Object Oriented Software Engineering Models in Software Industry

Farhad Soleimanian Gharehchopogh
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, West Azerbaijan, IRAN

Saman Jodati Gourabi
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, West Azerbaijan, IRAN

Isa Maleki
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, West Azerbaijan, IRAN

ABSTRACT

By the development of the software industry and the advances of the software engineering sciences, the use of Object Oriented Software Engineering (OOSE) has increased in the software complex world. The origin of the OOSE in evaluation and design of the software has expanded much and is now considered as one of the software integration processes. The OOSE is a set of the Object Oriented Analysis (OOA) models, Object Oriented Design (OOD) and the Object Oriented Programming (OOP) which provide a powerful framework for development of the software. The OOSE provides the possibility of OOP on the development and production of the software after the analysis and designing the software. In the paper, we study the OOSE models which is a progressing technology in software industry.

Keywords

Software Development, Object Oriented Software Engineering, Object Oriented Analysis, Object Oriented Design, Object Oriented Programming.

1. INTRODUCTION

Late in three decades the creation and application of the software has progressed a lot. Now, the software industry is so widely extended that the structured methods can no more analyze and design the complex software systems. So, the object oriented models make us defeat the software design and analysis in short time. OOSE is a relatively new method for designing and implementing the software systems. The main goal of the OOSE is the increasing the reusability capacity of the software, reducing the complexity and the software systems maintenance costs [1]. The OOP merges the designing and implementation stages in a high range. So, the workgroups can work on different sections on programming operations. OOSE is based on the classes which identify the behavior of the data present in the software system [2]. OOSE in fact describes a set of analysis, design and implementation processes and this has made the development of the most of the software systems on the bases of object oriented.

Object oriented was first suggested for the development of the software in 1960 [3]. Developers of the software focused on object oriented in 1980s and this method got to be used vastly in the software society [3, 4]. The object technology provides reusability capacity for the software elements and plays important role in acceleration of software development and production of software with higher efficiency [4, 5]. So, using the OOP it is easier to develop and implement the software.

One of the main goals of OOSE is emphasizing the reduction of costs and the time for production of software. OOSE uses the classes and this presents an efficient method for integration in development of software. So, software development using OOSE makes implementation take place once but be used more times. By the appearance of the OOSE the large software systems development has become possible and it has not also reduced the complexity of the software, but also has made the implemented classes be reused in development of other software [6].

The OOA is a powerful model for analyzing the software development. In OOA, each object is considered as a class in implementation procedure and the required and necessary classes are identified to reach the software development [7, 8]. In OOA model, creation of the classes required in system is emphasized. In OOD, the emphasis is on the implementation of the classes, adjectives and specifications which are considered as the core of the class. As a result, the combination of the OOA and OOD models leads to the OOP.

We have organized the general structure of this paper as follow: In section 2, the OOSE is studied; In section 3, the OOSE models is discussed; Section 4, includes the OOSE models are describes and at last the conclusion and the future works are presented in section 5. It must be added that the Unified Modeling Language (UML) software is used for modeling the capabilities of object oriented.

2. OBJECT ORIENTED SOFTWARE ENGINEERING

By the development of the software industry, the OOSE tries to create the models which make the development of the software fast and with low costs. So, the OOSE could be described as a set of the patterns which identify a set of activities and object oriented methods on development of the software. OOSE makes a structure in which the methods, processes and tools are combined for development of software.

The discussable point in OOSE is the design and analysis phases which make the role and the relation dependency for any of the classes more integrated and then the total sketch for software systems is created combining the classes. The object oriented programs generally include objects which use classes to get in relationship. So, the object oriented emphasizes on the completeness of the objects [9].

The object oriented models describe a set of the activities for development of the software and they develop any activity according to a set of the goals. So, OOSE leads to reuse of the software and enters the integration capability into the software. The integration process is a struggle to reach the best specifications of the models of the software process. In OOA, the primary requirements for software are logically and suitably defined. So, OOA process identifies the importance of relation of the customer requirement to the software and adds to the efficiency of the methods in analysis of software [1]. OOP is a set of concepts, bases, methods and tools which are considered in programming the software. The most important goal of the OOP is the reuse of the codes in software development process. The OOP for reuse will lead to reduction of the costs and the reduction of coding of the software. The goal of object oriented is the transforming the customer requirements from a set of defined capabilities to software. To reach this goal, the OOSE must use the OOP and create a total infrastructure for the development of the software. The OOD is focused on completing the classes for description of the objects. All methods and member functions are defined in description of a class. The main goal of the class designing is the mutual effect of the classes on each other [2]. Now, most of the software are object oriented because they make software development easier and fast.

