

CS 255 Model Application Short Paper

Adam Resemius

Adam.resemius@snhu.edu

Southern New Hampshire University

Process Model Application

Mapping processes by making a process model are a great way to show the lifecycle of a system to get a thorough idea of a lifecycle from start to finish. This provides a good idea how the system should work together with different aspects and how each process ties into one another.

I would apply a process model to the DriverPass scenario to map the processes that will happen from start to finish from the registration process to the student passing his drivers test. This will map the lifecycle a typical customer would go through with the usage of this application. I would first start by making a registration process, then tie that into a login authentication that would allow the user to access the learning portal. From here the user will be able to view content from the frontend while the backend processes tie in a booking system, updating system, and tracking system for the user's processes. This process will continue until the subscription is over, or until the user passes their drivers test at the DMV, then the students account will become inactive but still be stored in a database for a certain amount of time.

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.]

Object models are used to provide how objects relate to one another. These models are crucial because they often include object-oriented principles that developers can use and save a lot of time, because it shows developers how objects relate to one another.

I would apply an object model in DriverPass by creating a user object and relate with inheritance to child classes of Administrator, Instructor, or Student. These objects will get

different functionality to share between other objects and choose how they manipulate them. This helps encapsulate data and ensure that information stays protected and secure.

Objects that they would be allowed to manipulate would be the DriverPass webpage that is created. They can update account information in the Account class. They will be able to view/edit/delete information based on their roles and responsibilities given to them by each classification. Students will be able to access the Quizzes class but only edit where the prompted answers will be. Instructors can edit/update/delete the Quizzes that they create, and Admins can edit/update/delete any quiz in the system. Privileges like these would continue onward to the Calendar classes and Grades classes. Data will be shown and hidden by each user with the help of abstraction. During the creation process of Quizzes, Instructors and Administrators can use the Templates (builders) that will create new Quizzes based off their preferences. This would be explained as polymorphism in object-oriented programming.

Process and Object Model Comparison

Process models and Object models are a very helpful step to system analysis as they are used to describe relationships between pieces of a system. Process models have many advantages like showing a start to end structure of a lifecycle, show how processes are tied together, what steps need to be completed before continuing to a sequence, and allow test cases to be created to show where issues lay in a process model. The main disadvantage of a process model is it doesn't tie front end processes and backend processes together very well, making users create multiple diagrams to see both sides. Process models also doesn't explain in-depth functionality.

Object models are great for showing functionality in a system and show how each object relates to one another in a network. They also show object-oriented principles and help shorten development lifecycles as developers can follow the "Do Not Repeat Yourself" and "Keep It

Simple Stupid” methods. This helps with locating bugs in objects and ensures that each object being developed builds off of previous objects.

Disadvantages that an object model has, is it is hard to see a start to end process with how objects interact with one another. This is crucial because an end user should be able to manipulate objects in a system in order to reach a desired output. Object models don’t show this well.

If I were to interpret these models into DriverPass I would most likely create my process model first and then create an object model. Since I know my desired outcome for a system, I can now plan objects and functionality to workaround and reach that end goal. This ensures that I don’t miss any objects for a system.