Validating Object-Oriented Class Cohesion Metrics Mathematically

JEHAD AL DALLAL
Department of Information Science
Kuwait University
P.O. Box 5969, Safat 13060
KUWAIT
jehad@cfw.kuniv.edu

Abstract: - Class cohesion refers to the extent to which the members of a class are related. Several class cohesion metrics are proposed in the literature to indicate class cohesion and a few of them are mathematically validated against the necessary properties of class cohesion. Metrics that violate class cohesion properties are not well defined, and their utility as indictors of the relatedness of class members is questionable. The purpose of this paper is to mathematically validate nine class cohesion metrics using class cohesion properties. Results show that the metrics differ considerably in satisfying the cohesion properties; some of them satisfy all properties and others satisfy none.

Key-Words: - object-oriented class, software quality, class cohesion metric, class cohesion.

1 Introduction

Class cohesion is an important object-oriented software quality attribute. It indicates the relatedness between the methods and attributes in a class [1]. Assessing the class cohesion and improving the class quality accordingly during object-oriented design phase allows for lower management costs in the later phases. Software developers use class cohesion measure to assess the quality of their products and to guide the restructuring of poorly-designed classes. A class that has high cohesion cannot be easily split into separate classes. Highly cohesive classes are more understandable, modifiable, and maintainable [2].

Researchers have introduced several metrics to indicate class cohesion. In order to increase the likelihood that a cohesion metric is well defined and serves as a good indicator for the relatedness of the class members, researchers must validate the metric, both theoretically and empirically. Briand et al. [3] proposed four properties that must be satisfied by all class cohesion metrics. If a metric does not satisfy any of these properties, the metric is ill-defined and its usefulness as a cohesion indicator is questionable [3]. These properties provide a supportive underlying theory for the metrics. Empirical validation is necessary to show the usefulness of the metrics. Despite its importance, few researchers focus on the theoretical validation of the metrics. In this paper, we theoretically study the validity of nine class cohesion metrics, using the properties introduced by Briand et al. [3]. We provide mathematical proofs for the metrics that satisfy the cohesion properties, and we provide counter examples, otherwise. Our results show that most of the metrics satisfy all or the majority of the properties.

This paper is organized as follows. Section 2 provides an overview of the class cohesion metrics and necessary properties. In Section 3, the satisfaction of nine class cohesion metrics to the necessary properties is supported or refuted. Finally, Section 4 includes conclusions and a discussion of future work.

2 Related Work

This section overviews the considered class cohesion metrics and other class cohesion metrics. In addition, it includes a summary of the class cohesion necessary properties that all class cohesion metrics must satisfy.

2.1 Overview of class cohesion metrics

Researchers have proposed several class cohesion metrics in the literature. These metrics are based on the use or sharing of the attributes of the class. Bieman and Kang [4] describe two class cohesion metrics, TCC (Tight Class Cohesion) and LCC (Loose Class Cohesion), to measure the relative number of directly-connected pairs of methods and the relative number of directly- or indirectly-connected pairs of methods, respectively. TCC considers two methods to be connected if they share the use of at least one attribute. A method uses an attribute if the attribute appears in the method's body or the method invokes directly or indirectly another method that has the attribute in its body. LCC considers two methods to be connected if they share

the use of at least one attribute directly or transitively. Badri [5] introduces two class-cohesion metrics, DC_D (Degree of Cohesion-Direct) and DC_I (Degree of Cohesion-Indirect),that are similar to TCC and LCC, respectively, but differ by considering two methods connected also when both of them directly or transitively invoke the same method. Briand et al. [3] propose a cohesion metric (called Coh) that computes the cohesion as the ratio of the number of distinct attributes accessed in methods of a class. Fernandez and Pena [6] propose a class cohesion metric, called Sensitive Class Cohesion Metric (SCOM) that considers the cardinality of intersection between each pair of methods. In the metric presented by Bonja and Kidanmariam [7], the degree of similarity between methods is used as a basis to measure class cohesion. The similarity between a pair of methods is defined as the ratio of the number of shared attributes to the number of distinct attributes referenced by both methods. The cohesion is defined as the ratio of the summation of the similarities between all pairs of methods to the total number of possible pairs of methods. The metric is called CC (Class Cohesion).

