A Cohesion Metric Proposal for Object-Oriented Systems: COMIAS

H. ÖZLEM ERTEMEL1, YUNUS EMRE SELÇUK, OYA KALIPSIZ
Computer Engineering Dept.
Yıldız Technical University
A Blok Kat:1 34349 Beşiktaş, İstanbul
TURKEY
ozlem.ertemel@bte.mam.gov.tr, yunus@ce.yildiz.edu.tr, kalipsiz@yildiz.edu.tr

Abstract: Despite inherent difficulties, measuring software quality is important because it makes evaluation and improvement of various aspects of software. In this study, a new cohesion metric named COMIAS is proposed and its first empirical validation is given. Being a structural metric, COMIAS can be used in every stage of software lifecycle.

Key-Words: Cohesion, Object Oriented Metrics

1 Introduction
A complete agreement upon the software quality notion still does not exist and the term software quality carries different meanings for different people [1]. On the developer’s side or from the internal point of view, quality means more accurate estimation, easier testing, better maintainability of costs and delays [2]. From the user’s view or the external point of view, quality means usability, esthetic, understandability etc.

In order to improve the software’s quality, the software quality should be measured at various stages of software lifecycle and the obtained measures should be compared and evaluated. One or more metrics should be chosen before making measurement. At this time, one is faced with the difficulty of finding properties that are agreed upon about software quality, which is already hard to define. In order to begin measuring at the earlier phases of software lifecycle, a metric needs to use the internal quality factors. Coupling [3] and cohesion [4] are fundamental properties used for this aim. In the software that are produced by the proper usage of Object-Oriented Programming (OOP) approach, low coupling and high cohesion is expected [5]. There are lots of metrics for coupling and cohesion in the literature, but more empirical studies are needed as the verification of software metrics is not easy [6].

Despite all difficulties, work on software metrics still continue as they make the evaluation and improvement of software possible from various stakeholders’ viewpoints. When the metrics proposed for coupling and cohesion is investigated, it is evaluated that maturity level has been reached for coupling measures but not for cohesion measures. For this reason, a cohesion metric called COMIAS (Cohesion Method Invocation Attribute Sharing) has been developed. A significant portion of this paper introduces COMIAS.

2 Structural Cohesion Metrics
Cohesion can be defined as the ratio of consistency of the responsibilities of a module with each other. The definition of the module differs according to both the software development approach used and the chosen level to investigate the system. COMIAS, the metric examined in this paper, is a class-level cohesion metric that is developed for the OOP approach.

The following examples can be given for usage objectives of software cohesion metrics: Estimation of software quality and fault-proneness [7,8], software modularization [9,10], identification of reusable components [11], etc.

The proposals of cohesion metrics can be grouped according to the approaches used for metric computation, such as the following: Structural cohesion metrics, semantic cohesion metrics [12], entropy-based cohesion metrics [13], slice-based metrics [14], metrics based on data-mining [15], metrics for specific types of applications like knowledge-based systems [16] and distributed systems [17].

Structural cohesion metrics basically consider the usage of attributes of a class by the methods of this class. The relationships between methods of the class are defined from this viewpoint. The structural cohesion metrics proposed in the literature consider one or more structural properties such as attribute usage, attribute sharing, method invocation, parameter type intersection, parameter type usage, etc. This paradigm is based on Bunge’s ontology [4]. Bunge defines the similarity of two objects as the intersection of these objects’ properties. The similarity degree of the methods of a class is the basis of the structural cohesion metrics’ computation. Many metrics exist in the literature such as LCOM [18] and LCOM* [19] which use Bunge’s ontology.

1 H.Ö.Ertemel is also affiliated with the IT Institute of TÜBİTAK Marmara Research Center
3 Developed Metric: COMIAS

Tree steps are followed while calculating the cohesion of a class using the metric COMIAS. In the first step, the relationship between methods are examined by looking into the method invocations and the first graphic \((G_{xA})\) is obtained. In the second step, the relationship between methods are examined by looking into the attribute usage and the second graphic \((G_{xB})\) is obtained. In the third step, the graphics created in the previous steps are combined into one graphic \((G_x)\). In the remaining sections these steps are examined in detail.

3.1 Examination of Method Invocations

While examining the method invocations, the direct relations are sought first. A directed line is created between methods from the calling method to the called method if one of the methods calls the other. The obtained graphic \(G_{x1}\) can be expressed as below:

\[
E = \{ <m, n> \in V \times V \mid (\text{m calls n}) \} \quad (1)
\]

An example graphic of \(G_{x1}\) can be seen in Fig. 1.

![Fig. 1. An example direct connection graphic \((G_{x1})\)](image)

To find indirect connections, the directed lines are used in graphic \(G_{x1}\). By looking Fig. 1, it can easily be seen that method M1 is calling method M2 and method M2 is calling method M3. Consequently, method M1 is calling method M3 indirectly. Then, a directed line is created from method M1 to method M3. The connection graphic \(G_{x2}\) obtained in this way can be expressed as such: If a directed graphic \(G_{x2}(V, E)\) and \(V = M_x\),

\[
E = \{ <m, l> \in V \times V \mid (m \text{ calls } n) \land (n \text{ calls } l) \land (m \text{ calls } n \text{ or } n \text{ calls } l) \} \quad (2)
\]

The graphic \(G_{x2}\) obtained from graphic \(G_{x1}\) can be seen in Fig. 2.

![Fig. 2. An example indirect connection graphic \((G_{x2})\)](image)

As the last step of investigation of method invocations, the direct and indirect connected methods are shown by combining the previously obtained graphics \(G_{x1}\) and \(G_{x2}\) into one graphic \(G_{xA}\). The graphic \(G_{xA}\) can be seen in Fig. 3 which is obtained from example graphics \(G_{x1}\) and \(G_{x2}\).

![Fig. 3. An example method invocation graphic \((G_{xA})\)](image)

3.2. Examination of Attribute Usage

A graphic showing a sample case of method-attribute relation is drawn in order to see attribute usage by methods. If a method uses an attribute, an undirected line is created between the method and the attribute. An example graphic \(G_{xB}\) which is obtained in this way can be seen in Fig. 4.

![Fig. 4. An example direct attribute usage graphic \((G_{xB})\)](image)

3.3. Method Invocation and Attribute Usage

The graphic \(G_x\) is obtained by combining the graphics \(G_{xA}\) and \(G_{xB}\) created in the previous steps into one graphic. By the examination of indirect method invocations and determination of indirect attribute usage, the final version of \(G_x\) is obtained. A \(G_x\) graphic which is generated by using the example graphics given in previous steps can be seen in Fig. 5.

![Fig. 5. An example final relationship graphic \((G_x)\)](image)

3.4. Definition of the COMIAS Cohesion Metric

Let \(X\) be a class and \(I_x = X\) be the set of attributes and \(M_x = X\) be the set of methods of the class. When a directed graphic \(G_x(V, E)\) is considered, the method invocation relation between two methods can be defined as below:
The attribute usage relation between two methods can be found by looking at the graphic $G_x$ as below;

$$\text{Attr.sha}(M_i, M_k) = n \frac{\text{number of attributes shared by } M_i \text{ and } M_k}{\text{number of attributes of } M_i}$$

(4)

The relation between two methods, $\text{Relation}(M_i, M_k)$, is the sum of method invocation and attribute sharing between methods $M_i$ and $M_k$, where $M_i \in M_x$, $M_k \in M_x$, and $i \neq k$. In this case, it is said that:

$$\text{Relation}(M_i, M_k) = \text{Method.inv}(M_i, M_k) + \text{Attr.sha}(M_i, M_k).$$

(5)

The total relation value of the class is found from the relation of two methods as below:

$$\text{Total.Relation}(X) = \sum_{i=1}^{m-1} \sum_{k=i+1}^{m} \left( \text{Relation}(M_i, M_k) \right)$$

(6)

If Relation($M_i$, $M_k$) is high, the relation between two methods is said to be high and the relation value of the class increases. The total relation value of the class, Total_Relation($X$), is high, then the cohesion of the class increases.

It is needed to find the maximum relation value for the class and find the ratio of this value to Total_Relation($X$) value in order to make COMIAS normalized. The maximum relation value of the class, Max_Relation($X$), is found by the sum of the maximum method invocation relation and the maximum attribute sharing relation.

$$\text{Max.Relation}(X) = \text{Max}(\text{Method.inv}) + \text{Max}(\text{Attr.sha})$$

(7)

Max (Method_inv) in Equation 7 represents the maximum method relation in the class. This means an invocation exists in every ($M_i, M_k$) relation and this can be found by the number of all possible ($M_i, M_k$) relations;

$$M_i \in M_x \text{ and } M_k \in M_x, \ i \neq k.$$  

(8)

If a class $X$ has $m$ methods, then the number of all possible ($M_i, M_k$) relations will be $m(m-1)/2$. The existence of an invocation in every relation gives the maximum number of method invocations. The maximum number of method invocations can be expressed as below;

$$\text{Max}(\text{Method.inv}) = \frac{m(m-1)}{2} \times 1$$

(9)

Similarly, maximum attribute sharing relation in the class can be defined as:

$$\text{Max}(\text{Attr.sha}) = \text{means all attributes of the class are shared in every } (M_i, M_k) \text{ relation;}$$

$$M_i \in M_x \text{ and } M_k \in M_x, \ i \neq k.$$  

(10)

If an $X$ class has a number of attributes, then the maximum attribute sharing will be attribute count a times the number of all possible ($M_i, M_k$) relations, $m(m-1)/2$. The maximum number of attribute sharing can be expressed as below;

$$\text{Max}(\text{Attr.sha}) = \frac{m(m-1)}{2} \times a$$

(11)

The value of Max_Relation($X$) which is needed to normalize the metric can be found as below;

$$\text{Max.Relation}(X) = \left( \frac{m(m-1)}{2} \times 1 \right) + \left( \frac{m(m-1)}{2} \times a \right)$$

(12)

According to above explanations, the proposed metric can be formulated as below;

$$\text{COMIAS} = \frac{\text{Total.Relation}(X)}{\text{Max.Relation}(X)}$$

(13)

### 3.5. Exemplification of the COMIAS Metric

In this section, the computation of COMIAS metric is demonstrated by using the example given in Fig.6. When Fig.6 is examined, it is expected intuitively that the cohesion of Fig.6.b should be better than of Fig.6.a. If the indirect connection between methods M1 and M3 doesn’t taken into account, then this expected result will not be obtained.

![Fig. 6. Final relation graphics of an example system obtained by not considering indirect relations (a) and considering indirect connections (b).](image)

Fig. 6. Final relation graphics of an example system obtained by not considering indirect relations (a) and considering indirect relations (b).

If indirect connections aren’t taken into account, the COMIAS metric for the graphic in Fig.6.a will be calculated as below;

$$\text{COMIAS} = \frac{\left(1 + 0\right) + \left(0 + 0\right) + \left(1 + 0\right)}{\frac{3 \times (3 - 1) \times 1}{2} + \frac{3 \times (3 - 1) \times 0}{2}} = \frac{2}{3}$$

(14)

COMIAS metric can be calculated as below for the graphic that is obtained by considering indirect connections in Fig.6.b:
Therefore, it could be seen that if indirect connections are taken into account in the example system in Fig.6, then the cohesion will be calculated higher. Cohesion is expected to be measured as maximum in Fig.6.b because all possible connections exist in this figure. This example also implies that if interactions among the components of the class increases, the cohesion will increase as expected.

4 Points of Consideration

In this section, the points considered while developing COMIAS are verified.

**Hypothesis 1:**

“If maximum or minimum cohesion is expected intuitively, the metric should indicate this expected maximum or minimum cohesion as well.”

![Fig. 7. Example graphics for maximum relation](image)

When Fig.7 is examined, one intuitively expects the metric to give the maximum cohesion value “1” for both cases a and b. Metric computation for the graphic in Fig.7.a (m = 3, a = 2) is as follows:

\[
COMIAS = \frac{(1 + 2) + (1 + 2) + (1 + 2)}{3 \times \frac{(3 - 1)}{2} \times 1} + \frac{3 \times \frac{(3 - 1)}{2} \times 2}{2} = 1 \quad (15)
\]

As a result, the proposed metric gave the same value, depicting maximum cohesion for both of the graphics in Fig.7, in parallel with the intuition.

![Fig. 8. Example graphics for minimum relation](image)

In the case of Fig.8, the intuition leads one to expect minimum cohesion as there are no method interactions involved. Metric computation for the graphic in Fig.8.a (m = 3, a = 2) is as follows:

\[
COMIAS = \frac{(0 + 0) + (0 + 0) + (0 + 0)}{3 \times \frac{(3 - 1)}{2} \times 1} + \frac{3 \times \frac{(3 - 1)}{2} \times 2}{2} = 0 \quad (16)
\]

When COMIAS is computed for graphic in Fig.8.b (m = 2, a = 1), it is calculated as:

\[
COMIAS = \frac{(0 + 0)}{2 \times \frac{(2 - 1)}{2} \times 1} + \frac{2 \times \frac{(2 - 1)}{2} \times 1}{2} = 0 \quad (17)
\]

As a result, the proposed metric gave the same value, depicting minimum cohesion for both of the graphics in Fig.8, in parallel with the intuition.

