IS IT TRUE TO SAY? AOP VS OOP

Anurag Sharma*

Rishika Kalsi*

Abstract

There is a different programming paradigm in use by software programmers to make their software more reliable and to have great maintainability factor. In Software development OOPs (Object Oriented Programming) supported languages are generally used in industry, but now AOP is introduced in software industry which actually expands the functionality of OOPs with extra features of AOPs (Aspect Oriented Programming). In this paper, we focus on what is oops and if object oriented model is already used by software developers then what is the need of Aspect oriented Programming. AOP is aspect oriented programming, which actually extends the boundary of object oriented programming.

Keywords – ASPECT, ASPECT ORIENTED PROGRAMMING,CODE SCATTERING,CODE TANGLING, OBJECT ORIENTED PROGRAMMING.

* Department of IT, Jagan Institute of Management Studies, Delhi,India

Introduction

Main focus of this paper is to understand the OOPs and if OOPs is already there in market then why AOP was introduced to software development industry. Object oriented programming(OOPs) was developed in late 1960-70. First language which supported OOPs is SIMULA. Paper is divided into 5 sections. In first section we discuss the previous work on OOPs and AOP. The 2nd and 3rd sections are used to elaborate the concept of object oriented programming and Aspect oriented programming, we also discuss, why AOP came in limelight and what were the drawbacks in OOPs and were not solved by it. Basically, object oriented programming refers to object as its primary thing to interact between different classes and modules, but when multiple classes or modules interact it created a hustle in the program which ultimately led to code tangling and code scattering, which is further described in Aspect oriented programming section. These two problems which were un-solvable by Object-oriented programming, is now solved using Aspect oriented programming. All the terms which are used in aspect oriented programming are given a brief description.

Literature Reviewed

Previous research shows that how object oriented paradigm is useful for solving real life problems. [1] Rakesh Kumar Mishra, Bharat Lohani in his article of "*An object oriented Software development approach to design simulator for airborne altimetric Lidar*". In this paper, author uses an Object Oriented software design strategy where system designer thinks in terms of 'thing' instead of operations on function. Author uses an object oriented methodology so it can be extended further very easily. Java programing language is used to develop the software which makes this software robust and platform independent.

[2] Andrea Rinaldi in his article "*Transforming a company with OOP*". In this paper, author describes how changing necessity of company forces to adapt object oriented paradigm and how OOP resolves company problems. This paper presents an account of how object-oriented software development was introduced in his company, the foundation of our understanding object orientation, and the hurdles overcame by author's company in the process of building a reusable application framework. They developed a fully object oriented architecture that allow them to decouple their software development teams from the competitive peer pressure and from sudden changes in fiscal regulations. This approach helps user to make it possible to avoid the

lack of direction in the component ware approach, giving them a unity of purpose in the software development.

These two cases are the where object oriented concept is adapted by developer to resolve their problem and make a system very reliable and understandability, but there is also some case which leads to how OOP more complex and unstructured if we measure on large scale of productions.

[3] Mr. Kailash Patidar, Prof.RavindraKumar Gupta, Prof.Gajendra Singh Chandel article "*Coupling and Cohesion Measures in Object Oriented Programming"*. In this article, authors discuss the complexity in object oriented programming. In any software, its complexity affects most of the quality attributes of software quality. For finding the software quality, a large no of metrics has been built and proposed for finds or measures the properties of object-oriented software such as size, inheritance, cohesion and coupling. The coupling is an important aspect in the evaluation of reusability and maintainability of components or service. In this paper, author presents an idea on how to reduce coupling in object oriented programing. It is helpful for developers to check which concept is best between inheritance and interface.

[4] Lech Madeyski, Lukasz Sza in article of "*Impact of aspect-oriented programing on software development efficiency design quality: an empirical study"*. Aspect-oriented programming approach is supposed to enhance system's features, such as its modularity, readability and simplicity. As we discussed in [3] authors find the complexity in OOPs, we can say that complexity and understandability is prime factor for aspect oriented comes in limelight. AOP provides better modularization of crosscutting concerns, due to this developed system implementation would be less complex, and more readable.

