CS-255: System Analysis and Design*  
*October 2025*

A process model shows how data and actions move through a system to achieve results. It helps identify what happens first, what follows next, and what inputs or outputs are required. For the DriverPass project, a process modeling approach would focus on the sequence of events that occur when users interact with the system.

For example, a student would first create an account, log in, and then choose whether to take a practice test or schedule an on-the-road driving lesson. Each of these actions involves data moving between the user, the application, and the central database. A process model like a Data Flow Diagram (DFD) or Activity Diagram would visually show how the student’s request travels through the system and what processes handle that data.

Using a process model for DriverPass helps clarify the main functions the system must perform, such as verifying user credentials, updating schedules, and recording test scores. This approach also helps uncover inefficiencies or missing steps before development begins. It’s especially useful for communicating with non-technical stakeholders like management or instructors because the diagrams are easy to understand and show how data flows through the system in a logical way.

However, a process model focuses mainly on what happens to the data, not on the structure of the system itself. While it’s great for outlining workflows, it doesn’t always show how different parts of the system interact at the object level, such as how students, instructors, and lessons are connected in the database. That’s where object modeling comes in.

An object model focuses on the objects that make up a system, such as students, instructors, vehicles, and lessons, and how those objects interact. Each object has attributes (like a student’s name or test score) and behaviors (like scheduling a lesson or submitting a test). This method is perfect for designing systems like DriverPass that will later be developed using object-oriented programming languages such as Java.

For the DriverPass system, the object model would define the key components that make the platform work. Objects might include:

* Student – stores login information, test results, and lesson schedules.
* Instructor – manages available lesson times and updates completion records.
* Admin – oversees user accounts, updates DMV content, and generates reports.
* Lesson – tracks location, time, and instructor details for in-person sessions.
* Test – holds practice exam data and results linked to each student.

Using UML (Unified Modeling Language) class diagrams, each of these objects would be shown with their relationships like how a student object connects to multiple Lesson objects or how an instructor object interacts with the Schedule class.

The object modeling approach helps organize the system into smaller, reusable parts, making it easier to maintain and scale later. It also aligns closely with how the system will actually be coded. For instance, each class in the diagram could directly translate into a class in the final software, which makes development smoother.

**Process and Object Model Comparison**

Both modeling approaches have their strengths, and each contributes something valuable to the DriverPass system. The process model emphasizes the flow of information ideal for understanding the system’s logic and how users interact with it. It helps ensure that nothing is overlooked in the user journey, such as authentication steps or data updates.

On the other hand, the object model focuses on structure, making it better suited for long-term system maintenance and scalability. It provides a blueprint for developers to understand how each part of the program connects, ensuring that changes to one part don’t break the others.

For DriverPass, combining both models would create the strongest outcome. The process model could map out workflows like “student registers,” “student takes test,” and “instructor updates lesson,” while the object model defines the classes that make those workflows possible. Together, they give a complete picture: one showing how information moves through the system, and the other showing the components that make it happen.

The DriverPass system needs a design that’s both efficient and easy to maintain and using both modeling approaches supports that goal. The process model provides clarity for business users by showing how data moves from one point to another, while the object model gives developers the structure they need to build the system effectively.

In short, the process model helps the team understand how DriverPass operates from the user’s perspective, and the object model defines what makes those operations work behind the scenes. When used together, they ensure that the system will not only meet user needs but also remain adaptable as technology and DMV requirements evolve.

**References**

Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2020). *Systems analysis and design in a changing world* (8th ed.). Cengage Learning.

Southern New Hampshire University. (n.d.). *DriverPass interview transcript*. SNHU.