Object Oriented Analysis and Design (OOAD) identifies the customer requirements, the economic and technical facilities and identifies the operation for software, hardware, users and data base which are the key components of software development. The OOAD makes the software meet the requirements of the customers. The goal of OOAD is to reach comprehensive architecture of the data, applicative architecture and infrastructure of the software development. The OOAD makes the software development conform to the requirements and relation activities of modeling [10]. Also, the researchers [4] have studied the (Object Oriented Programming Structure) OOPS capabilities. The goal of OOPS is the modeling based on class and presentation of the relation of the sections of software systems and the analysis of the operative activities. In OOP, the reuse of the program codes is the key factor in success of progressing software. And the goal of OOA and OOD is the improvement of quality and efficiency of the process of software systems development. In OOD, it must be tried to make classes dependent to each other and make them change data with each. So, the classes must be designed in a way that the contents of one class accessible for others and then the test and progress of the software would be easier. The production and development models in each software packet are different in relation to methodology and the selective process model. The software development models are a structured method for attribution of the tasks and the responsibilities in a software development team and the goal is the production of software of high quality, meeting the end users' needs. In OOSE, the requirements and the mutual relationships of the classes in OOA and OOD are identified. Object oriented is very important in development of software, because the object oriented models are very effective in program structure, program sectioning and its complexity. Also, the Component Based Development is more advantageous in increasing the reusability capability, reduction of costs and the time of production and is a very important point in software development [8].

3. THE OBJECT ORIENTED SOFTWARE ENGINEERING MODELS

In OOSE, the software systems develop by a set of models which present complete requirements, designing and OOP [11]. So, OOSE is composed of a set of models which create a framework for software development. The fast changes in software engineering and the changes of technology in this field are the causes which lead to use of OOSE as a framework for integrity and organization of the software elements. Figure (1) shows the dependency of the OOSE for development for software development.

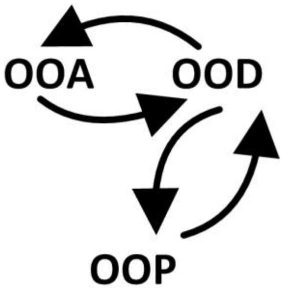

Figure (1): The Relation of the OOSE models

3.1. Object Oriented Analysis

The analysis requirements and providing the comprehensive description of them is the checkpoint in software development. Because software development takes place by the deep look to the findings of the analysis. The OOA conforms all software engineering activities to the software development requirement [12]. So, from the software process view, the OOA identifies the software requirements in software development activities. The OOA fulfills and executes the main requirements of the users, studies the feasibilities, validates the specifications and manages the requirements [13]. The OOA must identify all classes related to the software and the adjectives related to the classes, the relationship between the classes and their behavior [14]. So, the goal of OOA is development of a model which describes software for identifying the requirements of the users. The results of OOA lead to description of the operating specifications of the software, presentation of the relationship of the software with other system elements and identification of the limits which the software faces with [13]. So, OOA in OOSE makes the operating activities, the classes and their relationship, the system behavior and the data in relationship to the classes become clear.

3.2. Object Oriented Design

The main goal of OOSE is the efficiency, reliability, reusability and the capability of sharing the resources in software systems. So, the OOD can support these goals and increase the reusability factor in software development. The OOD is focused on organizing the objects in the classes and all methods and the functions in a class are defined in OOD stage [15]. OOD is a process in which the user requirements are transformed to a design for software creation. In OOD all requirements in an analysis model must be operated and conform to the requirements of the users [16]. OOD transforms the data objects defined in analysis model to the data classes which conform to software development [17]. So, OOD leads to a structure in which the class implementation takes place after requirements analysis. OOD in OOSE includes two main factors [15]. First, software should have no errors violating the operation of it. Second, the relations between the data objects and other information related to each other must be defined.

3.3. Object Oriented Programming

One of the basic techniques of coding is OOP. OOP is a new method in which parts are created and used in different sections of the software [18]. So, when software is implemented by OOP, it is possible to say that software includes a set of commands and in relation classes, and if the classes are organized hierarchically, reusable codes are created which reduce the time and the costs of production. So, the complexity of the operations is reduced and the large software could be manages better. The use of OOP makes the producers hide the complexity of the software systems in classes and use them as an interface for operations [19]. The OOP models the relationship between the objects of the classes and makes the objects get in relationship with each other via sending messages. The OOP packets the data and the functions and provides the reusability, creation and application of the packets to make software development takes place fast [5]. OOP model uses diverse facilities for software development systems. Three main capabilities of OOP are pointed in [2, 6 and 20].

