# CS 255 Model Application Short Paper

Eric Wallace

eric.wallace3@snhu.edu

Southern New Hampshire University

## Process Model Application

To determining all system requirements for the DriverPass application, I will apply process modeling by using use data flow diagrams (DFD) to gain a better understanding of how data flows throughout the system.

Due to the complexity and all the various processes taking place on the DriverPass application I would break the design into multiple process models which will make is somewhat easier to understand how data flows and the logic.

I would start with the customer diagram since most of the data involves customers in one way or another. To start the flow of data will fork into two different directions depending on the type of interaction being provided by the customer. The two forks could be online and telephone, because new customers must call in to get their information setup in the system and existing customers can also call in to schedule appointments or purchase a package.

The telephone fork might have a process where they call into DriverPass to setup an account and provides the employee with the information, the data then flows from the employee to a process to add the customers information to the system. There could be another flow of data that goes to a process that creates, changes, or cancels an appointment which could then lead to the notification process, which would notify the driver selected for the appointment and the customer.

The online fork is much more complex as it has a lot more functionality, I won’t go into much detail other than outlining the processes comprised of this fork. Customers would first supply data to the authentication/authorization process to get them access to the system, from there users could access processes related to the learning system, online appointment system, account processes.

Employees would access most of the same processes and functions used by customers, I won’t outline them in this section other than the only exception is the reporting processes which the owner needs access too, those processes are only accessible to him.

## Object Model Application

To gain knowledge of all requirements on the DriverPass application, I will apply an object model. Object models allow us to visualize the objects contained within a system, for the DriverPass application the object model I will use is the UML diagram. Unified Modeling Language (UML) is a graphical language that provides a standard way to depict how software should be written. UML diagrams allow us to show all the objects of a system, the methods and attributes objects have and to show how different objects are related to other objects and how they can interact with one another.

To start I would create a base user class that would contain the properties shared by all users of the system and methods for interacting with the object. A user on the system would have a unique username, password and role properties which would be used by the authentication system for gaining access to the application and roles would be for authorization which specifies what components of the system a user has access to. There are two types of user objects which will be derived from the user class, employee, and customer. The customer class would have another subclass associated with it, the account class or object is the object which stores the detailed information about the customer such as shipping and billing address, phone number and credit card information.

Other objects on the DriverPass application could include but not be limited to an appointment, report, class, test, instructor, vehicle, and package. Each would have its own properties and methods; through the UML diagram we would show how they are related or associated with one another. Following the guidelines of UML, the diagrams we create would be suffice for developing a complete application.

## Process and Object Model Comparison

From the reading above we know how each model visually represents how the application interacts with data. Below is a table of the advantages and disadvantages of both.

|  |  |  |  |
| --- | --- | --- | --- |
| **Comparison Of Object And Process Models** | | | |
| **Object Models** | | **Process Models** | |
| **Advantages** | **Disadvantages** | **Advantages** | **Disadvantages** |
| Reduced Maintenance | Steep learning curve | Easy to understand by clients | Extremely time consuming |
| Real-World Modeling | Difficult for clients to understand | Support data logic | Does not provide a complete view of a system |
| Improved Reliability and Flexibility | Do not show the specifics about the process | Diverse and can be applied to almost any system | Operate within finite processes that have a start and a finish  Not very dynamic |
| High Code Reusability |  |  |  |

Looking at the list above one can start to see a trend, Object models are very favorable to development teams whereas Process models are more beneficial to clients. While Object models have a steep learning curve the real-world modeling and code reusability are major advantages. On the other hand, Process models support data models and are very diverse, the major downside is they an extremely long time to create. If I had to choose one model it would have to be Object models and for two reasons: Object orientation is the way of the future, almost all modern programming languages are OOP based. Finally, the extremely large amounts of time required to create DFDs makes process models less attractive. In today’s world time is money and software development are moving towards streamlining the development process, reducing time it takes to develop software and being modular, so you don’t have to do the same thing twice.

## References

[You were **not** requiredto use external resources for this report. If you did not use any resources, you should remove this entire section. However, if you did use any resources, including the textbook, to help you with your interpretation, you **must** cite them. Use proper APA format for citations. Two samples have been included below for you to use as a reference.

For a book:

Author, A. A. (Year of publication). *Title of work: Capital letter also for subtitle*. Location: Publisher.

For a website:

Author, A. A. (Year, Month Day). Title of article. *Title of Website*. Retrieved from  
https://www.homeaddress.com/