# CS 255 Model Application Short Paper Tyler Ellis

Tyler.Ellis1@snhu.edu

Southern New Hampshire University

## Process Model Application

[How would you apply a process model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.]  
  
 In applying a process model to the DriverPass scenario I would begin by identifying the key business processes that drive the system. The first process is customer registration and account management. In this process a user provides personal information such as first name, last name, address, phone number and payment details. The system then validates the data and creates an account that will be used to schedule driving lessons and access practice tests. This initial flow is critical because it sets the stage for every subsequent transaction within the system.

Following registration, the next process is the reservation and scheduling of driving lessons. Customers select from three available training packages and specify their preferred lesson times. Each driving lesson is two hours long and may be distributed across multiple sessions depending on the chosen package. The process model details the steps a customer follows to schedule a lesson online or via direct contact with the secretary. It also outlines the modifications and cancellation flows which include notifying the system of any changes and recording the user responsible for the change. This flow ensures that every booking is properly tracked, and that the system maintains a clear audit trail.

Another key process is the integration with external systems such as the department of motor vehicles. The DriverPass system must continuously update its content to reflect current rules policies and sample test questions. The process model accounts for receiving updates from the external source and notifying system administrators when the latest information is available. This process ensures that the training material remains relevant, and that the system can adapt quickly to any regulatory changes.

Finally, the process model captures error handling and administrative interventions. For instance, if a customer fails to log in after several attempts the system initiates a security protocol that locks the account and alerts the IT officer. In addition, the process for password reset is clearly defined so that customers can regain access securely. Each of these processes is mapped out in the model to provide a comprehensive view of the dynamic interactions within the system. This approach aids in identifying potential bottlenecks and ensures that the system flow is both efficient and user friendly.

## Object Model Application

[How would you apply an object model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.]  
  
 The object model for the DriverPass scenario focuses on the static structure of the system by identifying the classes and the relationships between them. The central object in the design is the Customer class. This class encapsulates information such as name address contact details and login credentials. The Customer class interacts with several other classes including Reservation and Package. The Reservation class stores details of each driving lesson such as lesson time assigned driver and vehicle information. By clearly defining these classes the system can manage customer data and scheduling information in a structured manner.

In addition to the Customer and Reservation classes, the object model includes an employee class that serves as the parent for more specialized roles like Secretary and IT Administrator. These subclasses inherit common attributes like login credentials and personal information while also incorporating unique responsibilities. For example, the IT Administrator class includes methods for password resets and system maintenance tasks while the Secretary class focuses on appointment scheduling and customer record updates. This hierarchical structure enhances the reusability of code and simplifies the management of user roles within the system.

Another vital component in the object model is the Package class which defines the three training packages offered by DriverPass. The Package class stores attributes such as training duration and details of additional services provided. It also defines methods that allow the system to enable or disable packages without requiring code changes. The integration with the department of motor vehicles is modeled through an Update class that manages incoming data regarding changes in driving test regulations. This class works in tandem with the Reporting class that generates activity reports and logs changes made by users. By using these classes, the design ensures that data is organized and that the relationships between entities are clear and maintainable.

The object model also emphasizes the relationships between classes through associations and aggregations. For instance, a customer may have multiple Reservations, and each Reservation is associated with one specific Package. Similarly, a Vehicle is linked with a Driver and is part of the overall training process. By explicitly modeling these relationships the system becomes easier to understand from a development perspective. The static view provided by the object model lays a solid foundation for implementing the database schema and the business logic. It ensures that all entities and their interactions are clearly defined and that future modifications can be handled with minimal disruption to the overall design.

## Process and Object Model Comparison

[What are the advantages of each model for the DriverPass scenario? What are the disadvantages of each model for the DriverPass scenario]  
  
 The process model and the object model offer different but complementary perspectives for designing the DriverPass system. The process model excels at illustrating the dynamic behavior of the system. It captures the sequence of actions that occur when a customer registers for an account or schedules a driving lesson. This model clearly depicts decision points such as validating user credentials and handling password resets. One of the process model's advantages is that it provides an easy-to-understand flow of events accessible to both technical and nontechnical stakeholders. It simplifies the explanation of how the system operates in real time and highlights the user interactions that drive the business.

In contrast, the object model provides a static view essential for understanding the data structure and relationships between various entities. It focuses on defining the classes such as Customer Reservation Package and Employee and clarifies how these classes interact. The strength of the object model lies in its ability to outline the fundamental building blocks of the system. This model is especially beneficial during the coding phase when developers require a clear blueprint for creating the database schema and implementing the underlying business logic. It is well suited for tasks that involve data manipulation and system integrity.

Despite their individual strengths each model also has limitations when applied in isolation to the DriverPass scenario. The process model may become complex when accounting for every branch and decision point in a system that handles multiple user roles and external integrations. It does not provide a detailed view of how data is stored or organized which can lead to gaps during the implementation phase. However, the object model does not capture the flow of user interactions or the sequence of events. It presents a static picture that does not reflect the dynamic nature of scheduling appointments and handling real time updates from the department of motor vehicles.

For the DriverPass project both models play a key role and complement each other. The process model is ideal for mapping out the user journeys and operational workflows such as booking lessons and managing account security. In contrast the object model lays out the structure required to support these processes by defining clear classes and relationships. Using both models together creates a comprehensive design that addresses both the dynamic and static aspects of the system. This integrated approach ensures that all client needs are met while also providing a solid foundation for future expansion and system maintenance.

The complementary nature of these models becomes evident when considering the evolution of the DriverPass system. The process model allows the design team to quickly identify and resolve potential issues in the user interaction flows while the object model provides clarity for database design and business logic implementation. By iterating between the two models during the design phase the team can achieve a balanced system that is both user friendly and structurally sound. In the end, this dual approach results in a design well aligned with the client requirements and adaptable to future enhancements.

## References

* Hoffer, J. A., George, J. F., & Valacich, J. S. (2025). *Modern systems analysis and design* (10th ed.). Pearson.