Encapsulation: In OOP classes do encapsulation and the variables, methods and functions are hidden inside them and manage coding in software systems development [2]. So, encapsulation makes any part of coding to be considered as a class and organize the field and the methods better. Also, encapsulation is often called hiding information. Encapsulation creates limits in accessing the internal data of the classes. This limitation makes the specific details of objects' behavior be hidden. So, encapsulation creates a capsule which protects the internal status of the objects from the users. Capsulation is the completeness and integrity and shows the relation of the data and the methods of the classes in integration [6]. Although any of the classes are implemented for a specific responsibility in encapsulation, but all of them are related to each other. Their relationship is mutual and affects each other. Figure (2) shows encapsulation modeling in UML.

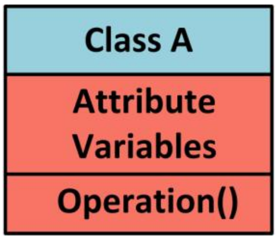

Figure (2): Encapsulation Modeling in UML

As you can see in Figure (2), using the encapsulation, it is possible to packet all variables and the functions in classes. This method makes the coding management in OOP better.

Polymorphism: Using polymorphism in OOP it is possible to define more than one same name functions with different structure in a class [6]. Polymorphism makes a function to be implemented in different ways in classes and sub classes and get different forms. Polymorphism creates a common interface for different implementation of a function for the programmer which operates different for different objects [2]. One of the most powerful facilities of polymorphism is the identification of different dynamic methods in execution time [6]. Totally, polymorphism means "one class, some functions with different structure". This makes the complexity reduce because of the use of one class for identification of some operating functions. Figure (3) shows the polymorphism modeling in UML.

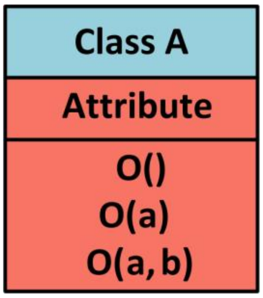

Figure (3): Polymorphism Modeling in UML

As it is seen in Figure (3), using polymorphism capability it is possible to identify some functions with the same name and different structure (number of parameters, type of the parameters).

Inheritance: Inheritance is an operation in which an object can get the specifications of another object. So, using the inheritance it is just needed to identify the specifications which are exclusive inside the classes and inherit the other adjectives from other classes [6]. SO, if changes take place in data and the functions of the inherited classes, they will be operated in other inheriting classes. SO, the hierarchical mechanism for propagating the changes is used in system. Inheritance is used frequently in OOP. Reusability of the codes is one of the main advantages of inheritance [6]. The most important specifications of inheritance are the reusability possibility and harmonization between the classes [20]. Figure (4) shows the inheritance modeling in UML.

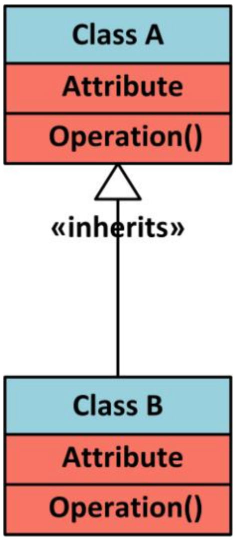

Figure (4): Inheritance Modeling in UML

If the three Encapsulation, Polymorphism and Inheritance basic regulations are applied suitably and if they are well combined, the more reliable software are produced. Using OOP it is possible to combine the different parts of a complex system and produce more efficient software.

4. DISCUSSION

OOSE is no doubt a revolution in Software Engineering lately which could have affected the software development and now we use the vast results of it in Software industry. To design and implement the software systems, OOSE analyzes and creates the objects considered according the software goals. So, no doubt the first step and the most important stage of OOSE, is identification of the objects. In the second stage, the relationship between the objects must be identified. So, to develop the software systems, the two said stages must be identified, and with no awareness there would be challenges in creating the software systems of object oriented procedure. So, it is possible to say that the system implementation process includes two important stages of OOA and OOD. The goal of OOA model is creation of the relationship models which show the software requirements. In OOA modeling takes place according to the objects, their adjectives and their relationships. So, the goal of analysis is providing description of information of software system for designing model. OOD method is in ever changing process toward new methods and better utilization. Utilization means that the software systems must conform to the goals considered for them. Despite this, now the object oriented technology tries to make the software systems flexible. OOD model creates this possibility for software engineering to modeling the software systems under creation. This model could be used for emphasizing the quality and improvement before programming. So, OOD model which includes the presentation of the classes is the primary production of the software development systems.

The OOP combines the encapsulation, polymorphism and inheritance to reduce the complexity of the software and make the programmer code in form of authenticated classes. In OOP, inheritance supports the grouping concept. The grouping technique in inheritance makes each object not define all characteristics of them. So, the objects can inherit the common characteristics of themselves from general classes. One of the key points in OOSE is the reusability concept. Object oriented makes each class to be used more than once for creation of different objects. Using the techniques like inheriting, it is possible to create procedures reusable for software development and just it is enough to call them right and use their speed and ease of use.

