



Class Cohesion Metrics in Object Oriented Systems

Amandeep Kaur¹, Puneet Jai Kaur²
University Institute of Engineering & Technology,
Panjab University, Chandigarh-160017.
INDIA

Abstract: In order to improve the quality of an application during the development process, developers use several metrics. These metrics measure the different software attributes such as cohesion, coupling and complexity. To measure the class cohesion several class cohesion metrics have been introduced till date. Cohesion is measured during the design phase to predict software quality. A high cohesive module is easier to understand, modify and maintain in comparison to a less cohesive module. Class cohesion metrics measure the relatedness of the methods and attributes within a class. Several metrics have been proposed in the literature to evaluate class cohesion based on the information that is available during high or low level design phases. This review paper discusses some of the metrics proposed till date.

Keywords: Cohesion; Class Cohesion metrics; fault prediction; object oriented; Cohesive Interactions.

I. Introduction

The aim of software engineering is to develop high quality software. During software design phase, internal software quality attributes such as coupling, cohesion and complexity are considered. All these three attributes are important and special attention must be given to them during software development phase. An Object oriented software is a collection of many classes and each class consist of members called methods and attributes. Class cohesion is an important object oriented quality attribute. Cohesion can be defined as a measure of the degree to which the elements of a module belong together. In object-oriented software systems, cohesion has normally been calculated on per-class or per-object basis. Cohesion is also regarded from a conceptual point of view. In this view, a cohesive module is a crisp abstraction of a concept or feature from the problem domain, usually defined in the requirements. In Object oriented software systems, cohesion is usually measured at the class level and many different OO cohesion metrics have been proposed.

Empirical studies show that class cohesion metrics are useful to predict the fault proneness of classes, to assess software design quality and to identify reusable components. Existing class cohesion metrics mainly fall into two categories –

1. **Low-level design (LLD) metrics:** These metrics are computed at the source code level. For example- LCOM1, LCOM2, TCC, LCC, SCOM, etc.
2. **High-level design (HLD) metrics:** These metrics are computed at the design level. For example- CAMC, NHD, SNHD, SCC, etc.

Software engineers prefer the cohesion metrics that best reflect the human oriented view of cohesion. Human oriented cohesion means collecting tasks into classes based on their logical relatedness from human's viewpoint. This paper analyzes and provides a brief summary of object-oriented class cohesion metrics available till now. This paper is organized as follows- Section II defines the various object oriented class cohesion metrics. Finally Section III concludes the paper and discusses future work.

II. Class Cohesion Metrics

Several metrics have been proposed in the literature to measure cohesion in Object oriented systems at different abstraction levels including method metrics and class metrics. The class cohesion metrics can be classified according to different perspectives such as the types of cohesive interactions considered, the development phase during which they are applicable and the types of methods considered. Cohesive interactions can be of following types:

1. **Method- Method interactions:** the method-method interaction between a pair of methods is defined when both methods access a common attribute, or when one method invokes the other one.
2. **Method-Attribute interactions:** the method-attribute interaction between a method and an attribute is defined when the method accesses the attribute.
3. **Attribute-Attribute interactions:** the attribute-attribute interaction between a pair of attributes is defined when both attributes are accessed by a common method.

Briand et al. define four properties that have been widely used to support the theoretical validation of several proposed class cohesion metrics. Metrics that do not satisfy any of these properties are considered to be ill

defined. The first property that is called the *non-negativity and normalization*, holds that a cohesion measure belongs to a specific interval [0, Max]. Normalization allows for easy comparison between the cohesion of different classes. The second property, that is called *null value and maximum value*, states that the cohesion of a class equals zero if the class has no cohesive interactions, whereas the cohesion of a class is equal to Max if all possible interactions within the class are present. The third property that is called *monotonicity*, holds that the addition of cohesive interactions to the module cannot decrease its cohesion. The fourth property that is called *cohesive modules*, states that the merging of two unrelated modules into one module does not increase the module's cohesion.

