BEST: International Journal of Humanities, Arts, Medicine and Sciences (BEST: IJHAMS)
ISSN 2348-0521
Vol. 2, Issue 10, Oct 2014, 43-50
© BEST Journals

LIMITATIONS OF ASPECT ORIENTED PROGRAMMING IN INDUSTRY

SWATI PARASHAR

Galgotias University, Greater Noida, Uttar Pradesh, India

ABSTRACT

Aspect Oriented Programming has become centre of attraction since its inception in 1996 at Xerox Palo Alto Research Centre (PARC).It supports the Object Oriented Programming paradigm by separating the cross cutting concerns from business logic[1].

In 2001 Aspect Oriented Programming was identified as one of the most influential technologies in coming days. However even after a decade of its birth it is still not being implemented as it was supposed to be. This paper discusses the limitations of Aspect Oriented Programming in IT Sector.

KEYWORDS: Adoption, Aspect Oriented Programming (AOP), Object Oriented Programming (OOP)

INTRODUCTION

Business has become very complex due to strategies in Information Technology Projects. IT managers have to face the risk arising out of vendors whether they are single or preferred and high cost due to use of proprietary standards and tools. Object Oriented Programming (OOP) features software development with the facility of modeling real life objects in programming languages thereby reducing complexity and increasing efficiency. Separation between application programmers and system programmers become more and more different. These concerns need to be more and more robust which can't be implemented within limited timelines of the project.

IT Organizations have tough time in managing all the diverse teams and ensuring project completion within budget and time. In collaborative software development many outsourcing partners get involved based on their domain competency, team size, skill set and other business requirements.

Aspect Oriented Programming helps in referring cross cutting concern which in turn is supposed to lessen the cost and complexity. The objective of this research is to identify pain areas and limitations of Aspect Oriented Programming on a big scale software development using enterprise wide computing. As the popularity of computer technology application is increasing the size of software is becoming bigger Software developers develop and renovate the programming technology by achieving more perfect software to improve the efficiency of developing and maintaining the software.

OOP technology is widely used to develop software at present. An OOP program is created with a series of object which are basic module of the program.

The communication happens by sending message within object. The special features of OOP are inheritance polymorphism and encapsulation. Now a day's some special features are more granular module, more excellent reusability and extensibility and increase the efficiency of programming.

Although, OOP has excellent features, still it has limitations. AOP offsets the limitations of OOP. The technology of AOP is more ideal and modularized.

SURVEY

The Basic Concept of AOP

Aspect Oriented Programming cannot be considered as a substitute of Object Oriented Programming, in contrast, it is the advance and perfect Object Oriented Programming.

AOP is not a substitute of OOP, It is the perfect version of OOP. OOP describes the vertical relationship between the core logic object in the system using inheritance. AOP is the horizontal action which is encapsulated in different object.

Concepts of Aspect Oriented Programming are

Crosscutting Concerns

It is the purpose of the program. If the code which has to perform a concern is distributed in many classes or methods, it is known as Crosscutting Concern.

Aspect

An aspect in modularized implementation for a crosscutting concern. It integrates the distributed code which is to be used for crosscutting concern and for horizontal representation of relation between the objects.

Join Point

A join is used for invoking a method or throwing or throwing an exception.

Advice

An advice is a programming code used to cross objects in a specific join point.

Pointcut

It is basically used for appointing the place where the advice will be applied.

The Implementing Strategy of AOP in Java Platform

Dynamic Proxies

The technique has been imported from Java 1.3 and is used as AOP solution. It has capacity of creating proxy object automatically for some interfaces without having knowledge of object.

Aspect J

Aspect J defines additional keyword for java. It gives perfect support to AOP. Aspect J needs a compiler to generate to java core to develop AOP program with aspect J software development kit (SDK) is needed to be installed.

Generation of Dynamic Byte Code

AOP has limitations of implementing dynamic proxies it requires that object which is provided proxy must implement one or more interfaces. Dynamic bite code generation software is CGLIB. This feature has been widely used by AOP.

There are two basic platforms which support Aspect Oriented Programming. One of them is Java and the second one, less known is .NET.

Industry Adoption of Aspect Oriented Programming

Aspect Oriented programming has history of successful implementation in many enterprises. Some of the big giants in industry which have adopted it are Motorola, Seimens, HP, SAP etc.