[5] Heba A. Kurdi in "*Review on Aspect Oriented Programming"*. Author describes Aspect-oriented programming has been introduced as a potential programming approach for the specification of nonfunctional component properties, such as fault- tolerance, logging and exception handling. These properties are referred to as crosscutting concerns and represent critical issues that conventional programming approaches could not modularize effectively leading to complex code. AOP concept uses to better code reusability, understandability and maintainability of code. Author concluded that AOP is a programming approach to solve crosscutting concerns by providing better modularization of the code. In this paper author also

shows demand of software industry for better and improved software quality, and how AOP uses to overcome this problem.

Object Oriented Programming

In this section, we understand what is object-oriented programming and how it is adapted by most of the software developers. Object oriented mainly focuses on data rather than function, so programmers are now thinking more about the data than the business logic. In object oriented, "object" refers to as the data structure which allows to create a new data structure in programing by using the concept of classes and fields and object can be a real world entity such as pen, chair, table, etc. It helps to maintain your code in a structural format and this structured code helps software developers to maintain the understandability of code.

It simplifies the software development and it is quite useful at the time of maintainability of code, reusability, efficiency, portability. These all are the most important attributes in software quality and oops helps us to provide a software which follows these software quality attributes in a better way. OOPs achieve this quality by using Object-Oriented programing features:

1. Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

Object: It is a real world entity or basis of object oriented programing. It can be physical and logical. In context of programming, it is instance of a class.

Class: It is group of similar type of objects. In programing it a user defined data structure which contains methods and function.

Inheritance: It is most important feature of object oriented programming. This features enables reusability of code. Program once and use multiple times in your code.

Polymorphism: *poly:* many and *morphism:* many, means same name and different functionality, which leads to less code redundancy and, more structured way of source code.

Abstraction: In abstraction, we hide all irrelevant detail and shows only relevant information to user.

Encapsulation: Binding code and data together into a single unit.

These are the important aspects of object oriented programming, due to these features object oriented programming is widely accepted as a main software developing programming paradigm. But for making a good software, software quality is more important aspect for any software vendor and buyer. Now every buyer wants that their software must be follows all standard quality attributes and if software will follow all quality attributes.

There are some Quality attributes which is defined under ISO 9126 model:

1. Functionality
2. Reliability
3. Usability
4. Efficiency
5. Maintainability
6. Portability

Now the question is if already object oriented programming is existing and it also follows software quality attributes than why AOP came in action. First understand what is AOP (Aspect Oriented Programming) then we will check how it extends the scope of object oriented programming.

Aspect Oriented Programming

Aspect oriented programming was introduced by Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingitier and John Irwin at Xerox Palo Alto Research Centre. According to this team, [6] Object oriented programming techniques are sufficient to clearly capture some of important design decisions the program must implement. This forces the implementation of those design decisions the program must implement. They find tangled code which is excessively difficult to develop and maintain. The basis of this new programming technique is called aspect-oriented programming, that makes it possible to clearly express programs involving such aspects, including appropriate isolation, composition and reuse of the aspect code.

[6] Software design processes and programming languages exist in a mutually supporting relationship. Design processes break a system down into smaller and smaller units. Programming languages provide mechanisms that allow the programmer to define abstractions of system sub-units, and then compose those abstractions in different ways to produce the overall system. A

design process and a programming language work well together when the programming language provides abstraction and composition mechanisms that cleanly support the kinds of units the design process breaks the system into.

Two major problem solved by Aspect Oriented Programming is

1. **Code Scattering:** It is arising when the functionality is scattered because it is existing in different modules.

   There are two type of code scattering:

   - Blocks of duplicated code which leads to redundant code in project.
   - Line of execution is distributed in different modules which create hustle in code (like authorization in different module and authentication is in different module.)

