Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 13 Issue 7 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Improve Relevancy of Object Oriented Class Cohesion Metrics with Inheritance

By Jaspreet Kaur & Rupinder Kaur
Lovely Professional University, India

Abstract - Cohesion is a very important quality attribute in software. As we know that there are number of cohesion metrics are proposed in the literature to measure the cohesion of software systems. These metrics gives undefined values for a large number of classes which comes under special cases. Because of this reason, these metrics became non-applicable for these classes as they are unable to give cohesion values for these classes. In this paper, a value assignment criterion would be used to make cohesion metrics applicable and the concept of inheritance would be included for these special cases. Study the effect of including or excluding the inherited elements i.e., methods and attributes.

Keywords : *object-oriented software quality, class cohesion, cohesion metric, HLD, LLD.*

GJCST-C Classification *: D.2.3*

Strictly as per the compliance and regulations of:

Improve Relevancy of Object Oriented Class Cohesion Metrics with Inheritance

Jaspreet Kaur [α] & Rupinder Kaur [σ]

Abstract - Cohesion is a very important quality attribute in software. As we know that there are number of cohesion metrics are proposed in the literature to measure the cohesion of software systems. These metrics gives undefined values for a large number of classes which comes under special cases. Because of this reason, these metrics became non-applicable for these classes as they are unable to give cohesion values for these classes. In this paper, a value assignment criterion would be used to make cohesion metrics applicable and the concept of inheritance would be included for these special cases. Study the effect of including or excluding the inherited elements i.e., methods and attributes.

Keywords : object-oriented software quality, class cohesion, cohesion metric, HLD, LLD.

I. Introduction

Software Engineering is the branch of computer science which is mainly concerned with developing large applications. There are number of quality attributes which are available to measure the quality of softwares i.e., maintainability, reusability, availability, reliability, cohesion, coupling, security, scalability, testability, usability etc.

a) Class Cohesion

Cohesion can be defined as the relatedness of elements in a module. There can be two types of modules in software system:

- Highly Cohesive Module
- Low Cohesive Module

Highly Cohesive Module is defined as those modules whose elements have a tight relationship among themselves. A **Low Cohesive Module** is defined as the module that has some elements that have little or no cohesion relation to others.

Class consists of following two members:

- Attributes
- Methods

Attributes can be defined as the things where the objects stores the data i.e., variables. On the other hand, **Methods** can be defined as Functions and Procedures that are attached to an Object and allowing the object to perform the different actions.

Author α : Student, CSE Department Lovely Professional University, Phagwara, 144411, India. E-mail : er.jaspreetkaur8789@gmail.com
Author σ : Assistant Professor, CSE Department Lovely Professional University Phagwara, 14441, India. E-mail : rupinder.16835@lpu.co.in

Class Cohesion Metrics are applicable on the following two phases:

- High Level Design (HLD Phase)
- Low Level Design (LLD Phase)

Several HLD and LLD metrics have been proposed in the literature to measure cohesion. HLD metrics require the information that is available during the HLD phase, such as types of attributes and method parameters. LLD metrics requires the information that is available during LLD phase, such as attributes referenced by the methods.

As we know that, there have been proposed a number of cohesion metrics in the literature in order to measure the cohesion of various software systems. But most Class Cohesion Metrics shows the undefined values for some special cases given as follows:

- Classes consisting of fewer than two methods.
- Classes that do not contain any attributes.
- Classes in which none of the methods has parameters.

In 1998,2004,2011,2012, authors of [1], [2], [6], [7], [8], [9], [10], [11], [12], [13] and [14] excluded all classes that come under special cases mentioned above for which 12 object oriented class cohesion metrics like LCOM1, LCOM2, LCOM5, TCC, LCC, NHD, CC, SCOM, COH, DCD, DCI and CAMC gives undefined cohesion values and cohesion value can never be infinity.