In this paper, we will briefly discuss the 20 metrics proposed till now. These are- LCOM1, LCOM2, LCOM3, LCOM4, LCOM5, LSCC, CC, SCOM, Coh, TCC, LCC, DC_D, DC_I, ICBMC, CBMC, PCCC, OL_n, CAMC, NHD and MMAC. CAMC, NHD and MMAC are applicable during the high level design phase, whereas the rest are applicable during the low level design phase. The theoretical validation for most of these metrics has been studied by Jehad Al Dallal (2010). Following are the various class cohesion metrics along with their definitions:

1. **The Lack of Cohesion in Methods (LCOM1):** the metric was proposed by Chidamber and Kemerer in 1991 [13]. LCOM1 is defined by the number of pairs of methods that do not share attributes.
2. **LCOM2:** the metric was proposed by Chidamber and Kemerer in 1994[14].

$$LCOM2 = \begin{cases} P - Q & \text{if } P - Q \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

where,

P=number of pairs of methods that do not share attributes.

Q=number of pair of methods that share attributes.

3. **LCOM3:** the metric was proposed by Li and Henry in 1993[15]. LCOM3 is the number of connected components in graph that represents each method as a node and the sharing of at least one attribute as an edge.
4. **LCOM4:** this metric was proposed by Hitz and Montazeri in 1995[16]. It is similar to LCOM3, and additional edges are used to represent method invocations.
5. **LCOM5:** metric was proposed by Henderson-Sellers in 1996[17].

$$LCOM5 = (kl-a)(kl-l)$$

Where, l=number of attributes

k=number of methods

a=summation of the number of distinct attributes that are accessed by each method in a class.

6. **Low-level design Similarity-based Class Cohesion (LSCC):** this metric was proposed by Al Dallal and Briand in 2010[2].

$$LSCC = \begin{cases} 0 & \text{if } l=0 \text{ and } k>1, \\ 1 & \text{if } (l>0 \text{ and } k=0) \text{ or } k=1, \\ \frac{\sum_{i=1}^l x_i (x_i - 1)}{Lk(k-1)} & \text{otherwise.} \end{cases}$$

Where, l=number of attributes,

k=number of methods

x_i=number of attribute that reference attribute i

7. **Class Cohesion (CC):** The metric was proposed by Bonja and Kidanmariam in 2006[18]. It is defined as the ratio of the summation of the similarities between all pairs of methods to the total number of pairs of methods. the similarity between methods i and j is defined as:

$$\text{Similarity}(i,j) = \frac{|l_i \cap l_j|}{|l_i \cup l_j|}$$

where,

l_i and l_j are the sets of attributes that are referenced by methods i and j, respectively.

8. **Class Cohesion Metric (SCOM):** this metric was proposed by Fernandez and Pena in 2006[11]. SCOM is the ratio of the summation of the similarities between all pairs of methods to the total number of pairs of methods. The similarity between methods i and j is defined as:

$$\text{Similarity}(i,j) = \frac{|l_i \cap l_j|}{|l_i \cup l_j|}$$

$$\frac{\min(|l_i|, |l_j|)}{l}$$

where, l is the number of attributes.

9. **Coh**: the metric was proposed by Briand et al. in 1998[6].

$$\text{Coh} = a/kl$$

Where, l =number of attributes

k =number of methods

a =summation of the number of distinct attributes that are accessed by each method in a class.

10. **Tight Class Cohesion (TCC)**: This metric was proposed by Bieman and Kang in 1995.[19] TCC is the relative number of directly connected pairs of methods, wherein two methods are directly connected if they are directly connected to an attribute. A method m is directly connected to an attribute when the attribute appears within the body of a method that is directly or transitively invoked by method m .
11. **Loose Class Cohesion (LCC)**: This metric was proposed by Bieman and Kang in 1995[19]. LCC defines the relative number of directly or transitively connected pairs of methods, wherein two methods are transitively connected if they are directly or indirectly connected to an attribute. A method m that is directly connected to an attribute j is indirectly connected to an attribute i when there is a method that is directly or transitively connected to both attributes i and j .
12. **Degree of Cohesion-Direct (DC_D)**: This metric was proposed by Badri and Badri in 2004[8]. It defines the relative number of directly connected pairs of methods.
13. **Degree of Cohesion-Indirect (DC_I)**: This metric was proposed by Badri and Badri in 2004[8]. It defines the relative number of directly or transitively connected pairs of methods.
14. **Cohesion Based on Member Connectivity (CBMC)**: Metric was proposed by Chae et al. in 2000[9].

$$\text{CBMC}(G) = F_c(G) \times F_s(G)$$

Where,

$$F_c(G) = |M(G)| / |N(G)|$$

$M(G)$ = number of glue methods in graph G

$N(G)$ = number of non-special methods in graph G

$$F_s(G) = [\sum_{i=1}^n \text{CBMC}(G^i)] / n$$

n = number of child nodes of G .

15. **Improved Cohesion Based on Member Connectivity (ICBMC)**: This metric was proposed by Xu and Zhou in 2001[20].

$$\text{ICBMC}(G) = F_c(G) \times F_s(G)$$

Where,

$$F_c(G) = |M(G)| / |N(G)|$$

$M(G)$ = number of edges in the cut set of G

$N(G)$ = number of non-special methods

represented in graph G multiplied by the number of attributes.

$$F_s(G) = [\sum_{i=1}^n \text{CBMC}(G^i)] / 2$$

16. **Path Connectivity Class Cohesion (PCCC)**: The metric was proposed by Jihad Al Dallal in 2012[4].

$$\text{PCCC} = \begin{cases} 0 & \text{if } l = 0 \text{ and } k > 1, \\ 1 & \text{if } l > 0 \text{ and } k = 0, \\ \text{NSP}(G_c) / \text{NSP}(FG_c) & \text{otherwise.} \end{cases}$$

Where,

NSP = number of simple paths in graph G_c

FG_c = corresponding fully connected graph

17. **OL_n**: This metric was proposed by Yang in 2002[21]. It is defined as the average strength of attributes, wherein the strength of the attribute is the average strength of the methods that reference that attribute. n is the number of iterations that are used to compute OL.
18. **Cohesion Among Methods in a Class (CAMC)**: The metric was proposed by Bansiya et al. in 1999 and by Counsell et al. in 2006[10].

$$\text{CAMC} = a/kl$$

Where,

l = number of distinct parameter types

k = number of methods

a = summation of the number of distinct parameter types of each method in the class.

19. Normalized Hamming Distance (NHD): This metric was proposed by Counsell *et al.* in 2006[10].

$$NHD = 1 - \frac{2}{lk(k-1)} \sum_{j=1}^l x_j(k-x_j)$$

where,

x_j = number of methods that have a parameter of type j .

20. Method-Method through Attributes Cohesion (MMAC): This metric was proposed by Jehad Al Dallal and Briand in 2012[12].

$$MMAC(C) = \begin{cases} 0 & \text{if } l = 0 \text{ and } k = 0, \\ 1 & \text{if } k = 1, \\ \frac{\sum_{i=1}^l x_i(x_i-1)}{lk(k-1)} & \text{otherwise} \end{cases}$$

where,

x_i = number of methods that have a or a return type j .

III. Conclusion

This paper is a review paper that provides a brief description of class cohesion metrics in object oriented systems. The aim of the paper is to help researchers to collectively gain knowledge about each cohesion metric without going into details. The paper discusses 20 class cohesion metrics and provides their brief overview. This can help researchers to learn about class cohesion and then helping them in selecting the metric that will be best suited for their research. This will save their time and will help in carrying out their research in efficient way. In future, research can be done on some newly proposed metrics such as Path Connectivity Class Cohesion (PCCC) metric, Similarity-based Class Cohesion (SCC) metrics, Method-Method through Attributes Cohesion (MMAC) metrics. The metrics can be evaluated for their discriminative powers and other such areas that has not been yet researched. Further study of these three metrics has been left open for future research.

