A Descriptive Study about Aspect Oriented Coupling and Cohesion Measures

Dr. Ananthi Sheshasaayee[1], Roby Jose[2]

[1]Research Supervisor, PG and Research Department of Computer Science & Application,
Quaid-E- Millath Government College for Women (Autonomous), Chennai 600 002, ananthi.research@gmail.com

[2]Research Scholar, PG and Research Department of Computer Science & Application,
Quaid-E-Millath Government College for Women (Autonomous), Chennai 600 002, roby.research@gmail.com

***Abstract* – Coupling and cohesion 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, cohesion measures and frameworks have been proposed. Some of these coupling and cohesion metrics are lately introduced to meet the architectural nuances introduced by aspect orientation while some are extended from traditional object oriented metrics. This study is planned to frame an idea about the coupling, cohesion measures and framework all along with tool support for the coupling measures.**

Keywords* - *Aspect Oriented Metrics, Cohesion metrics, Cohesion Framework, Coupling Metrics, Coupling Frameworks, Separation of Concerns, Tool Support

1. INTRODUCTION

Aspect Oriented Programming [17] is the new programming paradigm that overcomes the limitation of Object Oriented Programming and achieves modularity. AOP achieves modularity and eliminates code tangling in OOP by mining out concerns and condensing them into separate programming module which is otherwise scattered throughout the entire system.

The aspect oriented systems can be evolved from object oriented systems by altering the static OOP model to meet the new code that should be included to accomplish the further requirements. Thus it could be correctly stated that Aspect Oriented Programming complements Object Oriented Programming but not replaces. AOP makes the crosscutting concerns separate, which augment the complexity of the software. Coupling is one approach of calculating the complexity of the software. An indicator to the complexity of the new approach of software development can be attained by using coupling measures.

In this article an attempt is made to highlight the studies on aspect oriented coupling and cohesion metrics. In [11] Zhao et al pioneered the coupling metrics based framework for AspectJ [14] like programming languages. Further research into coupling frameworks was instigated by [7] which also performed empirical validation of coupling measures. From the literature it is understood that cohesion also is very important while evaluating the quality of the software.

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 work [1] identified some metrics for the maintainability studies of aspect oriented software. The present paper is an attempt to figure out the coupling and cohesion metrics from [1] based works in the area of aspect oriented software development.

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

II. ASPECT ORIENTED COUPLING METRICS

Burrows [3] et al in the study for maintainability metrics used Ceccato and Tonella[12] metrics for assessing maintainability. Apart from the above metrics, the work introduced a new metric BAC(Base Aspect Coupling) which measures the coupling between base class and aspect. The dependent variable used in the study was fault proneness and the study showed that BAC is highly correlated with presence of faults which lead to the high maintenenance.

Shen[5] et al used changeability as the dependent variable to estimate maintenance effort. For this purpose the sole metrics that was considered in the study was coupling metrics introduced by Ceccato and Tonella. The study initiated that these coupling metrics are highly correlated with maintenance tasks.

Przybyek[2] used modularity as the dependent variable for assessing the maintainability of the aspect oriented systems. This study also made use of coupling metrics to measure the modularity of aspect oriented system. The study initiated that while measuring modularity aggregate coupling should not be considered as coupling should be measured independent of the number of modules in the system.

Ceccato and Tonella[12] introduced many aspect oriented metrics which included aspect oriented coupling metrics as well. The coupling metrics introduced by the work was later adopted by many researchers. To assess the proposed metrics suite, we developed an AOP metrics tool that computes all the proposed measures for code written in the AspectJ [14] language. The metrics that the study used was the extension of the metrics suite from objects oriented metrics.

Zhao[11] proposed a coupling measure suite that implements the coupling framework for AspectJ like programming languages. The coupling measures are defined on the number of dependencies between aspects and some classes. 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[7] suggested a coupling framework for AspectJ, which is an extension of the coupling framework for Object Oriented systems[16]. 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.

Bartolomei et al [8] proposed a unified coupling framework for AOP. In order to consistently and unambiguously define coupling measurements the study defined new terminologies for representing numbers of Aspect Oriented systems. This framework considered two languages AspectJ and CaeserJ. The framework being extended from Briands framework , the work tried to maximize the use of terminology and coupling criteria in the original framework. This study failed to propose any new coupling measures.

Kumar et al[4] extended their framework for coupling measures from [8][16][13] 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[9] 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.

III. ASPECT ORIENTED COHESION METRICS

Sant'Anna[15] proposed two new metrics Lack of cohesion in operation(LCOO) and Lack of concern based cohesion(LCC). LCOO measures the amount of method/advice pair that do not access to the same instance variable.It is an extension of the Chidamber Kemerer metric[18] Lack of cohesion in methods (LCOM).According to the study a high LCOO value indicates the disparateness in the functionality provided by the aspect. LCC counts the number of concerns addressed by the assessed component. The drawback was that it did not provide quantification criteria for aspect in empirical evaluation.The definition of the metric was based on AspectJ programming language.