When the examples given in this section are examined, it is also seen that metric gave the same results for classes which are expected intuitively having the same cohesive degree. Indefinite, ambiguous and undefined conditions didn’t occur in the results.

**Hypothesis 2:**

“If the indirect connections are not taken into account, the cohesiveness cannot be measured precisely. In this case, a metric should indicate lower cohesion.”

When Fig.9 is examined, one intuitively expects the metric to give the cohesion value better for Fig.9.b than for Fig.9.a. However, if the indirect relations are not taken into account, this expected result cannot be seen.

![Fig. 9. Example graphics for implying the importance of considering indirect relations](image)

If the indirect connections are not taken into account, metric computation for the graphic in Fig.9.a is calculated as follows:
If the indirect connections are not taken into account, metric computation for the graphic in Fig.9.b is calculated as follows:

\[
\text{COMIAS} = \left( \frac{(1 + 0) + (0 + 0) + (1 + 0)}{3 \times (3 - 1)} \times 1 \right) + \left( \frac{3 \times (3 - 1)}{2} \times 0 \right) = \frac{2}{3}
\]  

(20)

When the indirect connections are not taken into account, the metric gave the same result. However, it is desirable for a good metric to differentiate these results.

If the indirect connections are taken into account, metric computation for the graphic in Fig.9.a stays as the same, as there are no indirect connections in this graphic.

If the indirect connections are taken into account, metric computation for the graphic in Fig.9.b is calculated as follows:

\[
\text{COMIAS} = \left( \frac{(1 + 0) + (0 + 0) + (1 + 0)}{3 \times (3 - 1)} \times 1 \right) + \left( \frac{3 \times (3 - 1)}{2} \times 0 \right) = \frac{2}{3}
\]  

(21)

As a result, the proposed metric gave the value in parallel with the intuition for the graphic in Figure 9.b. This example shows the importance of considering the indirect connections: Inclusion of indirect relations into COMIAS has enabled the metric to produce results in parallel with the intuition.

**Hypothesis 3:**

“If the interactions among the components of the class increase, the metric should indicate higher cohesion.”

Table 1. The results obtained by COMIAS metric for continuing Zemberek versions

<table>
<thead>
<tr>
<th>Version</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>0.133</td>
</tr>
<tr>
<td>0.6.3</td>
<td>0.303</td>
</tr>
<tr>
<td>0.6.4</td>
<td>0.310</td>
</tr>
<tr>
<td>2.2.1</td>
<td>0.337</td>
</tr>
</tbody>
</table>

5 Empirical Validation of COMIAS

The current implementation of COMIAS can measure the cohesion of classes coded by the Java programming language. Therefore a candidate to be used for the empirical validation of the proposed metric should be coded in Java as well. Moreover, various versions of this candidate software should exist to make a comparison. Zemberek, an open source natural language processing library for Turkish, satisfies these requirements.

The user views have been used in order to compare the obtained results produced by COMIAS with the real world data. An interview with Ahmet Afşin AKIN, a member of the developer group of Zemberek, revealed that in general the quality increases as the version number increases. It is expected from the proposed metric to give parallel results with this opinion. As a matter of fact, it can be seen that the results given in Table 1 meet the expectations.

6 Conclusion

As computing becomes more pervasive, a software defect can lead to bad results such as financial loss, time delay or the loss of human life in even worse cases. Therefore, software systems should operate as error free and consistent systems. Increasing demand on software quality introduces us the “quality” characteristic as an important factor for a software product.

Cohesion, an important quality factor for Object Oriented Programming paradigm, can be described as the degree of connectivity among the elements of a
single module. It is expected that cohesion to be high and coupling to be low in a well designed system. The lack of cohesion for a class suggests this class to be split into two or more classes. Low cohesion increases the complexity and thus causing the increase in the probability of occurrence of errors during the development phase. None of the cohesion metrics in the literature has been accepted as a standard [20]. One of the reasons for this is that more empirical validation studies about proposed metrics are needed [6].

The current state of the COMIAS metric is encouraging as it gives results that are normalized and parallel with intuitions. COMIAS uses both direct and indirect method relations in its calculations, however, it does not take inherited methods into account. For these reasons, continuation of empirical works and the ability to use inherited methods in calculations are planned as future work.

References:


