IJCTA, 9(27), 2016, pp. 93-97
ISSN: 0974-5572

A Review on Coupling Metrics in Aspect Oriented System

E. Kirubakaran* and K.R. Martin**

ABSTRACT

The aspect-oriented programming (AOP) is a new paradigm for improving the system's features such as modularity, readability and maintainability. Aspect-oriented software development (AOSD) is a new technique to support separation of concerns in software development. AspectJ is an implementation of aspect-oriented programming for Java. It adds to Java only a few new constructs: pointcuts, advice, inter-type declarations and aspects.Coupling is an internal software attribute that can be used to indicate the degree of interdependence among the components of a software system. Coupling is thought to be a desirable goal in software construction, leading to better values for external attributes such as maintainability, reusability, and reliability. Thus, in AO systems, the coupling is mainly about the degree of interdependence among aspects and/or classes. This paper addresses the development and implementation of various coupling metrics for AOP design paradigm and outlines the future directions.

Keywords: Aspect Oriented Programming (AOP), Aspect Oriented Software Development (AOSD), Cognitive, Coupling Metrics, Aspect J.

1. INTRODUCTION

Aspect-Oriented Software Development(AOSD) is a programming paradigm that overcomes the limitations of Object- Orientation (Programming) providing more suitable abstractions for modularizing crosscutting concerns that cannot be decomposed from the rest of the software artifacts. Software metric is a measure of some property of a piece of software or its specifications[2]. Coupling is one of the software metric for Aspect Oriented Software. Coupling measures have important applications in software development and maintenance. They are used to help developers, testers and maintainer's because regarding software complexity and software quality attributes. Coupling or dependency is the degree to which each program module relies on each one of the other modules[4].

Coupling is an internal trait of the software which can be applied to be a sign of the degree of system interdependence among the components of software. Coupling is considered to be one of the necessary goals in software construction, which will eventually lead to better maintainable, reusable and reliable software products. In aspect oriented software various coupling measures have been proposed[7].

In this article an attempt is made to highlight the studies on aspect oriented coupling metrics. Aspect Oriented software development is not a replacement for object oriented paradigm, but aspect orientation complements object orientation. So many of the metrics used in the aspect oriented systems may be extended from object oriented software.

The rest of the paper is organized as follows. The coupling metrics related works are discussed in Section 2. Section 3 presents the works done in coupling metrics related to aspect oriented scenario. Future Directions are discussed in section 4. The concluding remarks are given in Section 5.

* AGM (SSTPS), BHEL, Tiruchirapalli, TamilNadu, India, *Email: ekiru@bheltry.co.in*

** Department of Computer Science St.Joseph's College (Autonomous) Tiruchirappalli, India, *Email: krmartincs@gmail.com*

2. LITERATURE REVIEW

Ceccato and Tonella[7] introduced many aspect oriented metrics which included aspect oriented coupling metrics as well. The metrics that the study used was the extension of the metrics suite from objects oriented metrics. The work also collected the value for the metrics from software using the developed tool.

Zhao[10] proposed a coupling measure suite that implements the coupling framework for AspectJ like programming languages. The study also discussed the mathematical properties of these metrics, in which he showed that these measures satisfy properties that a good coupling measure should have. One drawback with the study was that it did not consider dependencies between aspects or between classes.

Bartch and Harrison[6] suggested a coupling framework for AspectJ, which is an extension of the coupling framework for Object Oriented systems[17]. This extended framework contains a specific definition of the different coupling mechanisms found in AspectJ. The authors validated existing coupling metrics theoretically for AOP language using these framework criteria. The work concluded that the definition of coupling metrics is difficult tasks that need a great deal of rigor if the measures are to be theoretically valid and well defined.

Kumar et al[4] extended their framework for coupling measures from [16][17][18] frameworks and proposed new coupling metrics for generic AO systems. They identified the connections that cause coupling in aspect oriented system and defined six coupling measures namely coupling on attribute type(CoAT), coupling on parameter type(CoPT), coupling on attribute reference(CoAR), coupling on operation invocation(CoOI), coupling on inheritance(CoI) and coupling on high level association(CoHA).

Kulesza et al[5] presents a quantitative study that assess the positive and negative effects of AOP on maintenance activities of a web information system. The study also considered the positive and negative influences of AOP on coupling measures when compared to the object oriented version of the same system.

AnanthiSheshasaayee and Roby Jose[2] studied about Aspect Oriented Coupling and Cohesion Measures for aspect oriented systems. This study is planned to frame an idea about the coupling, cohesion measures and framework all along with tool support for the coupling measures.

Mandeep Kaur and Rupinder Kaur [1]analysed Improving the Design of Cohesion and Coupling Metrics for Aspect Oriented Software Development. This study focuses on developing metrics for better calculation of coupling and cohesion values. OOPs also have some limitations like in the system decomposition there are some functionalities cannot be assigned to single module. There is one paradigm that enhances software design and promotes reusability called Aspect Oriented Paradigm (AOP).So they focused on several object oriented metrics and find how Aspect affect these metrics.

3. COUPLING METRICS FOR ASPECT ORIENTED SYSTEM

Number of AOP metrics has been developed in the past and still new metrics are proposed. Following are several metrics which have been developed for AOP.

4. FUTURE DIRECTIONS

AOSD is increasingly adopted technique in software development, but these are very few metrics for AOP to prove its low complexity.Coupling is the degree to which components depend on one another. There are two types of coupling, “tight” and “loose”. Loose coupling is desirable for good software engineering but tight coupling may be necessary for maximum performance. It has been recognized that good software design should obey the principle of low coupling. Another important issue of measuring the metrics is cognitive aspect of programming. None of the researchers were concentrate Cognitive complexity of AOP. Existing metrics are not appropriate for determining aspect complexity. The following coupling metrics are to be developed for Aspect oriented software in future is tabulated in Table 2.

Table 1
Coupling Metrics in AOP

Metrics	*Description*	*Proposed by*
Coupling on Intercepted Modules(CIM)	Number ofmodules or interfaces explicitly named in the point cuts belonging to a given aspect.	
Coupling on Method Call (CMC)	Number of modulesor interfaces declaring methods that are possiblycalled by a given module.	
Coupling on Field Access(CFA)	Number of modules orinterfaces declaring fields that are accessed by a givenmodule.	
Coupling on Advice Execution(CAE)	Number of aspectscontaining advices possibly triggered by the executionof operations in a given module.	
Crosscutting Degree of an Aspect (CDA)	Number ofmodules affected by the pointcuts and by the introductionsin a given aspect.	Cecceto et al.[7]
Number of Children (NOC)	Number of immediate subclassesor sub-aspects of a given module.	
Depth of Inheritance Tree(DIT)	Length of the longestpath from a given module to the class/aspect hierarchyroot.	
Response For a Module(RFM)	Methods and advicespotentially executed in response to a message receivedby a given module.	
Coupling Between Modules (CBM)	CBM measures the number of modules or interfaces declaring methods or fields that are possibly called or accessed by a given module.	
Crosscutting Degree of an Aspect caused by Pointcuts(CDP)	CDPmeasures the number of modules affected by the pointcuts in a given aspect.	
Crosscutting Degree of an Aspect caused by Intertype-declarations(CDI)	CDI measures the number of modules affected by the intertype-declarationsin a given aspect.	
Coupling on Advice Execution caused by Methods(CAM)	CAM measures the number of aspects containing advices possibly triggered by theexecution of methods in a given module.	
Coupling on Advice Execution caused by Advices(CAA)	CAA measuresthe number of aspects containing advices possibly triggered by the execution of advices in a given module.	
Coupling on Advice Execution caused by Intertype-declarations(CAI)	CAI measures the number of aspects containing advices possibly triggered by the execution of intertype-declarations in a given module.	J Zhao and H Shen [10]
Response For a Module caused by Methods(RFM)	RFM measures the number of methods potentially executed in response to a message received by a given module.	
Response For a Module caused by Pointcuts(RFP)	RFP measures the number of advices potentially executed in response to a message received by a given module.	
Coupling on Attribute Type(CoAT)	Specifies how each attribute in a module whether local or global, is coupled with other modules and vice versa.	
Coupling on Parameter Type(CoPT)	Specifies how each function is coupled in other programs and vice versa.	
Coupling on Attribute Reference(CoAR)	Specifies the extent to which a method is dependent on others.	
Coupling on Operation Invocation(CoOl)	Total no of operations invoked in a program that is coupled with other program invoking the same method.	Kumar et al[4]
Coupling on Inheritance(Col)	As the name states it's an inheritance based coupling. Methods are coupled with their parent via inheritance.	