Zhao and Xu[11] defined cohesion measures for aspect oriented systems, which are based on a dependency model for aspect oriented software that is a group of dependency graphs. The work defined cohesion as degree of relatedness between attributes and modules. The cohesion was measured for inter-modules and module attribute dependencies. This study suggested a complex way of measuring cohesion which may be very difficult in real development context.

Gelinas[10] proposed the metric ACoH to measure aspect cohesion. This cohesion measurement is based on dependency analysis. The aspect cohesion was based on two connection criteria viz Modules-Data connection criteria and Modules-Modules connection criteria. The mathematical evaluation for the new ACoH metric was also proposed by the authors.

Kumar et al[6] proposed a cohesion metric for aspect oriented systems. This work identified six connections that cause cohesion. Using these connections and framework criteria they defined a metric called as Unified Aspect Cohesion($UACoH_i$). The same work also proved that the cohesion metric cannot be used as an indicator for changeability assessment of aspect oriented systems.

IV. CONCLUSION

Aspect Oriented Software Development (AOSD) aims at improving the modularity of software by providing structures which permit to modularize crosscutting concern whose representation in code cannot be isolated using traditional object oriented approach. Using object oriented approach for this purpose may lead to code cloning, which may give difficulty (again) to developers

ISSN 2348 – 7968

to identify all modules using the concern which may make maintenance difficult. In this scenario aspect orientation is a rewarding change. But the paradigm is at its burgeoning stage. Measurement of the object oriented paradigm was given a solid philosophical basis through Chidamber Kemerer and Li Henry. Aspect Orientation also needs to be matured on the lines of object oriented with respect to measurement and prediction features of metrics.

REFERENCES

[1] Sheshasaayee, Ananthi, and Roby Jose. "International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS) www. iasir. net."

[2] A. Przybyek, “Where the truth lies: AOP and its impact on software modularity,” in *Proceedings of the 14th International Conference on FundamentalApproaches to Software Engineering*,Springer, Saarbrcken, Germany, 2011.

[3] R. Burrows, F. C. Ferrari, A. Garcia, and F. Ta¨ıani, “An empirical evaluation of coupling metrics on aspect-oriented programs,”in *Proceedings of the ICSE Workshop on Emerging Trends in SoftwareMetrics (ICSE ’10), pp. 53–58,ACM,CapeTown, South Africa, May 2010.*

[4] Kumar, Avadhesh, Rajesh Kumar, and P. S. Grover. "Generalized coupling measure for aspect-oriented systems." *ACM SIGSOFT Software Engineering Notes* 34.3 (2009): 1-6.

[5] 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 (TASE’08)*, pp. 233–236, IEEE Computer Society, June 2008.

[6] Grover, P. S., Rajesh Kumar, and Avadhesh Kumar. "Measuring changeability for generic aspect-oriented systems." *ACM SIGSOFT Software Engineering Notes* 33.6 (2008): 1-5.

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

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

[9] Kulesza, Uira, et al. "Quantifying the effects of aspect-oriented programming: A maintenance study." *Software Maintenance, 2006. ICSM'06. 22nd IEEE International Conference on*. IEEE, 2006.

[10] Gélinas, Jean-François, Mourad Badri, and Linda Badri. "A Cohesion Measure for Aspects." *Journal of object technology* 5.7 (2006): 75-95.

[11] Zhao, Jianjun, and Baowen Xu. "Measuring aspect cohesion." *Fundamental Approaches to Software Engineering*. Springer Berlin Heidelberg, 2004. 54-68.

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

[13] Arisholm, Erik, Lionel C. Briand, and Audun Foyen. "Dynamic coupling measurement for object-oriented software." *Software Engineering, IEEE Transactions on* 30.8 (2004): 491-506.

[14] Kiselev, Ivan. Aspect-oriented programming with AspectJ. Sams, 2003.

[15] Sant’Anna, Cláudio, et al. "On the reuse and maintenance of aspect-oriented software: An assessment framework." *Proceedings of Brazilian Symposium on Software Engineering*. 2003.

[16] Briand, Lionel C., John W. Daly, and Jürgen K. Wüst. "A unified framework for coupling measurement in object-oriented systems." *Software Engineering, IEEE Transactions on* 25.1 (1999): 91-121.

[17] Kiczales, Gregor, et al. Aspect-oriented programming. *11th European Conference on Object Oriented Programming in:LNCS,vol.1241,Springer Verlag*,1997,pp 220-242 1997.

[18] 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