# CS 255 Model Application Short Paper

Brooke Slampak

brooke.slampak@snhu.edu

Southern New Hampshire University

## Process Model Application

In the DriverPass scenario, a process model would help visualize how the system handles everyday tasks and user interactions. This includes processes like student account registration, student payments, appointment scheduling, lesson cancellations, taking online tests, and password recovery. These workflows involve various user roles, such as customers, secretaries, instructors, and administrators, with each interacting with the system in different ways. Applying a process model to the system allows developers and stakeholders to understand how data moves through the system and how tasks are completed from start to finish (Dennis, Wixom, & Roth, 2021).

For example, the process model would help map out how a customer books a driving lesson through the website. It would include steps like selecting a package, choosing a time slot, assigning a driver, and payment. Other workflows such as driver assignments, generating reports, and user authentication would also be illustrated. Tools like data flow diagrams and activity diagrams are useful for representing these types of workflows (Sommerville, 2016).

Process modeling is especially helpful early in the project because it can highlight inefficiencies, missing steps, or potential issues on the users’ end. It allows stakeholders like Liam and Ian from DriverPass to see how their business needs will be supported by the system and where improvements can be made (Pressman & Maxim, 2015). For instance, if a cancellation process involves too many steps or lacks proper user notifications, it can be redesigned before implementation.

## Object Model Application

An object model would be applied to the DriverPass system by identifying and organizing the main objects involved in the business. These objects include Customers, Drivers, Cars, Lessons, Appointments, Packages, and Admin Users. Each of these objects would be defined by attributes and behaviors (Satzinger, Jackson, & Burd, 2020).

Using an object modeling approach helps organize the system’s structure, relationships, and logic. For instance, a Customer object would have a relationship with multiple Lesson objects, each linked to a Driver and a Car. A Package object might define how many hours of instruction are included, what content is offered in the package, and whether online testing is enabled. This structure ensures flexibility and reusability, which is important for a system like DriverPass that may expand later (IEEE Computer Society, 2014).

Object modeling also lays the groundwork for database design and application logic. It ensures data is logically grouped together and that relationships are clearly defined. For example, Liam expressed interest in modifying or disabling packages in the future. A well-structured object model would allow such changes without disrupting other parts of the system (DriverPass Interview Transcript, 2025). This supports long-term flexibility and maintainability (Sommerville, 2016).

## Process and Object Model Comparison

Both modeling approaches have their own strengths and limitations when applied to the DriverPass system. Process modeling is ideal for outlining workflows, especially from a user's point of view. It is very visual and easy for non-technical stakeholders to understand, which helps during early planning and approval stages. It also makes it easier to design user interfaces by breaking down tasks like appointment booking or lesson tracking into clear and manageable steps (Pressman & Maxim, 2015).

However, process modeling lacks depth when it comes to organizing parts of a system and data structures. It doesn’t explain how data is stored, reused, or connected, which is important for maintaining a scalable system. That’s where object modeling becomes important. Object modeling allows for clear specifications of a system’s parts and their interactions, making it easier to implement, maintain, and scale the system over time (Satzinger et al., 2020).

The drawback of object modeling is that it’s less effective at showing how tasks are performed step by step. It assumes developers understand how these actions are triggered and processed. Additionally, object models can be harder to follow for non-technical stakeholders, potentially causing confusion and a lack of understanding during planning (Dennis et al., 2021).

The best approach for DriverPass is to use both models together. Process modeling helps define the "how" of the system, like how users interact with it and how workflows operate, while object modeling defines the "what", such as what entities exist in the system and how they relate. Used together, these approaches would provide a complete, user-friendly, and technically sound plan for system development (Sommerville, 2016; IEEE Computer Society, 2014).

By starting with process modeling, the team can ensure that all user interactions are accounted for and optimized. Then, by developing an object model, the technical team can build a clean, logical system architecture that supports future updates and improvements. This strategy to employ both models ensures DriverPass is delivered a system that’s both easy to use and easy to manage in the long-term.

## References

* Dennis, A., Wixom, B. H., & Roth, R. M. (2021). *Systems analysis and design* (7th ed.). Wiley. https://soclibrary.futa.edu.ng/books/Systems%20Analysis%20and%20Design,%208e%20(Alan%20Dennis,%20Barbara%20Wixom,%20Roberta%20M.%20Roth)%20(Z-Library).pdf
* DriverPass Interview Transcript. (2025). Internal project transcript for CS 255 consulting project.
* IEEE Computer Society. (2014). *Guide to the software engineering body of knowledge (SWEBOK)* (Version 3.0). <https://swebokwiki.org>
* Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner’s approach* (8th ed.). McGraw-Hill Education. https://intranetssn.github.io/www.ssn.net/twiki/pub/CseIntranet/CseBCS6403/PressmanBook.pdf
* Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2020). *Systems analysis and design in a changing world* (8th ed.). Cengage Learning. https://mygust.com/uploads/BOOK-Systems\_analysis\_and\_design\_in\_a\_changin.pdf
* Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson. https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf