

A Review on Adoption of AOSD for Design and Development of Large Software Systems

Alok Singh

*Assistant Professor -IT
Chandrabhan Sharma College of Arts, Science &Commerce
Powai ,ViharMumbai-400076
Maharashtra,India*

Lovely LakhmaniBalani

*Assistant Professor –IT
Smt. P.N. Doshi College
Ghatkopar(W), Mumbai
Maharashtra ,India*

Abstract - AOSD is an emerging software development Paradigm which is becoming increasingly popular in Software Development community. AOSDis currently being used in the design, development and maintenance of large software systems. In this paper we will discuss preliminary experience reports from early adopters. AOSD has matured greatly and is being adopted in software industry for large scale software system development. This paper examines the critical requirements and opportunities for the development of successful software systems based on theAOSD technology. It identifies different application areas; where AOSD can be applied effectively and efficiently. In this paper we will present some of the key successes and pitfalls in the application of AOSD. In this paper we will try to identify key factors which are critical for the successful adoption of AOSD in design and development and maintenance of large software systems.

Keyword: AOSD,AOM,AOP, aspect,AspectJ , Spring Framework, JBoss, etc.

I. INTRODUCTION

AOSD is an emerging software development Paradigm which is becoming increasingly popular in Software Development community. AOSD paradigm consist of AORE (Aspect Oriented Requirement Engineering), AOM (Aspect Oriented Modeling) and AOP.

Aspect Oriented Programming (AOP) helps in addressing cross cutting concerns which in turn is supposed to reduce cost and complexity. The objective of this research is to identify opportunities and challenges in adoption of AOSD for large scale software development. Our aim is to identify key application scenarios where AOP can be effectively used to add value through cost reduction, overcoming integration challenges and controlling maintenance overheads [7].

AOSD is currently being used in the design, development and maintenance of large software systems. In this paper we will discuss preliminary experience reports from early adopters. AOSD has matured greatly and is being adopted in software industry for large scale software system development.

This paper examines the opportunities and challenges in developing reliable software systems based on the AOSD technology. It identifies different application areas; where AOSD can be applied effectively and efficiently. Some of the early adopters of AOSD are IBM, BEA, JBOSS, and spring. In this paper we will present some of the key successes and pitfalls in the application of AOSD. In this paper we will try to identify key factors which are critical for the successful adoption of AOSD in design and development and maintenance of large software systems. AOSD provides ability to deal with the complexities associated with transactions, persistence, security, logging, tracing and exception handling[1].

In order to measure potential of adoption of AOSD for large software development we need to conduct a rigorous study of strength, weakness, opportunities and challenges of AOSD technology.

II. STRENGTH OF AOSD

- Aspect-oriented software development focuses on the identification, specification and representation of cross-cutting concerns[9].
- AOSD provides better modularization into separate functional units as well as their automated composition into a working system[2][9].
- AOSD provides better understanding of problem domain.
- AOSD provides better understanding of effects of cross cutting concerns through early aspects i.e. Aspect Oriented Requirement Engineering.
- AOSD allows expressing multiple concerns separately and weaving it into a single working system.
- AOP handles dynamic crosscutting in a more effective and efficient way.
- Availability of Aspect Oriented Programming Languages and Frameworks: aspectJ and AJDT(Eclipse)-IBM[8], Aspectwerkz and Weblogic-BEA[8],JBoss AOP- JBossgroup[8], Spring AOP- spring project, AspectS-DOCOMO Lab, CaesarJ-Technical university of DarmStad ,AspectC- University of British Columbia and Aspect C++ - University of Erlangen-Nuremberg[8].

These frameworks facilitate AOSD development tools that can be utilized to design and development of large software.

III. LIMITATIONS OF AOSD

The AOSD paradigm also has limitations which can affect its smooth adoption by software development community and Industry these minor glitches such as conflict between aspects, base language and aspects& complexity in Aspect-Oriented Analysis [5].

IV. OPPORTUNITIES OF AOSD

Most of the current application focus for AOSD is Requirement Engineering, Modeling, Architecture and design, development Testing and maintenance of large software systems. Following are some of the key areas where AOSD can be applied effectively and efficiently:

- To work on development of Aspect Oriented models above the level of the relational database implementation.
- To work on development of debugging and maintenance toolset for AOSD.
- To provide much simpler programming model for EJB developers.
- The most exciting opportunity is in the field of Aspect Refactoring. Aspect refactoring will be a key technique for Agile/XP developers. It will allow them to systematically reduce the complexity and rigidity of large legacy applications[10].
- The New Aspects approach is to focus oncrosscutting application security enforcement for privacymanagement[11].
- There are many promising application development areas where AOSD technology can be applied effectively; like Web services, augmenting or replacing application servers (e.g., EJBalternatives or customized J2EE policies)[12].
- Other infrastructure domains (e.g., security, management,persistence, application frameworks, domain frameworks)[13].
- Mobile computing[13].
- Embedded and real-time computing
- Tools support for existing development (tracing,refactoring, “fluid AOP”to provide views on existingsystems, etc.), which appeals especially to agilepractitioners[14].

- Developer testing (e.g., improving mock objects, instrumentation, code in-variants)[15].

V. INDUSTRY ADOPTION

Aspect Oriented Software Development (AOSD) technology is used to develop large scale Software by prominent software development organizations. Few notable applications of AOSD in industrial software development are:

Sr. No.	Software Product	Company	AOSD Feature used
1	IBM WebSphere Application Server(WAS)	IBM	AspectJ is used to isolates features of different editions that supports different features
2	JBOSS Application Server(JBOSS AS)	JBOSS	JBOSS AOP is used to deploy security and transaction management
3	Oracle TopLink	Oracle	Spring AOP is used to achieve persistence transparency
4	Java ME platform	Sun Microsystems	Aspects are used to simplify the development of mobile applications.
5	Sorian- Health information Management System	Siemens	Used aspectJ to integrating crosscutting features such as tracing, auditing and performance monitoring.
6	Wi 4 control software	Motorola	Weaver is used for debugging and testing purposes.
7	Glassbox	Glassbox	The glassbox inspector monitors the activity of JVM using AspectJ
8	.NET 3.5	Microsoft	Supports Aspect Oriented concepts through Unity container.
9	MySQL	MySQL	Logging feature in MySQL is implemented using aspect.

VI. CONCLUSION

Aspect Oriented Software Development (AOSD) technology is used to develop large scale Software by prominent software development organizations. Few notable applications of AOSD in industrial software development are IBM websphere, JBoss, Apache Tomcat, Oracle TopLink etc.

REFERENCES

- [1] Dave Thomas: Reflective Software Engineering from MOPS to AOSD, in Journal of Object Technology, vol. 1, no. 4, September-October 2002, pages 17-26. http://www.jot.fm/issues/issue_2002_09/column1 Ref.
- [2] https://en.wikipedia.org/wiki/Aspect-oriented_software_development

- [3] Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J., Irwin, J. *Aspect Oriented Programming*. In Proc. of ECOOP '97, LNCS 1241, pp. 220-243, Springer-Verlag, 1997
- [4] Elrad, T., et al., *Discussing Aspects of AOP*, Communications of the ACM, 2001, 44(10), p. 33-38
- [5] Xavier Medianero, Sérgio Crespo C.S. Pinto and Clifton Clunie, *Aspect Oriented Programming Methodology to Support the Design of Specific Domain Framework Design of Specific Domain Framework*, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012 ISSN (Online): 1694-0814 www.IJCSI.org
- [6] Kiczales, G., *Aspect-Oriented Programming – the Fun Has Just Begun*, Vanderbilt Workshop, New Visions for Software Design & Productivity: Research & Applications, 2001
- [7] Dr. Debashis Jena, Dr. S.C. Das, Ambika Prasad Das, *Adopting Aspect Oriented Programming in Enterprise Wide Computing*, International Journal of Engineering Science Invention ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726 www.ijesi.org Volume 2 Issue 7 July. 2013 PP.29-35.
- [8] Dr. Awais Rashid, *Computing Department Lancaster University, Aspect oriented Software Development*.
- [9] https://en.wikipedia.org/wiki/Aspect-oriented_software_development.
- [10] **Dave Thomas**, Bedarra Corporation, Carleton University and University of Queensland, *Reflective Software Engineering - From MOPS to AOSD*
- [11] Ravi Uyyala, Kundan Kumar Mishra, *Aspect Oriented Program Design in Distributed Application*, International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 3, March 2012) .
- [12] www.ibm.com/support/.../SSEQTP_7.0.../cejb_bindingsejbf.html
- [13] https://en.wikipedia.org/wiki/Software_framework
- [14] Kiczales, G., *Aspect-Oriented Programming – the Fun Has Just Begun*, Vanderbilt Workshop, New Visions for Software Design & Productivity: Research & Applications, 2001.
- [15] <http://guide.agilealliance.org/guide/tdd.html>