Table 2
Case study of various Coupling Metrics

Metrics Component	CAE	CFA	CIM	CDA	CMC	DIT	NOC	CBM	RFA	CDP	CDI	CAM	CAA	CAI	RFM	RFP	CoAT	CoPT	CoAR	CoOI	COI
Coupling Metrics in AOP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Cognitive Complexity	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

- To measure the overall complexity of Aspect.
- To identify Cognitive complexity for Aspect Pointcut, Advice and Inheritance.
- To measure the complexity for coupling metrics

5. CONCLUSIONS

AOSD is motivated by rigorous quality factors, such as maintainability, reusability, cohesion, coupling etc. This survey presents various coupling metrics for Aspect Oriented Software Development. Emphasis has been given on coupling metrics because the principle for good quality software is "Low Coupling and High Cohesion". Even though the Aspect oriented software development is increasingly being adopted for software development, measuring the cognitive complexity for the metric is still a difficult task. Can be formed out thus there is a need of complexity metric that can measure the time with all the aspects of the software.

REFERENCES

[1] Mandeep Kaur, "Improving the Design of Cohesion and Coupling Metrics for Aspect Oriented Software Development", *International Journal of Computer Science and Mobile Computing*, **4(5)**, 99-106,May- 2015.

[2] AnanthiSheshasaayee and Roby Jose,"A Descriptive Study about Aspect Oriented Coupling and Cohesion Measures",*International Journal of Innovative Science, Engineering & Technology*, **2(8)**, August 2015.

[3] Ritika Chaudhary and Ram Chatterjee," Reusability in AOSD - The Aptness, Assessment and Analysis", *International Conference on Reliability, Optimization and Information Technology -ICROIT*, 2014.

[4] Kumar, Avadhesh, Rajesh, and P. S. Grover. "Generalized coupling measure for aspect-oriented systems." *ACM SIGSOFT Software Engineering Notes*, 1-6, 2009.

[5] Kulesza, and Uira "Quantifying the effects of aspect-oriented programming: A maintenance study",Software Maintenance, *22nd IEEE International Conference onIEEE,* 2006.

[6] Bartsch, Marc, and Rachel Harrison. "Towards an empirical validation of aspect-oriented coupling measures",*Proc. Assessment of Aspect Techniques*, 2007.

[7] Ceccato, Mariano, and Paolo Tonella. "Measuring the effects of software aspectization." *1st Workshop on Aspect Reverse Engineering*, **12**, 2004.

[8] Zhao, Jianjun, and Baowen Xu. "Measuring aspect cohesion",*Fundamental Approaches to Software Engineering. Springer Berlin Heidelberg*, 54-68, 2007.

[9] H. Shen, S. Zhang, and J. Zhao, "An empirical study of maintainability in aspect-oriented system evolution using coupling metrics",*in Proceedings of the 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering* , 233–236, IEEE Computer Society, June 2008.

[10] Haihao Shen and Jianjun Zhao," An Evaluation of Coupling Metrics for Aspect-Oriented Software", 2007.

[11] Ms. Puneet Jai Kaur and Ms. SakshiKaushal, Cohesion and coupling measures for aspect oriented systems, *Science of Computer Programming*, 2011.

[12] S.R.Chidamber and C.F. Kemerer. A metrics suite for object oriented design. *IEEE transactions on Software Engineering*,**20(6)**:476-493,June 1994.

[13] Burrows, R., Ferrari, F. C., Garcia, A., and Taïani, F., An empirical evaluation of coupling metrics on aspect-oriented programs. *In Proceedings of the 2010 ICSE Workshop on Emerging Trends in Software Metrics*, 53-58, ACM 2010.

[14] Bartsch, M., and Harrison, R., "An evaluation of coupling measures for AspectJ. *In LATE Workshop AOSD*", 2010.

[15] Zhao, "Measuring coupling in aspect-oriented systems", *IEEE International Soft. Metrics Symposium*, Chicago, 2004.

[16] Bartolomei, Thiago T., "Towards a unified coupling framework for measuring aspect-oriented programs." *Proceedings of the 3rd international workshop on Software quality assurance. ACM*, 2006.

[17] Briand, Lionel C., John W. Daly, and Jürgen K. Wüst, "A unified framework for coupling measurement in object-oriented systems",*Software Engineering, IEEE Transactions*, 91-121, 2001.

[18] Arisholm, Erik, Lionel C. Briand, and AudunFoyen. "Dynamic coupling measurement for object-oriented software." *Software Engineering, IEEE Transactions*, 491-506, 2001.

[19] The AspectJ Team, "The AspectJ Programming Guide",2001.

[20] Joseph D.Gradecki, Nicholas Lesiecki, "Mastering AspectJ – Aspect-Oriented Programming in Java", 2003.

[21] O'Reilly, "AspectJ Cookbook", 2005.