Motorola developed WEAVER for telecom needs, HP adopted AOP in C++ for developing VLSI CAD applications. Seimens used AOP by using java for better architectural benefits.SAP has analyzed Enhancement framework developed by using ABAP to access the issues in AOP. They have found the basic factors which promote AOP in industry

History and Evolution of Aspect Oriented Programming Research [6]

Gregor Kiczales and his friends at Xerox Palo Alto Research Center (PARC) developed the concept of Aspect Oriented Programming between November 1995 and May 1996.It was based on a strong foundation of prior work, but at that time there was not existing terminology to describe what was being done and very soon it spread in the whole world.

The RG Case Study

Earliest projects at PARC which contributed to the development of Aspect Oriented Programming as a paradigm was Reverse Graphics.

The paper on RG case study [3] discusses how Object Oriented Programming’s shortcomings would be overcome using Aspect Oriented Programming techniques.

Annotated Matlab (Aml)

The second project that contributed in the development of Aspect Oriented Programming paradigm was Annotated Matlab (AML) [5]. It discussed optimization of certain Matlab Programs focusing on memory usage and operation fusion. Annotated Matlab (AML) was not given much importance because it did not match with other Aspect Oriented Programming systems at PARC.

Evaluated Time Control Meta Language

This work was done by John Lamping who gave the concept of a small system called ETCML for providing a set of directions that programmers could use in order to instruct the language processor as to when evaluate the given parts of code. This contributed in providing another dimension of separation of concerns of enriching Aspect Oriented Programming.

Related Work

Aspect Oriented Programming researchers have been curious about the benefits of AOP and its quality parameters. Still there are many challenges that are to be looked after in managing enterprise wide computing and enterprise architecture by using AOP in place of OOP.

Kickzales et al [1] identified the problems with OOP in 1997 and gave the concept of AOP by doing image processing in RG. The result was that AOP was much better than OOP in meeting the result. But still problems like implementation cost and its complexity were not in view. There was no way to reduce the complexity of code size.

In second example, AOP was implemented on a distributed digital library which had huge store of documents in many forms. The result was almost appropriate. Still question of maintenance, domain, application were not answered.

Roger T Alexander [8] and James M Bieman discussed certain aspects like understandability fault tolerance, cognitive burden in practical scenario. The complexity coming out of weaving process is still not measurable with accuracy.

Avadhesh Kumar, Rajesh Kumar and P S Grover [10] gave ideas on maintainability of Aspect Oriented Systems and have found that average impact of change in AO systems is less than of OO systems which indicates better maintenance in AO systems but still according to them cross cutting concern was very important to maintain this aspect.

Zhao [9]also did some work but that was not applied as it was less practical.

Joon Sang Leau and Doo Hwan Baeb [11]in 2002 proposed an Aspect Oriented Development Framework (AODF)in which functional behaviours are encapsulated within the components. AODF was supposed to enable intra component behavior. Open Java has been used providing reflective support during compile time. This provided light on feature component writing standard.

Roberta Coellio et al[12] in 2010 discussed about the drawbacks of Aspect Oriented Programs realted to exception handling. A verification tool called SAFE based on static analysis was presented to check the reliability of exception handling code in Aspect J Programs.

Jianjun Zhao [8] put light on difficulties in AOP software maintenance because less attention has been paid to it till now.

Gail C Murphy et al found that while evaluating a software engineering methodology three factors are kept in mind which are – validity, realism and cost.

OBSERVATIONS

Jorg Kiezle et al pointed out two important facts in his paper on concurrency and failure of Aspect Oriented Programming.

Aspect Oriented Language like any other macro language can be useful for coding.

Concurrency and failures are some concerns which are difficult to use unlike other concerns like logging and exception handling.

Some possible reasons have been listed out to the research problem:

Developers lack of detailed knowledge of AOP and modularized concept.

Developers fail to understand units which are similar but cross cut other modules.

The capabilities of Aspect J and its side effects.

Use of Aspect Oriented Programming in Enterprise

As per review in January /February 2001 by MIT, ten emerging areas of technology will have profound impact on economy and our lifestyle. One of the said emerging technologies was Aspect Oriented Programming. However, it did not gain expected popularity.

What is an Enterprise?

Basically enterprise is a collection of organizations having a common set of goals. It may be any government agency, a corporation or anything else. In those places Aspect Oriented Programming may be required to record failed financial transactions across all divisions which requires architecture of enterprise to be flexible to handle these concerns in a unified, seamless way independent of other issues like functional requirements.

Use of AOP in Architecture of Enterprise

Aspect oriented Programming provides a way to address orthogonal concerns in a unified approach of an enterprise architecture.

AOP reduces code tangling, coupling and helping in architecture enforcement.

