# CS 255 Model Application Short Paper

Joseph Farrish

joseph\_a\_f@hotmail.com

Southern New Hampshire University

## Process Model Application

From the analysis phase, there are two sub-phases, requirement determination, and requirement structuring. From these phases, processes are identified we build upon this identification by creating a structured process model that will meet the customer requirements. Doing so involves the construction of an activity diagram and use-case diagrams both of which need to depict how business systems operate.

From the driver pass scenario, analysts have gathered requirements from the users. These requirements have been classified as either non-function or functional requirements.

|  |  |
| --- | --- |
| **Non-Functional Requirements.** | **Functional Requirements** |
| System needs to provide on/off-line access | User activity tracking. |
| Client needs full access over website. | Reservation via Web/Phone. |
| Support for user types - i.e., Admin, User, etc. | Teacher and Student ID. |
| System needs to be cloud based. | Customer Registration needs specific info. |
| Accessible via web page. | Interface with DMV. |
| Client provided GUI design. |  |
| Teacher input. |  |

With this information, we can begin to construct the use-case and activity diagrams. Since the goal of the use-case diagram illustrates the activities performed by the users of the system. Both functional and non-functional requirements can be used to construct a functional view that depicts how the user will use the system. The sequence of actions should be used; place and draw use cases, place, and draw actors, draw subject boundaries, and then add associations can be used as a guideline for constructing a use-case diagram. It is important to remember there are guidelines to follow which I will not list but will mention that when constructing a use-case diagram you must conform to the guidelines if you are to apply them correctly. Further, you must conform to the proper use of the diagram elements of a use case diagram.

Since the goal of the activity diagram is to illustrate the processes, activities, and how objects or data move among them we can use the functional and non-functional requirements to begin the construction of the activity diagram. It is important to note as an analyst you will be using a particular set of elements and conforming to a set of guidelines just as is the case with a use-case diagram. These elements and guidelines are considerably more complex. A sequence of actions should be taken when working with activity diagrams; Choose a business process, identify activities, identify control flows and nodes, identify object flows and nodes, layout, and draw a diagram (Denis et al., 2012 Ch. 4).

**Object Model Application**

In the creation of an Object model for the driver pass, I would begin to develop a conceptual diagram to illustrate the overall class hierarchy necessary to meet the customer requirements. A conceptual diagram represents the concepts in the domain. Once the overall conceptual diagram is complete, we can begin building upon it by adding specifications to the diagram. Here we focus on the interfaces of abstract data types in our design. Finally, the implementation phases will describe how classes will implement their interfaces (UML Class Diagram Tutorial, n.d.-b).

In the conceptual phase, we would simply add a name for each object. For example, using inheritance, I would build a base user object, and from that object, I would create an object called User, Admin, Employee, and Teacher. As the project approaches specification and implementation phases**,** analysts using polymorphismwould add base functionality to the root object and add unique functionality to higher-order objects.

## Additional objects involved might include appointments, appointment lists, lessons, tests, test lists, and authentication services.

## When considering security in the overall design and during the implementation phase it would be wise to structure objects according to the design pattern that we might use. For instance, if wish to use a singleton or domain design pattern. This might affect the number of instances of objects that can exist within the system and this concern would warrant adding the appropriate cardinality notation to object connections (UML Class Diagram Tutorial, n.d.-b).

## Process and Object Model Comparison

The application of the process model is capturing how the system will be used from the perspective of the user. Process models capture the sequence of events that need to take place and how the user will ultimately carry them out. However, a process model does not emphasize the object instance needed to perform these functionalities.

The object model unconventionally described represents the system more from the perspective of the system developer. It shows the objects involved in the system during the conceptual, specification and implementation sub-phases. While it can communicate processes involved in a system through attributes it doesn’t do so in the sequential and visual manner often desired from a process model.

**Conclusion.**

Process and Object models have their respective applications, and both are necessary elements in the software development lifecycle. Both require iterative and analytical efforts to achieve a state of completeness. To contradict this point, they are often too neat and don’t represent the real world where logic and order are often messy in which case a complete model may not be desirable or necessary.

**References**

Joey F. George, Joseph Valacich (2016). Modern Systems Analysis and Design. Pearson; 8th edition.

Alan Dennis, Barbara Haley Wixom, David Tegarden (2012). System Analysis and Design with UML, 4th Edition. https://learning.oreilly.com/library/view/systems-analysis-and/9781118037423/

UML Class Diagram Tutorial. (n.d.-b). Visual Paradigm https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/