**CS 255 Model Application Short Paper**

Jaden B. Knutson

Ja-knut@hotmail.com

Southern New Hampshire University

**Introduction**

Software development is a complex, multifaceted task. It requires a well-structured approach to ensure productivity and risk management in order to deliver high-quality products. The key to this approach is tools like process and object models. Process and object models serve as tools in system analysis and design. These models serve as guides for developers to understand system workflows, interactions, and behaviors. They also provide an outline of the procedures that a system should follow to achieve the desired outcombae.

In the following, I provide analysis and application of these models in the context of the DriverPass scenario – an interactive platform designed to aid students in studying for their driving tests and offering flexible session packages.

**Process Model Application**

In software development, process models provide a strategic roadmap for the organization and project approach. They divide the development process into distinct stages providing a high-level view of the system's behaviors and interactions. This helps align workflows with the business’s requirements and goals. Process models facilitate focused, goal-oriented development processes by ensuring each development stage is purposeful and aligned with project objectives.

To put the process model into practice in DriverPass, we would identify and outline workflows that manage the unique interactions of different system users like students, instructors, and administrators. Through a series of step-by-step developments, the process model ensures completeness, consistency, and timeliness.

**Object Model Application**

Object models focus on individual components called "objects." These objects contain data and behaviors forming the system's fundamental building blocks. This approach treats the system as a complex network, allowing developers to analyze interactions and facilitate problem-solving.

In developing software for DriverPass, the object model would be a powerful tool for system design, identifying system objects, and outlining their behaviors. It includes mechanisms to handle unexpected situations which will ensure system flexibility. This approach enhances the user experience by providing a reliable and user-friendly platform that would be applicable to the DriverPass scenario.

**Process and Object Model Comparison**

In the DriverPass scenario, both the process model and the object model have their unique advantages and disadvantages. The process model:

* offers flexibility, allowing for changes and refinements to accommodate evolving requirements and user feedback.
* facilitates early feedback and validation by involving stakeholders and users ensuring the final product aligns with user needs.
* promotes risk reduction by facilitating early identification and mitigation of issues.

However, without proper control, the process model can lead to scope creep, where new requirements continuously expand the project beyond its initial objectives.

Additionally, managing multiple iterations and tracking changes can introduce complexity.

On the other hand, an object model:

* promotes modularity and reusability by combining data and behavior within objects. This approach allows for easier maintenance and extension of the system.
* provides a clear representation of entities and their relationships, thus aiding in understanding the system's structure and behavior.
* encourages stronger code organization, making the codebase more readable and maintainable.

To be considered, the object model may require a learning curve for developers unfamiliar with object-oriented programming. There's also a risk of overengineering the system, introducing excessive abstraction and complexity.

**Conclusion:**

In conclusion, the process model and the object model both provide valuable tools for system design and development along with their own unique strengths and disadvantages. By leveraging the strengths of both models, it is possible to create a system that is reliable, efficient, and user-friendly.

In the case of DriverPass, where there is a need for speedy development and a clear representation of user interactions, the object model may be more beneficial. Its focus on object interactions and behaviors can expedite development time. However, it is important to maintain a balanced approach and utilize both models where appropriate. This ensures that the system is not only efficient but also flexible enough to adapt to future changes and enhancements. By combining the process model's comprehensive visualization of the system with the object model's detailed perspective on individual components, the resulting system is more likely to be reliably sound, flexible, and able to meet the evolving needs of the users.

In summary, a thoughtful, balanced approach incorporating both the process and object models is key to the success of projects like DriverPass, as it ensures an efficient, flexible system tailored to user needs.

**References**

Valacich, Joseph S, and Joey F George. *Modern Systems Analysis and Design*. 9th ed., Hoboken, Nj, Pearson Education, Inc, 2020.

Indeed Editorial Team. (2023, February 3). *What is a Software Process Model? 7 models you should know*. Indeed. https://www.indeed.com/career-advice/career-development/what-is-software-process-model

Khifer, Zen. “What Are the Differences between an Object and a Process Model?” *Brogramo.com*, brogramo.com/what-are-the-differences-between-an-object-and-a-process-model/?fbclid=IwAR008pCEcLL-Ypmb78lYLn7EstNGOOyEPAvjDJogJ9fk1YH3dZRGhGAMW\_4. Accessed 2] June 2023.

“Object Model.” *Wikipedia*, 13 Apr. 2023, en.wikipedia.org/wiki/Object\_model?fbclid=IwAR04kEuscRQK0nM6A-eyLDrGxby9rirTERzlTXOjye2xgu3fdb\_IERNITW0. Accessed 5 June 2023.