

# RECENT ADVANCES IN OBJECT ORIENTED TECHNIQUES

Ms.Nutan D Sonwane<sup>1</sup>, Mr.Saurabh Taley<sup>2</sup>

<sup>1</sup> Assistant Professor, Computer Science & Engg, DBACER, Maharashtra, India

<sup>2</sup> Assistant Professor, Computer Technology, SDMP, Maharashtra, India

## ABSTRACT

Software development is a field of engineering that came into existence owing to the various problems that developers of software faced while developing software projects. This paper analyzes some of the most important technological innovations in object oriented software engineering in recent times. The advancements in Object technologies that have been analyzed here include object cloning, introspection and reflection, class co-evolution, global software development contexts, interfaces, namespaces, query-able source codes, meta-model for generating design alternatives, magic methods, design pattern detection, auto-active functional verification. Developments in software technology continue to be dynamic. New tools and techniques are announced in quick succession. This has forced the software engineers and industry to continuously look for new approaches to software design and development, and they are becoming more and more critical in view of the increasing complexity of software systems as well as the highly competitive nature of the industry.

**Keyword :** - Application, Object, Oriented, Paradigm, Software, Development, Concepts

## 1. INTRODUCTION

The major motivating factor in the invention of object-oriented approach is to remove some of the flaws encountered in the procedural approach. OOP treats data as a critical element in the program development and does not allow it to flow freely around the system. OOP and Software development industries are gaining advantages where the two phenomena are emerging and converging to produce efficiency in industries software that helps to solve specific organizational and personalization problems. Take for instance, a lot of sophisticated apps are being produced and release in the market today through the application of OOP. Almost desktop apps are being converted to mobile apps through Java, C++, PHP & MySQL, R, Python etc platform which form the testimony of OOP in the software industries. Software developer has been revolving using procedural language for the past decade before the advent of OOP. The relationships between them is that procedural languages focus on the algorithm but OOP focuses on the object model itself, therefore, making the software programming more visible and increasing the efficiency and usability of the existing object thereby reducing cost.

## 2. Object Oriented Programming

The knowledge of object-oriented programming paradigm in software development is numerous. OOP structuring provides a closer representation of reality than other programming techniques thereby solving complex software systems. A proper analyzed OOP system can implement changes at the class level without affecting the other level in the system development, thereby reducing maintenance cost. This is form a formidable computing power and advantageous of OOP in the industries software development problem solving. The concepts of OOP, the polymorphism and inheritance, allows the reuse individual components, hence, reduces the development time of OO software. OOP also helps in reducing rate of work involved in revising and maintaining the system because during the design state, problems are easily debugged

### 2.1 The Generic of Software development Project

Software Development Project is not an easy process which is made up of many phases. Each phase requires a lot of sketch and report in all the development initiative processes. The expert and the novice do not think alike in software industry with respect to code generation during implementation.

### 2.2 Super Model of Software development

Software development should always follow a particular software development life cycle model like the water fall model and so on in the development of an application from its initial feasibility study down to its deployment and maintenance in the field. There are many other models that could be used to explain various approaches during SDLC process. The SDLC model is widely used during software development because it makes it easier to describe the steps that are followed within the life-cycle framework.

## 3. Features of Object-Oriented Programming

It is necessary to understand some of the concepts used extensively in object-oriented programming. These include:

### 3.1 Data Abstraction and Encapsulation

The wrapping up of data and function into a single unit (called class) is known as *encapsulation*. Data and encapsulation is the most striking feature of a class. The data is not accessible to the outside world, and only those functions which are wrapped in the class can access it. These functions provide the interface between the object's data and the program. This insulation of the data from direct access by the program is called *data hiding or information hiding*. Abstraction refers to the act of representing essential features without including the background details or explanation.

### 3.2 Inheritance

*Inheritance* is the process by which objects of one class acquired the properties of objects of another classes. It supports the concept of *hierarchical classification*. In OOP, the concept of inheritance provides the idea of *reusability*. This means that we can add additional features to an existing class without modifying it. This is possible by deriving a new class from the existing one. The new class will have the combined feature of both the classes.

### 3.3 Polymorphism

*Polymorphism* is another important OOP concept. Polymorphism, a Greek term, means the ability to take more than one form. An operation may exhibit different behavior in different instances. The behavior depends upon the types of data used in the operation. For example, consider the operation of addition. For two numbers, the operation will generate a sum. If the operands are strings, then the operation would produce a third string by concatenation. The process of making an operator to exhibit different behaviors in different instances is known as *operator overloading*.

### 3.4 Message Passing

An object-oriented program consists of a set of objects that communicate with each other. The process of programming in an object-oriented language, involves the following basic steps:

1. Creating classes that define object and their behavior,
2. Creating objects from class definitions, and
3. Establishing communication among objects.

Objects communicate with one another by sending and receiving information much the same way as people pass messages to one another. The concept of message passing makes it easier to talk about building systems that directly model or simulate their real-world counterparts.

A Message for an object is a request for execution of a procedure, and therefore will invoke a function (procedure) in the receiving object that generates the desired results. *Message passing* involves specifying the name of object, the name of the function (message) and the information to be sent.

## 4. CONCLUSIONS

Therefore, the increasing demand of software in the market enterprise today due to numerous problem facing the human being and organization is continuously creating gaps for development of new technologies in OOP. Based on customer and user request, software developers are rapidly finding solution to solving problems. In effort to solve

these regenerating issues, new advanced technologies in the field of OOP are continuously being created and additional technologies are emerging. Software developers are faced with code usability, refractory, maintainability, enhancement of software features, developing from the scratch and many more research are emerging in order to find considerable approach to this clarion call in the software engineering domain.

## 5. REFERENCES

- [1]. Achi Ifeanyi Isaiah, Agwu Chukwuemeka Odi, Alo Uzoma Rit, Anikwe Chioma Verginia, Okemiri Henry Anaya “Technological Advancement in Object Oriented Programming Paradigm for Software Development” International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 8 (2019)
- [2].Saha B. and Muhherjee D., (2017), “Analysis of Applications of Object Orientation to Software Engineering, Data Warehousing and Teaching Methodologies”, International Journal of Computer Sciences and Engineering, Vol. 5, Issue 9, pp 244-248.
- [3] Onu F. U., (2016) “Comparative Study of Structured and OOP Paradigms”, International Journal of Advanced Research in Computer Science and Software Engineering vol. 6, No 8, pp 30-39.
- [4] Kiczales G., John L., Anurag M., Maeda C., Cristina V., Jean-Marc L., and Irwin J., (1997), Aspect Oriented Programming, proceedings of the European Conference on Object-Oriented Programming (ECOOP),Finland. Springer-Verlag LNCS 1241.
- [5] Skelly J. F., (1997), Application Of Object Oriented Programming Techniques In Front End Computers, Paper submitted to: QCALEPCS ' 97 Beijing , China, pp 3-5
- [6] Schuster F., Tendyck T., Liebchen C., and Davi L., (2015), Counterfeit Object-oriented Programming: On the Difficulty of Preventing Code Reuse Attacks in C++ Applications, 2015 IEEE Symposium on Security and Privacy, pp 745-762
- [7] Corral R., Morgado-Estevéz A., Jimenez G., and Civit A., (2014), A game-based approach to the teaching of Object-oriented programming languages, Computer and Education, Vol 73., pp