

Object Oriented Software Engineering

Amal Byju & Edwin Thomas

Roll number:16CO205,16CO218

National Institute of Technology, Karnataka Surathkal,India

amalbyju98@gmail.com, edwinthomas444@gmail.com

Abstract

Software Engineering has become an integral and inevitable necessity in today's world. From simplest elementary software like a simple text editor to modern sophisticated rocket launching software using number crunching algorithms, software has established the foundation for technology and development. Hence, this also calls for the need to simplify the conventional approaches of software design and implementation. Object Oriented Software Engineering (OOSE) is targeted to simplify the development process and to produce high quality maintainable software. A maintainable software may reduce the maintenance cost and high quality may enhance the sustainability of the software. This paper implementation focuses on usage of the OOSE concept in the development of the Trip Advising Software.

Keywords

Object oriented software engineering (OOSE), Object Oriented Programming (OOP), Object Oriented Design (OOD), Object Oriented Analysis (OOA).

1. Introduction

With the rapidly developing software Engineering domain, the use of structural model for software development have been overpowered by the object oriented software engineering concept.

The object oriented software engineering is a software design technique that is used in software design in object oriented programming. OOSE is developed by Ivar Jacobson in 1992. It is the first software engineering methodology that employs use cases in software design. It included requirements, analysis, design, implementation and a testing model.

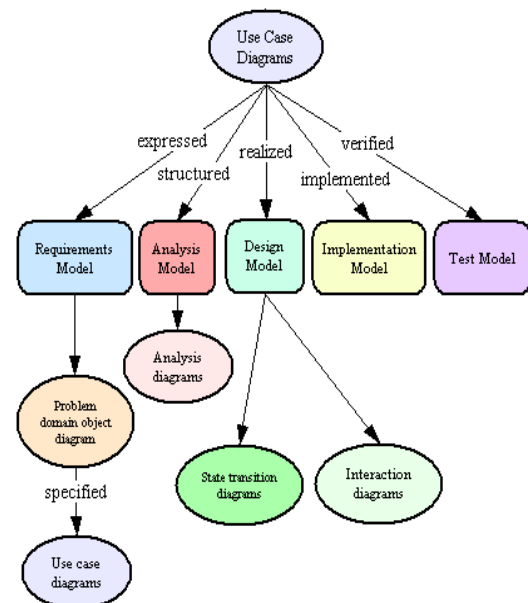


Figure 1. Object-Oriented Software Engineering.

Usage of the models mentioned above with respect to the Trip Advising Software Project:

1) Requirement Model:

It uses a combination of text and diagrammatic forms to depict the requirements in a way relatively easy to understand. Scenario based models, Class oriented models, data models, flow oriented models and data models can be used for the requirements phase of this project.

2) Analysis Model:

This model can be used to provide a logical model of the system in terms of classes and relationships.

3) Design Model:

This model takes requirements specifications relating the environment, use case descriptions, class diagrams and produces sequence diagrams and state charts.

4)Implementation Model:

This phase of the software engineering is done using object oriented programming concepts.

Drawbacks:

We might encounter such situations like having multiple objects sharing a common behaviour (method) in a program. In such situations, typing the method repeatedly for each class is not practical. Moreover, modifying this function means changing the function code in all of the classes.

To solve this problem, a possible approach that can be considered is to modularize the common function as a separate object and set relationships between the existing objects with this new object.

From the design and analysis point of view of the program using OOAD, managers and developers often assume that OOAD will save the from bottlenecks during the development process and will provide modularity in the architectural design. Another reason why OOAD is adopted is because of the promise of increased productivity coupled with a shortened development schedule. However, a sufficient amount of development time is often not allowed that will make sure the design processes are followed correctly. This may actually result in bigger problems. These include missed deadlines, schedule slippage, and project failures.

Conclusion and Future works

Object Oriented software engineering is still practiced widely. It does provide benefits like low maintenance costs and modularity. Nevertheless, there is scope for improvement. The first challenge can be overcome by adapting to an intuitive new approach towards organising program code. Modularity and structure of the program code should be preserved. Moreover, the code should be coherent. A more optimized method of dividing time for each stage of the SDLC should be developed.

References

[1] Farhad Soleimanian Gharehchopogh, Saman Jodati Gourabi, Isa Maleki "Object Oriented Software Engineering Models in Software Industry" , International

Journal of Computer Applications (0975 – 8887) Volume 95– No.3, June 2014

[2] Dr. Waralak V. Siricharoen "Ontologies and Object models in Object Oriented Software Engineering", IAENG International Journal of Computer Science, 33:1, IJCS_33_1_4

[3] Iqbaldeep Kaur, 2Navneet Kaur, 3Amandeep Ummat, 4Jaspreet Kaur, 5Navjot Kaur, " Research Paper on Object Oriented Software Engineering"JCST Vol. 7, Issue 4, Oct - Dec 2016.

[4] Ashwin Urdhwaresh " Object-Oriented Programming and its Concepts" International Journal of Innovation and Scientific Research ISSN 2351-8014 Vol. 26 No. 1 Aug. 2016, pp. 1-6

[5] Sunil T. D. and Dr. M. Z. Kurian, " A METHODOLOGY TO EVALUATE OBJECT ORIENTED SOFTWARE SYSTEMS USING CHANGE REQUIREMENT TRACEABILITY BASED ON IMPACT ANALYSIS", International Journal of Software Engineering & Applications (IJSEA), Vol.5, No.3, May 2014

[6] Pramod Mathew Jacob, Muhammed Ilyas H, Joyce Jose and Josna Jose, "A comparative analysis of structured and object-oriented programming methods," International Conference on Information Science (ICIS). IEEE, 2016.

[7] Dr. Vipin Saxena and Ajay Pratap "Transformation of UML Class Diagram for Object Oriented Database System", International Journal of Computer Applications (0975 – 8887), Volume 26– No.10, July 2011.

[8] Dr. Preeti Gulia, Dr. Amita Dev and Shrikant Patel, "Comparative Analysis of Object Oriented Programming and Aspect Oriented Programming Approach " In 2015 2nd International Conference on Computing for Sustainable Global Development (pp. 1836-1842). IEEE, 2015.

[9] Ye Peilei, "A object-oriented development process and UML modeling tools", IITA International Conference on Services Science, Management and Engineering, pp. 225-228. IEEE, 2009.

[10] S. Clarke, and R. J. Walker, "Towards a Standard Design Language for AOSD", In Proceedings of 1st International Conference on Aspect-Oriented Software Development (AOSD), 2002.