# 

# CS 255 Model Application Short Paper

James Lee

James.lee4@snhu.edu

Southern New Hampshire University

## Process Model Application

Process model is used to represent “the functions, or processes, that capture, manipulate, store, and distribute data between a system and its environment and between components within a system” (Valacich et al., 2019, Chapter 7). One form of process model, such as the Data Flow Diagramming (DFD), can be used to illustrate the flow of data within a system. Understanding the flow of data within a system helps with understanding the process flow, elements, and the behaviors of the system.

For the DriverPass system, the context diagram of the DFD would depict the DriverPass process as the single process, with Customer, Administrator, Secretary, and Owner as the four sources. For the Customer, the login credential, payment, and driving lesson scheduling data would flow from the source to the process, and the receipt/confirmation data would flow from the process to the Customer source. For the Administrator, the user profile update data, password reset data, and system maintenance data would flow in to the process, and the system activity update report data would flow from the process to the source. For the Secretary, the customer information data would flow in to the process, and the confirmation data would flow from the process to the source. Finally, for the Owner, the system report data would flow from the process to the Owner source.

For the Level-0 DFD, the DriverPass process would be decomposed to include system login process, driving lesson package purchase process, driving lesson appointment scheduling process, online classes & quizzes process, update system & accounts process, and produce system report process. Each of these six processes would be further decomposed until each process contains only one operation. These processes would interact with various data stores; individual database for user credentials, driving lesson package purchase/payment, driving lesson scheduling, driving lesson logs, online classes and quizzes, and system backup.

## Object Model Application

Object modeling allows us to visually represent components of the system, along with their domains, behaviors, states, and characteristics in a way that can be easily understood. The various components of the DriverPass system that would be represented as objects are:

* Users
  + Customer object inherits from Users
  + Administrator object inherits from Users
  + Secretary object inherits from Users
  + Owner object inherits from Users
* Shopping Cart
  + For customers to purchase driving lesson package(s) through the system
* Driving Lesson Appointments
  + For either the customers or the secretary to create for each customer. Each customer can have zero to many associated appointments.
* System Report
  + Associated with Owner object
* Databases
  + Inheritance database objects are:
    - User credentials
    - Driving lesson package purchase/payment information
    - Driving lesson scheduling
    - Driving lesson logs
    - Online classes and quizzes
    - System backup

## Process and Object Model Comparison

For the DriverPass scenario, process modeling allows for representation of system components by their flow of data and functions. With process modeling, we can easily see the functions of the system and the movement of required data throughout the system. With the object modeling, we can easily identify the object components of the DriverPass system, and their relationships and interaction with each other.

While each of the two models can depict certain aspects of the system’s functionalities, each model does not depict the system as a whole. For instance, the object model would not depict how the administrators and the owner would receive notifications on system processes such as a potential hacking attempt using a user’s login credentials. The process model would not depict the various object components of the system such as the various states of users (customers, administrators, secretary, owner) and their interaction with the various states of the system. Because of these limitations, both the process model and the object model are required to fully depict the system as a whole.

## References

Valacich, J., & George, J. (2019). *Modern Systems Analysis and Design*. Pearson.