Bansiya et al. [8] proposed a design-based class cohesion metric called Cohesion Among Methods in a Class (CAMC). In this metric, only the method-method interactions are considered. The CAMC metric uses a parameter occurrence matrix that has a row for each method and a column for each data type that appears at least once as the type of a parameter in at least one method in the class. The value in row *i* and column *j* in the matrix equals 1 when the *i*th method has a parameter of *j*th data type. Otherwise, the value equals 0. The CAMC metric is defined as the ratio of the total number of 1s in the matrix to the total size of the matrix.

Counsell et al. [9] propose a design-based class cohesion metric called Normalized Hamming Distance (NHD). In this metric, only the method-method interactions are considered. The metric uses the same parameter occurrence matrix used by CAMC metric. NHD calculates the average of the parameter agreements between each pair of methods. The parameter agreement between a pair of methods is defined as the number of places in which the parameter occurrence vectors of the two methods are equal. Other related work in the area of software cohesion can be found in [10,11,13,14]

In a previous paper [12], we considered the theoretical validation of six lack-of-cohesion based metrics and this paper is a continuation of that work. In this paper, we validate another nine class cohesion metrics.

2.2 Class cohesion metric properties

Briand et al. [3] defined four properties for cohesion metrics. The first property, Property 1, called non-negativity and normalization, is that the cohesion measure belongs to a specific interval [0, Max]. Normalization allows for easy comparison between the cohesion of different classes. The second property, Property 2, called null value and maximum value, holds that the cohesion of a class equals 0 if the class has no cohesive interactions; the cohesion is equal to Max if all possible interactions within the class are present. The third property, Property 3, called monotonicity, holds that adding cohesive interactions to the module cannot decrease its cohesion. The fourth property, Property 4, called cohesive modules, holds that merging two unrelated modules into one module does not increase the module's cohesion. Therefore, given two classes, c_1 and c_2, the cohesion of the merged class c' must satisfy the following condition: $cohesion(c') \leq \max\{cohesion(c_1), cohesion(c_2)\}$.

3 Theoretical Validation

This section studies the theoretical validation of nine class cohesion metrics. The definition of each metric is overviewed and the satisfaction of the metric to the four class cohesion necessary properties is proved mathematically or disproved, using a counter example.

3.1 TCC and LCC [4], DC_D and DC_I [5], and Coh [3]

Definition: TCC, LCC, DC_D, DC_I, and Coh are defined as the relative number of cohesive interactions. They differ only in their definitions for the cohesive interactions as discussed in Section 2.

Property 1 and Property 2: For the following discussion, the five metrics are referenced as R. The minimum value for R is 0 when the class has no cohesive interactions. The maximum value for R is 1 when the class has the maximum possible number of interactions. Therefore, the five metrics satisfy both Property 1 and Property 2.

Property 3: Since R is defined as the relative number of cohesive interactions, it increases when a cohesive interaction is added to the class model. Therefore, the five metrics satisfy Property 3.

Property 4: To prove the satisfaction of R to Property 4, we introduce the following *numerator-denominator cohesion proving* model:

$$\frac{N(A)}{D(A)} \geq \frac{N(B)}{D(B)} \Rightarrow D(B)N(A) \geq D(A)N(B)$$
$$\Rightarrow [D(B)+D(A)]N(A) \geq D(A)[N(A)+N(B)]$$
$$\Rightarrow \frac{N(A)}{D(A)} \geq \frac{N(A)+N(B)}{D(A)+D(B)}$$

Given the following conditions:
Condition 1: N(M)≤N(A)+N(B)
Condition 2: D(M)≥D(A)+D(B), then

$$\frac{N(A)}{D(A)} \geq \frac{N(A)+N(B)}{D(A)+D(B)} \geq \frac{N(M)}{D(M)}$$

This means that max{cohesion(A),cohesion(B)}≥ cohesion(M). Therefore, if a cohesion metric satisfies Conditions 1 and 2, it satisfies Property 4.
R is a relative metric, and therefore, to prove its satisfaction to Property 4, we prove its satisfaction for Conditions 1 and 2 above as follows:

When unrelated classes A and B are merged into class M, the number of interactions in M is equal to the summation of the number of interactions in classes A and B. Thus, N(M)=N(A)+N(B), which satisfies Condition 1. In addition,

$$D(M) = \frac{(k+m)(k+m-1)}{2} = \frac{k(k-1)}{2} + \frac{m(m-1)}{2} + km$$
$$= D(A) + D(B) + km \Rightarrow D(M) > D(A) + D(B)$$

Hence, R satisfies Condition 2 also and, therefore, the five metrics satisfy Property 4.

3.2 SCOM [6]

Definition: Given a class that has l attributes, the similarity between a pair of methods i and j, which reference the set of attributes I_i and I_j, respectively, is formally defined as follows:

$$Similarity(i,j) = \frac{\left|I_i \cap I_j\right|}{\min(\left|I_i\right|,\left|I_j\right|)} \cdot \frac{\left|I_i \cup I_j\right|}{l}$$

Cohesion is defined as the ratio of the summation of the similarities between all pairs of methods to the total number of possible pairs of methods.
Property 1 and Property 2: The minimum value for SCOM is equal to 0 when none of the methods share common attributes, which includes the case in which none of the methods use any attribute (i.e., the model does not have any cohesive interaction). The maximum value for SCOM is 1 when all methods share all attributes (i.e., the model has all possible cohesive interactions). Therefore, the SCOM metric satisfies both nonnegativity and normalization, as well as null and maximum value cohesion properties.
Property 3: In some cases, when a cohesive interaction is added to the model, the SCOM value of the class decreases to some extent. Figure 1 shows an example (Class A and Class B) for which the metric violates Property 3. This decrease is due to the fact that, in SCOM, the similarity is reversely proportional to the minimum number of attributes used in both methods. In some cases, adding a cohesive interaction increases this number and, consequently, decreases the similarity between some pairs of methods. When this decrease is greater than the increase of the similarity between some other pairs of methods in the class, the SCOM value decreases.
Property 4: To use our *numerator-denominator cohesion proving* model, we adjust the definition of similarity as follows:

$$Similarity(i,j) = \frac{\left|I_i \cap I_j\right|}{\min(\left|I_i\right|,\left|I_j\right|)} \cdot \frac{\left|I_i \cup I_j\right|}{l} = \frac{m_{ij}}{l}$$

Therefore,

$$SCOM(C) = \frac{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} Similarity(i,j)}{\frac{k(k-1)}{2}} = \frac{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} m_{ij}}{\frac{lk(k-1)}{2}} = \frac{N(C)}{D(C)}$$

When two unrelated classes A and B are merged in class M, m_{ij} between each pair of methods in class A and class B does not change, because none of the parameters on which the m_{ij} value depends change. Since classes A and B are unrelated, m_{ij} between any method in class A and any method in class B equals 0 because none of the attributes are shared between the methods. Therefore, N(M)=N(A)+N(B) (i.e., satisfies Condition 1). The following proof shows that SCOM satisfies Condition 2.

$$D(M) = 0.5[(l+n)(k+m)(k+m-1)] = 0.5lk(k-1)$$
$$+ 0.5nm(m-1) + 0.5[(lm+nk)(k+m-1) + m(lk+nk)]$$
$$= D(A) + D(B) + 0.5[(lm+nk)(k+m-1) + m(lk+nk)]$$
$$> D(A) + D(B)$$

As a result, SCOM satisfies the cohesion module property.

3.3 CC [7]

Definition: The similarity between a pair of methods i and j is defined as follows:

$$Similarity(i,j) = \frac{\left|I_i \cap I_j\right|}{\left|I_i \cup I_j\right|}$$, where I_i and I_j are the sets of attributes referenced by methods i and j, respectively. Cohesion is defined as the ratio of the summation of the similarities between all pairs of methods to the total number of possible pairs of methods.
Property 1 and Property 2: The minimum value for CC equals 0 when none of the methods share

common attributes, which includes the case in which none of the methods use any attribute (i.e., the model does not have any cohesive interaction). The maximum value for CC is 1 when all methods share the same set of attributes, which includes the case in which all methods share all attributes (i.e., the model has all possible cohesive interactions). Therefore, CC metric satisfies both nonnegativity and normalization, as well as null and maximum value cohesion properties.