2. **Code Tangling:**It occurs when a module has to manage several concerns at the same time such as Logging, Exception Handling, Security and Caching and many more, defined by spring 2.5 AOP programming.

AOP Solution

As we have seen earlier, code tangling and code scattering are the major issues generated by object-oriented programming, there also is code duplicity divides the code into different modules, due to which further implementations creates a problem. This is resolved by AOP

1. By creating cross-cutting concerns.
2. Uncoupling of the modules
3. Using aspects, which helps to removing the dependency between modules.

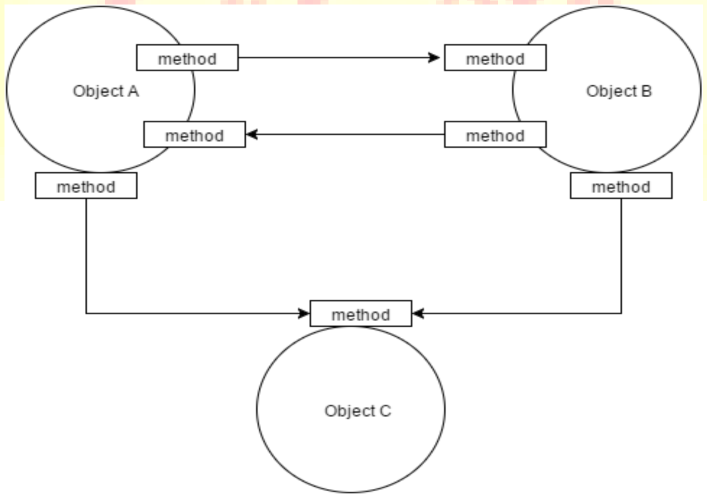

Figure 1: Shows normal interactions between objects

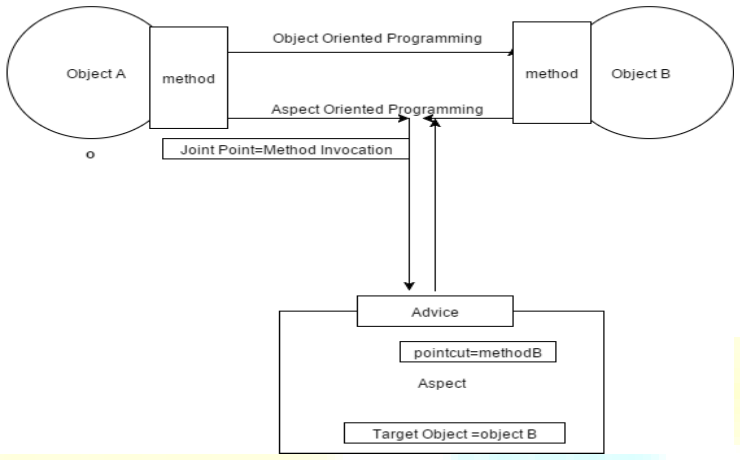
Object A
method
Object Oriented Programming
Aspect Oriented Programming
method
Object B
Joint Point=Method Invocation
o
Advice
pointcut=methodB
Aspect
Target Object =object B

Figure 2: Execution flow of the invocation of a method in case of AOP and OOP.

In figure 2, crosscutting concerns are placed in the implementation of A, B and C and helps to understand the problem of code tanaling and code scattering.

What is crosscutting concerns?

In simple words, we can say that when several classes or modules affected by a concern, that concern can be defined as a crosscutting concerns. As a solution, AOP overcomes this problem by introducing a new unit of modularity –an aspect. Aspect is used to avoid code tangling and code scattering problem.

Some AOP terms which are rapidly used in AOP are:

- **Aspect:** Important part in AOP, this is the point where we implements AOP to increase the modualrity of code.
- **JoinPoint:**This is a point where you call method or raises any kind of exception.
- **Advice:** When any method is called at particular joint point, a behaviour is trigger, that is advice.
    - These Advice are futher divided in spring 2.5 for better understanding:
        1. Before Advice
        2. After Returning Advice
        3. Throws Advice
        4. After(finally) advice
        5. Around advice
- **PointCut:** This is the expression for jointpoint's selection.