II. Related Work

This guideline is used for all journals. These are the manuscript preparation guidelines used as a standard template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This guideline is used for all journals. This guideline is used for all journals. These are the manuscript preparation guidelines used as a standard template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This guideline is used for all journals. This guideline is used for all journals. These are the manuscript preparation guidelines used as a standard template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This guideline is used for all journals. This guideline is used for all journals. These are the manuscript preparation guidelines used as a standard

template for all journal submissions. Author must follow these instructions while preparing/modifying these guidelines. This guideline is used for all journals.

a) Review Stage

In 2011, [3] introduces criteria for assigning cohesion values to classes of special cases, such as classes having lesser than two methods, classes containing no attributes, classes having methods with no parameters. They used a value-assignment criteria to assign values to special cases. But they have not included two factors including inheritance and the accessibility levels i.e., public, protected and private). Through this value-assignment criteria, the applicability of considered metrics in paper increases to 100%.

This paper worked on same criteria as [3] mentioned i.e criteria for assigning cohesion values for special cases, such as classes having lesser than two methods, classes containing no attributes, classes having methods with no parameters and additionally, it will include a factor of inheritance i.e., the inherited attributes and methods would be included.

This paper considered four scenarios:

- both inherited methods and attributes are excluded,
- only inherited attributes are excluded,
- only inherited methods are excluded,

both inherited methods and attributes are included.

Following table shows assigned values are used for metrics to calculate cohesion for special case:

Metric	m=0 and a>0	m=1 and a=0	m=1 and a>0	m>1 and a=0
NHD	0	0	0	1
Coh	1	1	Assigned Before	0
TCC, LCC, CC, SCOM, CAMC	1	1	1	0
DCD, DCI	1	1	1	Assigned Before
LCOM1, LCOM2	0	0	0	Assigned Before
LCOM5	0	0	2*(1-α/a), where α is the summation of the number of distinct attributes accessed by each method in a class.	m/(m-1)

Table 2.1 : The assigned values for the metrics under consideration [3]

In the table 2.1 **m** refers to method, **a** refers to attribute and **assigned before** means the metric does not show undefined value for the case shown in respective column

b) Final Stage

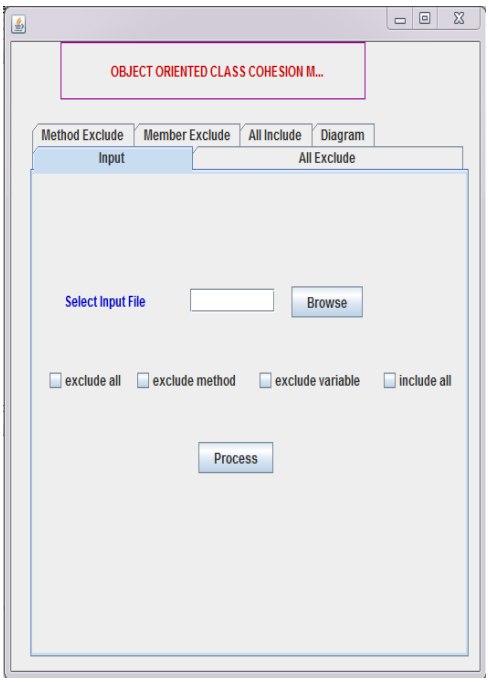
This paper proposed a tool named SoftMetric Tool to show the comparisons and results of four scenarios using Netbeans IDE 7.3. Java is a high-level programming language originally developed by Sun Microsystems and released in 1995 [4]. **Netbeans** is a Java IDE that is open source and free. Most developers recognize the NetBeans IDE as the original free Java IDE [5].

i. *SoftMetric*

SoftMetric Tool is implemented using Netbeans. This tool will calculate the cohesion values for the following 12 metrics:

- TCC (Tight Class Cohesion)
- LCC (Loose Class Cohesion)
- NHD (Normalized Hamming Distance)
- CC (Class Cohesion)
- SCOM (Class Cohesion Metric)
- LCOM1 (Lack of Cohesion 1)
- LCOM2 (Lack of Cohesion 2)
- LCOM5 (Lack of Cohesion 5)
- COH
- DCD (Degree of Cohesion - Direct)
- DCI (Degree of Cohesion - Indirect)
- CAMC (Cohesion Among Methods in a Class)

Following is shown the snapshot of the tool:



Following we take an example of class named test1.class to show the calculation of cohesion using this tool:

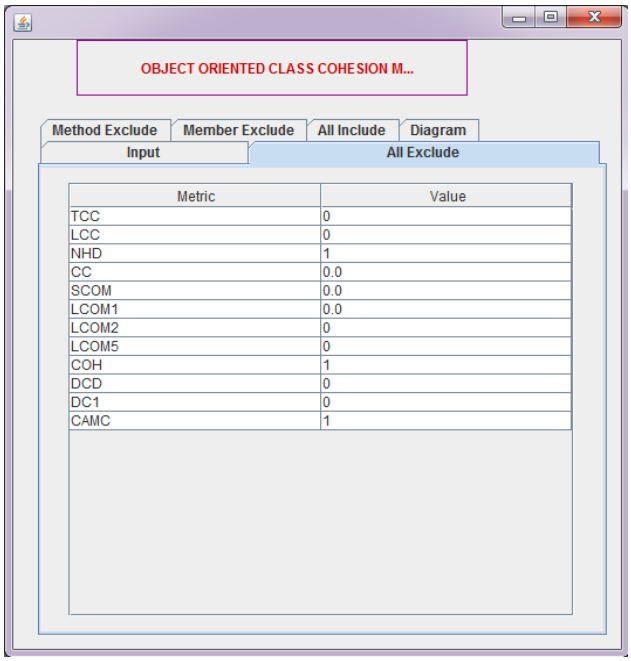
1. This following snapshot shows "**Select Input File**", this will browse and select .class file. After selecting a class file, we will select one option from four options:

- exclude all,
- exclude method,
- exclude variable
- include all.

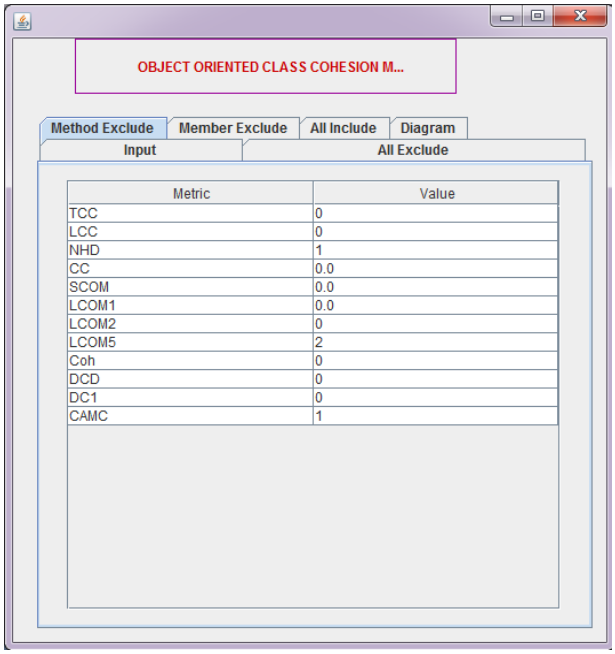
Exclude All will exclude all inherited members of a class in cohesion calculation, **Exclude Method** will exclude inherited methods in cohesion calculation, **Exclude Variable** will exclude inherited methods in cohesion calculation, and **Include All** will include all inherited members of a class in cohesion calculation. **This tool will calculate cohesion for normal classes as well as for special classes also.**

But in this paper, we have just included the result only for special classes like classes those have less than two methods, classes that do not contain attributes, classes whose methods have no parameters.