Property 3: CC does not satisfy Property 3 in some cases. That is, when a cohesive interaction is added to a class, the counterintuitive result may be a class with a lower CC value, as depicted in classes C and D, shown in Figure 1. This occurs because the addition of a cohesive interaction may increase the similarities between pairs of methods and decrease the similarities between other pairs of methods. In this case, the cohesion increases if the summation of the similarities between pairs of methods increases, and vice versa.

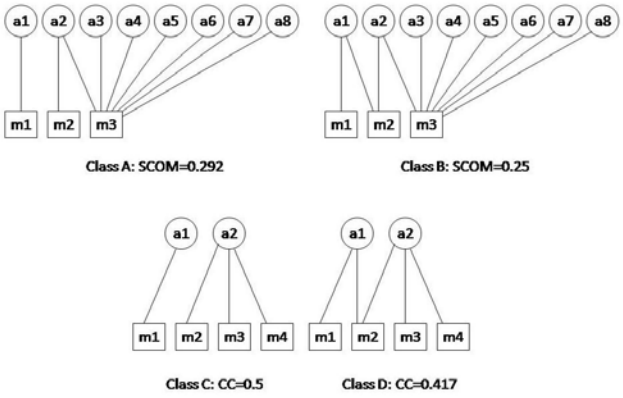

Figure 1: Violation of CC and SCOM for monotonicity property [1]

Property 4: When two unrelated classes A and B are merged in class M, the similarity between each pair of methods in class A and class B does not change. This is because the similarity of a pair of methods is defined as the ratio of the number of shared attributes between both methods to the number of attributes used by both methods. These two numbers remain the same in class M. Since classes A and B are unrelated, there are no similarities between methods in class A and methods in class B. Therefore, N(M)=N(A)+N(B) (i.e., satisfies Condition 1). CC also satisfies Condition 2 (the proof is identical to the corresponding one stated above for R metric). As a result, CC satisfies the cohesion module property.

3.4 CAMC [8]

Definition: The ratio of the total number of 1s in the parameter occurrence matrix to the total size of the matrix.

Property 1 and Property 2: The minimum value for CAMC is $CAMC_{min} = (k+l-1)/kl$ when each parameter type is used by only one method and the class type is used by all methods. The maximum value for CAMC is 1 when all methods have the same parameter types. Since the minimum value for CAMC is greater than zero, the metric does not satisfy Property 1. Since the model of the class used by CAMC cannot be free of cohesive interactions, the null and maximum value property is not applicable.

Property 3 and Property 4: CAMC is defined as the relative number of cohesive interactions in the model that represents the class. Therefore, similar to R metrics, CAMC satisfies the monotonicity and cohesive modules properties.

3.5 NHD [9]

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

where k is the number of methods, l is the number of distinct parameter types, and x_j is the number of 1s in the jth column of the parameter occurrence matrix (i.e., number of methods that use parameter j).

Property 1 and Property 2: NHD metric has the minimum value when each column in the matrix that models the class has the maximum possible disagreements, by setting x_j=k/2 in the NHD formula [9]. In this case, NHD_{min} = (k-2)/[2(k-1)]. The maximum value for NHD is equal to 1 when the matrix contains only 1s (i.e., the class has all possible interactions). Since the minimum value for NHD is greater than zero, the metric does not satisfy Property 1. Since the model of the class used by NHD cannot be free of cohesive interactions, the *null and maximum value* property is not applicable.

Property 3: Adding a cohesive interaction to the class is represented in the matrix by changing two entries in a column in the matrix from 0 to 1 if neither method used the parameter type, or changing one entry from 0 to 1 if one of the methods was using the parameter type. If a column n in the matrix has x_n>k/2, where x_n is the number of 1s in the column, according to the NHD formula, the value of NHD after adding the cohesive interaction is less than it was before adding the cohesive interaction,, which violates Property 3.