IV. References

- [1] Jehad Al Dallal, "The impact of accounting for special methods in the measurement of object-oriented class cohesion on refactoring and fault prediction activities", *The Journal of Systems and Software*, vol. 85, 2012, 104 2–1057.
- [2] Jehad Al Dallal and Lionel C. Briand, "An object-oriented high-level design-based class cohesion metric", *Information and Software Technology*, vol. 52, 2010, 1346-1361.
- [3] Jehad Al Dallal, "Measuring the discriminative Power of Object-oriented class cohesion metrics", *IEEE transactions on software Engineering*, vol. 37, no.6, November/December 2011.
- [4] Jehad Al Dallal, "Fault prediction and the discriminative powers of connectivity based object-oriented class cohesion metrics", *Information and Software Technology*, vol. 54, 2012, 396-416.
- [5] Letha H. Etzkorn *et al.*, "A comparison of cohesion metrics for object-oriented systems", *Information and Software Technology*, vol. 46, 2004, 677–687.
- [6] Briand, L.C., Daly, J., Wuest, J., "A unified framework for cohesion measurement in object-oriented systems", *Empirical Software Engineering: An International Journal*, vol. 3, 1998, 65–117.
- [7] Etzkorn *et al.*, "A practical look at the Lack of cohesion in methods metric", *Journal of object-oriented programming*, vol. 11, 1998, 27-34.
- [8] Badri M. and Badri L., "A proposal of a new class cohesion criterion: an empirical study", *Journal of object Technology*, vol. 3, 2004, 145-159.
- [9] Chae, H.S., Kwon, Y.R., Bae, D., "A cohesion measure for object oriented classes", *Software: Practice & Experience*, vol. 30, 2000, 1405-1431.
- [10] Counsell, S., Swift, S., Crampton, J., "The interpretation and utility of three cohesion metrics for object oriented design", *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 15, 2006, 123-149.
- [11] Fernández, L., Pena, R., "A sensitive metric of class cohesion", *International Journal of Information Theories and Applications*, vol. 13, 2006, 82–91.
- [12] Jehad Al Dallal and Lionel C. Briand, "A precise method-method interaction-based cohesion metric for object oriented classes", *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 21, 2012.
- [13] Chidamber, S.R., Kemerer, C.F., "Towards a metrics suite for object-oriented design." *Object-Oriented Programming Systems, Languages and Applications (OOPSLA)*, vol. 26, 1991, 197–211.
- [14] Chidamber, S.R., Kemerer, C.F., "A metrics suite for object oriented design.", *IEEE Transactions on Software Engineering*, vol. 20, 1994, 476–493.
- [15] Li, W., Henry, S.M., "Maintenance metrics for the object oriented paradigm." In: *Proceedings of 1st International Software Metrics Symposium*, Baltimore, 1993, 52–60.

- [16] Hitz, M., Montazeri, B., "Measuring coupling and cohesion in object oriented systems.", Proceedings of the International Symposium on Applied Corporate Computing, 1995, 25–27.
- [17] Henderson-Sellers, B., "Object-Oriented Metrics Measures of Complexity", 1996, Prentice-Hall.
- [18] Bonja, C., Kidanmariam, E., "Metrics for class cohesion and similarity between methods.", In: Proceedings of the 44th Annual ACM Southeast Regional Conference, Melbourne, 2006, 91–95.
- [19] Bieman, J., Kang, B., "Cohesion and reuse in an object-oriented system.", In: Proceedings of the 1995 Symposium on Software Reusability, Seattle, Washington, United States, 1995, 259–262.
- [20] Xu, B., Zhou, Y., "Comments on 'A cohesion measure for object-oriented classes'", Software: Practice & Experience, vol. 31, 2001, 1381–1388.
- [21] Yang, X., "Research on Class Cohesion Measures.", M.S. Thesis. Department of Computer Science and Engineering, Southeast University, 2002.