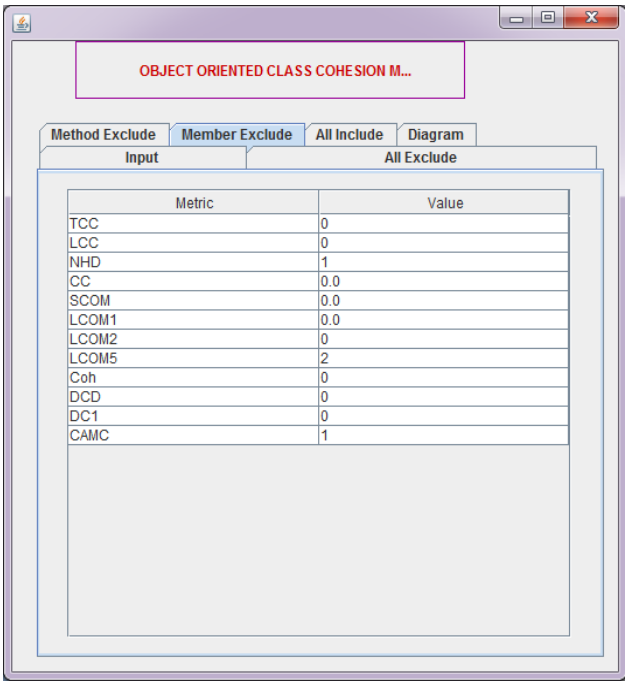
2. The following snapshot shows the cohesion values of class test1 using 12 metrics calculated for the scenario in which both inherited methods and attributes are excluded.



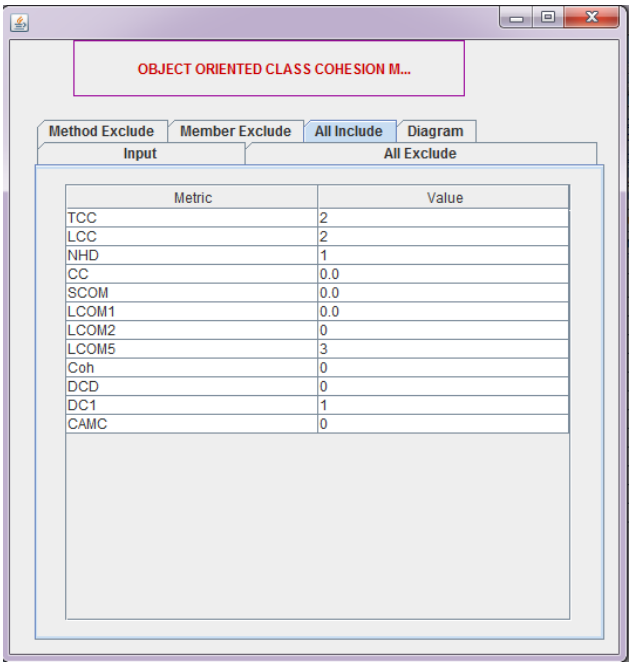
3. The following snapshot shows the cohesion values of class test1 using 12 metrics calculated for the scenario in which **inherited methods** are excluded.



4. The following snapshot shows the cohesion values of class test1 using 12 metrics calculated for the scenario in which inherited attributes are excluded.



5. The following snapshot shows the cohesion values of class test1 using 12 metrics calculated for the scenario in which both inherited methods and attributes are included.



6. The following snapshot shows comparison of four scenarios, which shows that with including inherited members increase the overall cohesion of a class i.e., with including inherited member improve the applicability of maximum metrics of the literature.

ii. *Results*

We have considered 55 classes of special cases and calculated cohesion for classes using each four scenarios for each class. Following bar-chart shows the overall conclusion for four scenarios.

