CS 255 Model Application Short Paper

Kain Mason  
k.mason@snhu.edu  
Southern New Hampshire University

Process Model Application

In the DriverPass scenario, a process model would be used to design the system by mapping out each task in the workflow for scheduling driving lessons, managing user accounts, and tracking test results. Process modeling helps visualize the series of tasks that occur from start to finish for every action a user can perform on the system. For instance, the process of booking a driving lesson begins when a student logs into their account, selects a time slot, and confirms the booking. The system would then handle tasks like notifying the student, instructor, and administrator of the reservation and updating the schedules accordingly.

A process model for DriverPass would ensure that no important steps are missed during the workflow. For example, it could prevent overbooking of instructors or vehicles by ensuring that each task involved in booking a lesson occurs in a specific order. It would also help document how users interact with the system, ensuring that the user experience is consistent and efficient. However, the downside of this approach is that it focuses heavily on workflows and tasks, making it more difficult to adjust the system if there are significant changes, such as adding new lesson packages or modifying existing workflows (Dumas et al., 2018).

Object Model Application

An object model for the DriverPass system would focus on defining the different entities involved, such as "Student", "Driver", "Lesson", "Instructor", and "Test", and the relationships between them. Each object would represent real-world entities and would contain data and behaviors relevant to the system. For example, the "Student" object would store personal details, such as name, contact information, and test history. The "Lesson" object would store details about the scheduled driving lesson, such as the date, time, assigned instructor, and vehicle.

In an object-oriented approach, the relationships between these objects allow the system to handle complex interactions between entities in a flexible way. For instance, a "Lesson" object would be associated with a "Student", a "Driver", and a "Vehicle". If DriverPass wanted to add or modify lesson packages, this could be done by updating the relevant objects without redesigning the entire system (Larman, 2021). However, the downside is that object modeling doesn’t always clearly capture the full user experience or workflow, as it is focused more on data and relationships rather than specific tasks or processes.

Process and Object Model Comparison

When comparing the process and object models for DriverPass, both have distinct advantages and disadvantages. The process model provides a clear visualization of user workflows, which is beneficial for understanding how tasks are performed within the system. This model helps ensure that no steps are missed and can identify bottlenecks or inefficiencies in the process. For example, it can make sure that lesson bookings are confirmed, notifications are sent, and schedules are updated in the correct order (Dumas et al., 2018). However, it can be more rigid when it comes to implementing changes in the system structure, as workflows would need to be adjusted to accommodate any new features.

In contrast, the object model is more flexible, allowing the system to be easily updated or expanded. By focusing on the objects and their interactions, the system can grow with minimal disruption. For example, if DriverPass wanted to add new training packages, the system could be updated by simply adding new properties or methods to the relevant objects without affecting the overall workflow (Larman, 2021). The downside of object modeling is that it doesn’t provide a clear view of the process flow, which can make it harder to visualize how different users will interact with the system.

Ultimately, a combination of both models could provide the best solution for DriverPass. The process model ensures that workflows are optimized for user experience, while the object model provides the flexibility needed to adapt the system to future requirements.

References

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). Fundamentals of business process management (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-56509-4

Larman, C. (2021). Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development (3rd ed.). Pearson.