# CS 255 Model Application Short Paper

Mike Brown

Michael.Brown72@snhu.edu

Southern New Hampshire University

## Process Model Application

A process model shows the flow of steps that people follow when they use a system. In the DriverPass project, this approach would outline how students, administrators, instructors, and IT staff interact with the system. For example, one process could begin when a student logs into their account, selects a training package, and schedules a driving lesson. The system would then check availability, assign a driver and vehicle, confirm the booking, and log the details. Another process might be for practice tests, where the student chooses a test, completes it, and immediately receives feedback.

Using a process model helps ensure that no important steps are left out. Each process can be broken into smaller actions such as “log in,” “choose package,” “check DMV updates,” and “confirm reservation.” This makes it easier for stakeholders, such as DriverPass management, to see how the workflows will function (Dennis, Wixom, & Tegarden, 2015). However, a drawback is that process models can become complicated when covering exceptions like cancellations or reschedules. Even with that challenge, they are a useful tool for capturing the big picture of how the system will operate.

## Object Model Application

An object model takes a different perspective by focusing on the components inside the system. Objects represent real-world items that the system manages, such as User, Lesson, Car, Driver, Reservation, and PracticeTest. Each object contains attributes that store information and methods that describe its behaviors.

For instance, the Reservation object could include attributes like date, time, driver, and vehicle. Its behaviors might include createReservation(), cancelReservation(), and modifyReservation(). A PracticeTest object would include attributes such as test questions, student score, and completion status, along with behaviors to takeTest() and gradeTest(). These objects also have relationships; for example, a User may have multiple reservations, and each reservation links to one driver and one car.

The benefit of object modeling is that it matches how the system will eventually be coded. It also makes it easier to organize the system into reusable parts and to support role-based access. For example, the User object could be specialized into subtypes such as Customer, Instructor, and Administrator. A limitation is that this approach can be less intuitive for stakeholders without a technical background because it focuses on structure rather than the sequence of steps (Dennis et al., 2015).

## Process and Object Model Comparison

Both process and object modeling approaches are valuable for the DriverPass system, but they serve different purposes.

* Advantages of process models: They are simple for stakeholders to understand, show the order of activities, and help confirm that all steps are included.
* Disadvantages of process models: They can become complicated with many scenarios, and they do not show how the system is built internally.
* Advantages of object models: They provide a blueprint for developers, capture the structure of the system, and make it easier to manage user roles and permissions.
* Disadvantages of object models: They are more technical and can be harder for non-technical users to interpret, and they do not highlight the order of actions as clearly as process models.

For the DriverPass project, the best solution is to use both. The process model ensures that every workflow, such as scheduling lessons or taking practice tests, is mapped and verified. The object model ensures that the system’s structure, including users, reservations, and tests, can support these workflows. Together, they provide a complete design that addresses both business needs and technical implementation (Dennis et al., 2015).

## References

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *Systems analysis and design with UML* (4th ed.). Wiley.