- **Introduction:** it can be used to forces existing class to implement an interface or to add as annotation to anything when aspect is attached to particular class.
- **Code Weaving:** it is used at that time when you create linking between objects and aspect.

Now the most important thing is that, aspect oriented programming is not a replacement of object oriented prgramming but it rather compliments it. We can say that, the problem which is not solved by OOP is solved by AOP, or in a statstical way 90% of problem is solved by OOPs but rest 10% of it is handled by AOP. In some contexts AOP is a good solution provider than OOP.

In this section, we state all those problems which led to the birth of a new programming paradigm that is aspect oriented programming. Object oriented programming is strictly for those languages which follow OOPs concept but AOP is different as it supports most of the lanuages either OOPs or any other pogramming paradigm. This is the most crucial point of using Aspect-Oriented Programming.

Conclusion

In this paper, we are trying to understand why AOP was introduced to software development and how it is a powerful tool,if you have a great knowledge about this programming paradigm. Object oriented programing will alwaysbe the first preference of every programmer. But there are some constraints which were not resolved by OOPs, here AOP comes in power.

As we discussed OOP solves 90% of the problem but rest 10% is beautifully handled by aspect oriented programing.There is no comparison between OOPs and AOP as both are different in their respect. AOP requires lot of research for better use in software developing. If it is used properly in software development, its quite surprising that how AOP will help to attain all important quality attributes with ease.

To conclude, there is no comparison between these two programming paradigms, invidually both are best and we can also say that aspect oriented programming is a compliment to object oriented pogramming.

Referrences

[1] Rakesh Kumar Mishra, Bharat Lohani, "*An object oriented Software development approach to design simulator for airborne altimetric Lidar"*.

[2] Andrea Rinaldi,"*Transforming a company with OOP"*.

[3] Mr. Kailash Patidar, Prof.RavindraKumar Gupta, Prof.Gajendra Singh Chandel article "*Coupling and Cohesion Measures in Object Oriented Programming"*.

[4] Lech Madeyski, Lukasz Sza, "*Impact of aspect-oriented programing on software development efficiency design quality: an empirical study"*.

[5] Heba A. Kurdi, "*Review on Aspect Oriented Programming"*.

[6] Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingitier and John Irwin, "Aspect Oriented Programming".

[7] J.H. Hayes and L Zhao, "Maintainability Prediction: a Regression Analysis of Measures of Evolving Systems," Proc. 21st IEEE International Conference on Software Maintenance, 26 - 29 Sept. 2005, pp. 601 -604, 2005.

[8] Avadhesh Kumar, Rajesh Kumar, PS Grover,"An Evaluation of Maintainability of Aspect-Oriented Systems: a Practical Approach"

[9] Daesung Park, Sungwon Kang, Jihyun Lee," Design Phase Analysis of Software Qualities Using Aspect-Oriented Programming"

[10] Avadhesh Kumar, Rajesh Kumar, PS Grover "A Fuzzy Logic Approach to Measure Complexity of Generic Aspect-Oriented Systems

[11] IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990

[12] R. Laddad, "Aspect Oriented Programming will improve Quality", 2003,published by IEEE Computer Society 0740-745.

[13] Pradeep Kumar Singh, Om Prakash Sangwan, Amar Pal Singh, Amrendra Pratap,"A Framework for Assessing the Software Reusability using Fuzzy Logic Approach for Aspect Oriented Software", IJITCS, vol.7, no.2, pp.12-20, 2015. DOI: 10.5815/ijitcs.2015.02.02

[14] Alessio Ishizaka and Ashraf Labib, "Review of the main developments in the Analytic Hierarchy Process".

[15] Mehmet Yüksel, "Evaluating the Effectiveness of the Chemistry Education by Using the Analytic Hierarchy Process"