Property 4: In some cases, NHD violates cohesion modules property. For example, consider two classes, A and B, where each has two methods; one

of the methods has two parameter types and the other method does not have any parameter types. In this case, the cohesion of each class is equal to 0. When the two classes are merged, the new matrix is 4×4, and the NHD of the merged class is 0.5, which is greater than the NHD value of each of classes A and B.

4 Conclusions and Future Work

This paper shows how to prove or disprove the satisfaction of class cohesion metrics to the class cohesion necessary properties. Table 1 summarizes the results. The results show that five of the considered metrics satisfy all the properties, three satisfy the majority of the properties, and one does not satisfy any property, which raises questions about its ability to indicate class cohesion.

Table 1: Summary of the theoretical validation results

Metric	P1	P2	P3	P4
TCC, LCC, DC_D, DC_I, and Coh	Yes	Yes	Yes	Yes
SCOM	Yes	Yes	No	Yes
CC	Yes	Yes	No	Yes
CAMC	No	N.A.	Yes	Yes
NHD	No	N.A.	No	No

In the future, we plan to theoretically validate other existing class cohesion metrics and empirically explore the relationships between the theoretical and empirical validation results.

Acknowledgment

The author would like to acknowledge the support of this work by Kuwait University Research Grant WI04/07.

References

[1] J. Al Dallal and L. Briand, A precise method-method interaction-based cohesion metric for object-oriented classes, TR, *Simula Research Laboratory*, 2009.

[2] Z. Chen, Y. Zhou, and B. Xu, A novel approach to measuring class cohesion based on dependence analysis, *Proceedings of the International Conference on Software Maintenance*, 2002, pp. 377-384.

[3] L. C. Briand, J. Daly, and J. Wuest, A unified framework for cohesion measurement in object-oriented systems, *Empirical Software Engineering - An International Journal*, Vol. 3, No. 1, 1998, pp. 65-117.

[4] J. M. Bieman and B. Kang, Cohesion and reuse in an object-oriented system, *Proceedings of the 1995 Symposium on Software reusability*, Seattle, Washington, United States, pp. 259-262, 1995.

[5] L. Badri and M. Badri, A Proposal of a new class cohesion criterion: an empirical study, *Journal of Object Technology*, Vol. 3, No. 4, 2004.

[6] L. Fernández, and R. Peña, A sensitive metric of class cohesion, *International Journal of Information Theories and Applications*, Vol. 13, No. 1, 2006, pp. 82-91.

[7] C. Bonja and E. Kidanmariam, Metrics for class cohesion and similarity between methods, Proceedings of the 44th Annual ACM Southeast Regional Conference, Melbourne, Florida, 2006, pp. 91-95.

[8] J. Bansiya, L. Etzkorn, C. Davis, and W. Li, A class cohesion metric for object-oriented designs, *Journal of Object-Oriented Program*, Vol. 11, No. 8, pp. 47-52. 1999.

[9] S. Counsell , S. Swift , and J. Crampton, The interpretation and utility of three cohesion metrics for object-oriented design, *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Vol. 15, No. 2, 2006, pp.123-149.

[10] J. Al Dallal, A design-based cohesion metric for object-oriented classes, *International Journal of Computer Science and Engineering*, 2007, Vol. 1, No. 3, pp. 195-200.

[11] J. Al Dallal, Software similarity-based functional cohesion metric, *IET Software*, 2009, Vol. 3, No. 1, pp. 46-57.

[12] J. Al Dallal, Theoretical validation of object-oriented lack-of-cohesion metrics, proceedings of the *8th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS 2009)*, Cambridge, UK, February 2009.

[13] J. Al Dallal and L. Briand, An object-oriented high-level design-based class cohesion metric, TR, *Simula Research Laboratory*, 2009.

[14] J. Al Dallal, Measuring the discriminative power of object-oriented class cohesion metrics, *IEEE Transactions on Software Engineering*, In press.