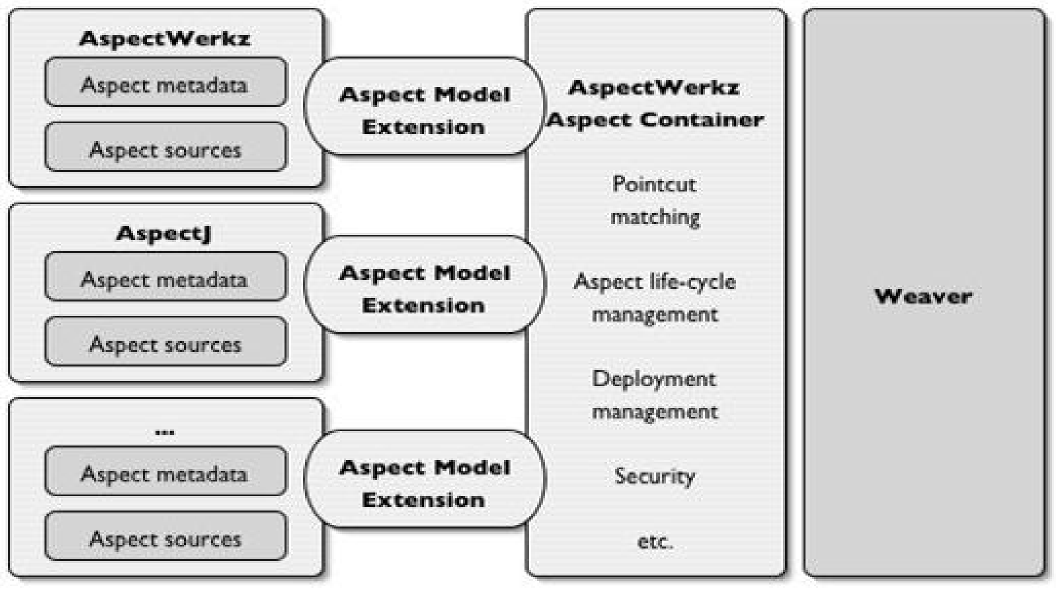

Figure 1

Problem Definition and Objectives [7]

Social and Psychological Factors

An internal survey of SAP shows that there is lack of experts in Aspect Oriented Programming. Developers feel less confidence in implementing it and are afraid of effects on their base code.

Technical Factors

For AOP strong support tools for tracing, verifying, debugging is needed. Risks rising out of Aspect are involved.

Economic Factors

Its very difficult to predict the overall cost for its implementation as it is still less used in industries.

RESULTS ANALYSIS

Reasons which have hindered adoption of AOP:

AWARENESS

Less market penetration has lead to very specific usage. It’s still less user friendly.

Lack of Supporting Framework

There is still lack of universal platform for AOP framework.

Lack of Technical Experts

It has been still less heard of so technical experts are very few in number.

Legal Issues

It still does not have clear laws though it deals with sensitive issues such as payroll, financial transactions etc.

Multi Vendor Integration

Multi vendor integration leads to confusion because one aspect code cutting through other vendor's module is very unclear.

CONCLUSIONS

Aspect Oriented Programming has been in existence for more than a decade but still it has been as popular as Object Oriented Programming though it's much more efficient. Its penetration into IT sector is still scarce because of less awareness.

There are still only a few successful implementation known so far. Much study is still needed so that Aspect Oriented adoption in industry increases to its optimum.

REFERENCES

1. Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier, John Irwin, European Conference on Object-Oriented Programming (ECOOP), Finland. Springer-Verlag LNCS 1241, June 1997, 2-3
2. Cristina Videira Lopes, Aspect Oriented Programming: An Historical Perspective - Institute of Software Research, University of California, Irvine, December 2002, 9-10
3. Gregor Kiczales, John Lamping, Anurag Mendhekar RG: A case study for Aspect-Oriented Programming, February 1997,12-13
4. Karl Lieberherr and david H. Lorenz - Coupling Aspect-Oriented and Adaptive Programming April, 2003, 1-2
5. John Erwin, Jean-Marc Loingtier, John R. Gilbert, Gregor Kiczales, John lamping, Anurag Mendhekar, Tatiana Shpeisman - Aspect-Oriented Programming of Sparse Matrix Code –- December, 1997, 2-7
6. Gregor Kiczales, Eric Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm and William G. Griswold - An Overview of Aspect J — June, 2001,1-9
7. Xin Ma, Lian-he Yang -AOP Research based on Enterprise Application – World Academy of Science and Technology, 2008
8. Roger T. Alexander, James M. Bieman - Challenges of Aspect Oriented Technology – Workshop on Software Quality, Florida, 2002, 1-3
9. Jianjun Zhao -Change Impact Analysis for Aspect Oriented Software Evolution – Department of Computer Science and Technology, Fukuoka Institute of Technology – 2002, 3-5
10. Avadhesh Kumar, Rajesh Kumar, P. S. Grover - Maintainability of Aspect Oriented Software Systems – 2006, 6-7

11. Joon-Sang Leea, Doo-Hwan Baeb - An aspect-oriented framework for developing component-based software with the collaboration-based architectural style, August 2002, 7-9
12. Roberta Coelho, Arndt von Staa, Uirá Kulesza, Awais Rashid, Carlos Lucena Unveiling and taming liabilities of aspects in the presence of exceptions: A static analysis based approach – June 2010, 1-4
13. Jianjun Zhao - Maintenance Support for Aspect-Oriented Programs: Opportunities and Challenges – 2008, 2, 6
14. Gail C. Murphy, Member, IEEE Computer Society, Robert J. Walker, Student Member, IEEE, and Elisa L. A. Baniassad - Evaluating Emerging Software Development Technologies: Lessons Learned from Assessing Aspect-Oriented Programming, August 1999, 1-4
15. Cristina Videira Lopes, Martin Lippert - A Study on Exception Detection and Handling Using Aspect-Oriented Programming – 2000, 10-11
16. Freddy Munoz, Benoit Baudry, Romain Delamare, Yves Le Traon - Inquiring the Usage of Aspect-Oriented Programming: An Empirical Study – 2009, 2-4
17. http://www.ccs.neu.edu/research/demeter/aop/publicity/mit-tech-review.html - visited on 10-Dec-2012
18. Uirá Kulesza Cláudio Sant‟Anna, Alessandro Garcia, Roberta Coelho, Arndt von Staa, Carlos Lucena - Quantifying the Effects of Aspect-Oriented Programming: A Maintenance Study – 2008
19. J. Kienzle, R. Guerraoui - AOP: Does it Make Sense? The Case of Concurrency and Failures.- Proc. ECOOP'02 – 2002, 2-5
20. he Open Group Architecture Framework (TOGAF) – Version – 9.0
21. Paulo Merson - Using Aspect-Oriented Programming to Enforce Architecture – Software Engineering Institute – 2007
22. http://www.ibm.com/developerworks/java/library/j-aopwork15 -Ramnivas Laddad, Feb 2006 - Visited on 01-Jan-2013
23. Kim Mens and Tom Tourwe - Evolution Issues in Aspect-Oriented Programming –
24. Christoph Pohl, Anis Charfi, Wasif Gilani, Steffen Göbel, Birgit Grammel, Henrik Lochmann, Andreas Rummler, Axel Spriestersbach - Adopting Aspect-Oriented Software Development in Business Application Engineering - AOSD Industry Track 2008 Brussels, Belgium
25. T. Cottenier, A. van den Berg, and T. Elrad. The Motorola WEAVR: Model Weaving in a Large Industrial Context. - AOSD, 2007, 9-10
26. D. Wiese, R. Meunier, and U. Hohenstein. How to Convince Industry of AOP- AOSD, 2007, 7-8
27. M. Mortensen and S. Ghosh. Using Aspects with Object-Oriented Frameworks - AOSD, 2006, 1-4
28. HE Qinghe, AOP Programming Thinking Study, Software Guide, 2010 (3)
29. ZHANG Haifeng, KONG Leilei YUAN Tao, Overview of AOP and design patterns, Heilongjiang Science and Technology Information,2009(35).
30. WEI Zhenyuan, Research and Discussion on AOP Technology 基, Silicon Valley, 2008 (15).

31. HU Bing, CHENG Jiaxing GUO Jianye, An Approach to Implement AOP Framework Under .NET Platform, Computer and Modernization, 2009 (11).

32. TANG Zukai, PENG Zhiyong, Survey of Aspect-Oriented Programming Language, Journal of Frontiers of Computer Science and Technology, 2010 (4).

33. MENG Jie, Mastering Spring—Java lightweight Framework Developing Practice, BeiJing: Posts&Telecom Press, 2006(10).

34. Tom Archer, Andrew Whitechapel, Inside C#, Second Edition, BeiJing: China Machine Press, 2003 (7).

35. *Reference:* http://www.theserverside.com/news/1365141/AspectWerkz-20-An-Extensible-Aspect-Container