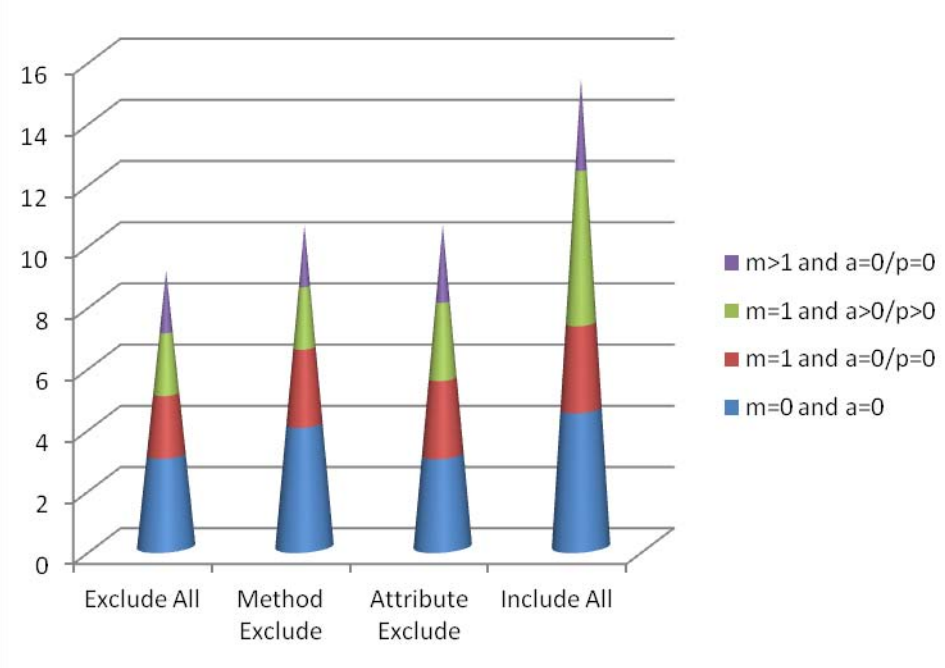


Figure 2.1 : Results of four scenarios for 55 classes

The above chart shows that with including inherited methods and attributes shows the better cohesion results. In it, ***m*** refers to methods, ***a*** refers to attributes and ***p*** refers to parameters. These results shows that including the inherited members the cohesion of a classes increases, thus it improves the applicability of cohesion metrics by using assigned values for special cases in all four scenarios increase the cohesion of maximum number of classes.

III. Conclusion

As there are number of metrics that show undefined values for classes of special cases. This paper proposed a tool for cohesion calculation and studied four scenarios (1) both inherited methods and attributes are excluded, (2) only inherited attributes are excluded, (3) only inherited methods are excluded, and (4) both inherited methods and attributes are included. After studying these scenarios we get the results that relevancy of maximum of metrics increases with including the inherited metrics.

References Références Referencias

1. Al Dallal, Jehad. "Fault prediction and the discriminative powers of connectivity-based object-oriented class cohesion metrics." *Information and Software Technology* 54.4 (2012): 396-416.
2. Ibrahim, Safwat M., et al. "Identification of Nominated Classes for Software Refactoring Using Object-Oriented Cohesion Metrics." *International Journal of Computer Sciences Issues* 9.2 (2012): 68-76.
3. Al Dallal, Jehad. "Improving the applicability of object-oriented class cohesion metrics." *Information and Software Technology* 53.9 (2011): 914-928.
4. http://www.tutorialspoint.com/java/
5. https://netbeans.org/about/
6. Al Dallal, Jehad. "Measuring the discriminative power of object-oriented class cohesion metrics." *Software Engineering, IEEE Transactions on* 37.6 (2011): 788-804.
7. Singh, Kuljit Kaurand Hardeep. "An Investigation of Design Level Class Cohesion Metrics."
8. Al Dallal, Jehad, and Lionel C. Briand. "A Precise method-method interaction-based cohesion metric for object-oriented classes." *ACM Transactions on Software Engineering and Methodology (TOSEM)* 21.2 (2012): 8.
9. Kaya, Mehmet, and James W. Fawcett. "A new cohesion metric and restructuring technique for object oriented paradigm." *Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual. IEEE, 2012.*
10. Yadav, A., and R. A. Khan. "IMPACT OF COHESION ON RELIABILITY." *Journal of Information and Operations Management* 3.1 (2012).
11. Apel, Sven, and Dirk Beyer. "Feature cohesion in software product lines: an exploratory study." *Software Engineering (ICSE), 2011 33rd International Conference on*. IEEE, 2011.
12. Chae, Heung Seok, Yong Rae Kwon, and Doo Hwan Bae. "Improving cohesion metrics for classes by considering dependent instance variables." *Software Engineering, IEEE Transactions on* 30.11 (2004): 826-832.

13. Perepletchikov, Mikhail, Caspar Ryan, and Zahir Tari. "The impact of service cohesion on the analyzability of service-oriented software." *Services Computing, IEEE Transactions on* 3.2 (2010): 89-103.
14. Bieman, James M., and Byung-Kyoo Kang. "Measuring design-level cohesion." *Software Engineering, IEEE Transactions on* 24.2 (1998): 111-124.