5. CONCLUSION AND FUTURE WORKS

Software industry is the key element in completion of systems and the productions based on computer and are one of the most important techniques in the world. So, the goal of OOSE of using object oriented models is the software development to have low complexity. OOSE emphasizes the reuse of the classes and the functions for software development. OOSE is a set of methods, functions and the classes and their goal is the integrity and the combination of data. In OOSE, analysis models, design and programming emphasize on better identification of the requirements, integrity and the reduction of complexity. We with present this paper hope in the future for production of better software in which the design, analysis and programming take place according to OOSE and utilize their capabilities and efficiencies.

6. REFERENCES

[1] N.F.M. Sani, A.M. Zin, S. Idris, "Analysis and Design of Object-Oriented Program Understanding System", International Journal of Computer Science and Network Security, Vol. 9, No. 1, pp. 125-134, January 2009.

[2] K. Gujar, S. Mishra, P. Kawadkar, "Analysis of Function generation on the basis of Object Oriented paradigm", International Journal of Advanced Computer Research, Vol. 1, No. 1, pp. 91-95, September 2011.

[3] A. Dennis, B.H. Wixom, D. Tegarden, "Systems Analysis and Design: An Object-Oriented Approach", John Wiley & sons, New York, 2002.

[4] S. Parthasarathe, N. Anbazhagan, "Analyzing the Software Quality metrics for Objected Oriented Technology", Information Technology Journal, Vol.5, No. 6, pp. 1053-1057, 2006.

[5] S. Georgantaki, S. Retalis, "Using Educational Tools for Teaching Object Oriented Design and Programming", Journal of Information Technology Impact, Vol. 7, No. 2, pp. 111-130, 2007.

[6] J. Din, S. Idris, "Object-Oriented Design Process Model", International Journal of Computer Science and Network Security, VOL.9 No.10, pp. 71-79, October 2009.

[7] D. Budgen, "Software Design", Second Edition, Addison Wesley, New York, 2003.

[8] M. Belal, A. Khedr, A. Gohar, "Frameworks between Components and Objects", Advanced Computing: An International Journal, Vol. 3, No. 5, pp. 9-17, Sep 2012.

[9] F.S. Gharehchopogh, E. Amini, B. Zebardast, "Aspect-Oriented Software Development based Solution for Intervention Concerns Problems: Case Study", International Journal of Computer Applications, Vol. 63, No. 4, pp. 16-25, February 2013.

[10] S.Pasupathy, R.Bhavani, "Measuring the Quality of Software through Analytical Design by OOAD Metrics", International Journal of Computer Applications, Vol. 63, No. 13, pp. 39-44, February 2013.

[11] S.K. Dubey, A. Rana, "A Comprehensive Assessment of Object Oriented Software System using Metrics Approach", International Journal of Computer Science and Engineering (IJCSE), pp. 2726-2730, 2010.

[12] J.W. Satzinger, R.B. Jackson, "Making the Transition from OO Analysis to OO Design with the Unified Process", Communications of the Association for Information Systems, Vol. 12, pp. 659-683, 2003.

[13] S.R. Schach, "Introduction to Object-Oriented Analysis and Design with UML and the Unified Process", Irwin-McGraw Hill, New York, 2004.

[14] C. Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", Prentice Hall, 2005.

[15] A. Dennis, B.H. Wixom, D. Tegarden, "Systems Analysis and Design: An Object-Oriented Approach", John Wiley & sons, New York, 2002.

[16] R. Kumar, D. Gupta, "Object Oriented Design Heuristics", International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 1, pp. 459-463, Jan 2011.

[17] W. Al-Ahmad, "Object-Oriented Design Patterns for Detailed Design", Journal of Object Technology, Vol. 5, No. 2, March-April 2006.

[18] M. Ben-Ari, N. Ragonis, R.B. Levy, "A Vision of Visualization in Teaching Object-Oriented Programming", In Proceeding of 2nd Program Visualization Workshop, HornstrupCentret, Denmark, pp. 83-89, 2002.

[19] C.J. Harrison, O.M. Sallabi, S.E. Eldridge, "An Initial Object-Oriented Programming Language (IOPL) and its Implementation", IEEE, Transactions on Education, Vol. 48, No. 1, pp. 3-10, 2005.

[20] R. Or-Bach, I. Lavy, "Cognitive Activities of Abstraction in Object Orientation: An Empirical Study", ACM SIGCSE Bulletin, Vol. 2, pp. 82-86, 2004.