A Proposal of Class Cohesion Metrics Using Sizes of Cohesive Parts

Hirohisa Aman
Ehime University
40 PUBLICATIONS **73** CITATIONS
SEE PROFILE

Matu-Tarow Noda
Ehime Campus Information Service, Co. Ltd.(…
67 PUBLICATIONS **304** CITATIONS
SEE PROFILE

A Proposal of Class Cohesion Metrics Using Sizes of Cohesive Parts

Hirohisa AMAN[1], Kenji YAMASAKI[1], Hiroyuki YAMADA[1], Matu-Tarow NODA[1]
[1]*Department of Computer Science, Ehime University,*
Bunkyo-cho 3, Matsuyama, Ehime 790–8577, Japan
aman@cs.ehime-u.ac.jp

Abstract. Cohesion is one of traditional and important software attributes. This could be used to assess object-oriented class design. Several metrics have been proposed in order to capture class cohesion in terms of connections among methods within a class. These metrics are based on (1) the number of sets of connected methods, or (2) the density of method connections within the class, but they do not consider sizes of sets of connected methods. In this paper, two new metrics for class cohesion have been proposed, which are focusing on sizes of sets of connected methods, with considering strength of method connection. These proposed metrics are analytically verified using a mathematical framework.

1 Introduction

Cohesion is one of software attributes representing the degree to which the components are functionally connected within a software module[1]-[3]. This notion could be applied to object-oriented software: An object class corresponds to a module, whose attributes and methods correspond to module components. In general, highly cohesive class tends to have high maintainability, reusability and reliability[4], so that class cohesion would be one of criteria used for assessing class design. In order to measure class cohesion, several metrics have been proposed : 'Lack of Cohesion in Methods' (*LCOM*)[5]-[8], 'Tight Class Cohesion (*TCC*)', 'Loose Class Cohesion (*LCC*)'[7], 'Information flow-based Cohesion (*ICH*)'[9]; see [4] for details of them. These metrics are based on (1) the number of sets of connected methods, or (2) the density of method connections within the class, but they do no consider sizes of sets of connected methods. In this paper, two new metrics for class cohesion have been proposed, which are focusing on sizes of sets of connected methods, with considering strength of method connection. The proposed metrics satisfy the mathematical properties of cohesion metrics proposed by Briand and others[10].

2 Models and Metrics

Preliminary to developments of metrics, we present several underlying definitions.

Definition 1 (binary relation on methods).

Given a class. Let M be the set of methods within the class. We define a binary relation $S : M \rightharpoondown M$ as

$$S = \{ \ (u, v) \in M \times M \mid u \text{ invokes } v \ \} .$$

Then we define the *reflective transitive closure* $S^* : M \rightharpoondown M$ as

$$S^* = \left\{ \ (u, v) \in M \times M \ \middle| \ (u = v) \vee \left(\bigvee_{n \geq 1} u \ S^n v \right) \ \right\} , \tag{1}$$

where $S^n = S^{n-1} \circ S \ \ (n \geq 2)$, and $S^1 = S$. ("$\circ$" indicates the composition of relations[11].)

■

Based on S^*, we represent indirect relationships between method and attribute, as well as direct relationships between them.

Definition 2 (method accesses to attributes).

Given a class. Let M be the set of methods, and A be the set of attributes, within the class. For any $m \in M$, $a \in A$, we define the following predicates *ac*, *wr* and *re* :

$$ac(m, a) \overset{def}{\iff} \exists m' \in M \text{ s.t. } [\ (m \ S^* m') \wedge (m' \text{ accesses } a) \ ] , \tag{2}$$

$$wr(m, a) \overset{def}{\iff} \exists m' \in M \text{ s.t. } [\ (m \ S^* m') \wedge (m' \text{ writes data onto } a) \ ] , \tag{3}$$

$$re(m, a) \overset{def}{\iff} \exists m' \in M \text{ s.t. } [\ (m \ S^* m') \wedge (m' \text{ reads data from } a) \ ] . \tag{4}$$

■

The predicate *ac* considers not only direct access to attribute but also indirect access to attribute via access-methods. *wr* and *re* are focusing on data-writing and data-reading through attribute access. Using the above three predicates, we introduce two graph models.

Definition 3 (weak-connection graph).

Given a class. Let M be the set of methods, and A be the set of attributes, within the class. We define *weak-connection graph* as an undirected graph $G_w(V, E)$, where $V = M$ and

$$E = \{ \ \{u, v\} \in M \times M \mid \exists a \in A \text{ s.t. } \ (\ ac(u, a) \ \wedge \ ac(v, a) \ ) \ \}. \tag{5}$$

■

Definition 4 (strong-connection graph).

Given a class. Let M be the set of methods, and A be the set of attributes, within the class. We define *strong-connection graph* as a directed graph $G_s(V, E)$, where $V = M$ and

$$E = \{ \ (u, v) \in M \times M \mid \exists a \in A \text{ s.t. } \ (\ wr(u, a) \ \wedge \ re(v, a) \ ) \ \}. \tag{6}$$

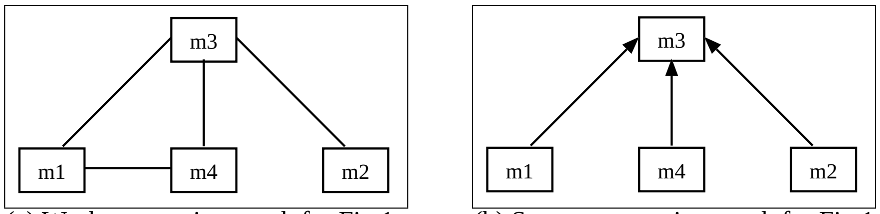
■

When two or more methods access to one attribute, those methods seem to share the attribute. The weak-connection graph represents attribute-sharings.

Furthermore, accesses to attributes include data-readings and data-writings. If a method writes data onto an attribute, and another method reads data from the attribute, then a dependent relationship might be occurred between the methods. Such relationship is emphasized by the strong-connection graph.

```
public class C {
    private int a1, a2;
    public void m1(int x){ a1 = x; }
    public void m2(int y){ a2 = y; }
    public int m3(){ return a1 + a2; }
    public void m4(){ m1(0); }
}
```

Figure 1: An example of class written in Java.



(a) Weak-connection graph for Fig.1 (b) Strong-connection graph for Fig.1

Figure 2: Weak-connection graph and strong-connection graph for the class shown in Fig.1.

For example, consider a class shown in Fig.1. According to Defs.3 and 4, we obtain the weak-connection graph and the strong-connection graph shown in Fig.2 :

$ac(\text{m1}, \text{a1})$, $ac(\text{m3}, \text{a1})$ and $ac(\text{m4}, \text{a1})$ are true, so that the edges {m1,m3}, {m1,m4} and {m3,m4} are shown in Fig.2(a);

'm4 invokes m1', 'm1 writes data onto a1' and 'm3 reads data from a1', so that $wr(\text{m1}, \text{a1})$, $wr(\text{m4}, \text{a1})$ and $re(\text{m3}, \text{a1})$ are true. Thus the edges (m1,m3) and (m4,m3) are shown in Fig.2(b);

etc.

While the strong-connection graph is sensitive to directions of attribute accesses, the weak-connection graph is insensitive to them. In other words, the weak-connection graph seems to be optimistic about method connections, but the strong-connection graph looks pessimistic about them.

Using the weak-connection graph and the strong-connection graph, we propose two metrics for class cohesion.

Definition 5 (Optimistic Class Cohesion (OCC)).

Given a class C. Let M be the set of methods, and A be the set of attributes, within C. Consider the weak-connection graph $G_w(V, E)$, where $V = M$ and E is in Eq.(5). Let $n = |M|$.

For each method $m_i \in M (i = 1, \ldots, n)$, let $R_w(m_i)$ be the set of methods which are reachable by m_i on $G_w(V, E)$:

$$\begin{aligned} R_w(m_i) = \{\ & m_j \in M \mid \exists m_{k_1}, \ldots, m_{k_p} \in M \ \text{ s.t.} \\ & \{m_{k_s}, m_{k_{s+1}}\} \in E \ \ (s = 1, \ldots, p-1), \ \ m_i = m_{k_1}, \ m_j = m_{k_p}, \ i \neq j\ \}. \end{aligned} \tag{7}$$

We define *Optimistic Class Cohesion* (*OCC*) for class C as

$$OCC(C) = \begin{cases} \max\limits_{i=1,\ldots,n} \left[\dfrac{|R_w(m_i)|}{n-1} \right], & (n > 1), \\ 0, & (n = 1). \end{cases} \tag{8}$$

■

$|R_w(m_i)|/(n-1)$ denotes the percentage of methods to be reachable by m_i on the weak-connection graph ('-1' means excepting itself). *OCC* is the maximum value of them. That is, *OCC* quantifies the maximum extent of attribute-sharings among methods within the class. For example, in the case of Fig.2,

$$\begin{aligned} R_w(\text{m1}) &= \{\text{m2, m3, m4}\}, \quad R_w(\text{m2}) = \{\text{m1, m3, m4}\}, \\ R_w(\text{m3}) &= \{\text{m1, m2, m4}\}, \quad R_w(\text{m4}) = \{\text{m1, m2, m3}\}. \end{aligned}$$

Thus $|R_w(\text{m1})| = \cdots = |R_w(\text{m4})| = 3$. We can calculate as $OCC = \max[3/(4-1)] = 1.0$.

Definition 6 (Pessimistic Class Cohesion (PCC)).

Given a class C. Let M be the set of methods, and A be the set of attributes, within C. Consider the strong-connection graph $G_s(V,E)$, where $V = M$ and E is in Eq.(6). Let $n = |M|$.

For each method $m_i \in M (i = 1,\ldots,n)$, let $R_s(m_i)$ be the set of methods which are reachable by m_i on $G_s(V,E)$:

$$\begin{aligned} R_s(m_i) = \{\ & m_j \in M \mid \exists m_{k_1},\ldots,m_{k_p} \in M \ \text{ s.t.} \\ & (m_{k_s}, m_{k_{s+1}}) \in E \ \ (s = 1,\ldots,p-1), \ \ m_i = m_{k_1}, \ m_j = m_{k_p}, \ i \neq j \ \}. \end{aligned} \tag{9}$$

We define *Pessimistic Class Cohesion* (*PCC*) for class C as

$$PCC(C) = \begin{cases} \max\limits_{i=1,\ldots,n} \left[\dfrac{|R_s(m_i)|}{n-1} \right], & (n > 1), \\ 0, & (n = 1). \end{cases} \tag{10}$$

■

$|R_s(m_i)|/(n-1)$ denotes the percentage of methods to be reachable by m_i on the strong-connection graph. *PCC* is the maximum value of them. That is, *PCC* quantifies the maximum extent of dependent relationships among methods within the class. This would be the maximum size of highly cohesive part of the class. For example, in the case of Fig.2,

$$R_s(\text{m1}) = \{\text{m3}\}, \ \ R_s(\text{m2}) = \{\text{m3}\}, \ \ R_s(\text{m3}) = \{\}, \ \ R_s(\text{m4}) = \{\text{m3}\}.$$

Thus $|R_s(\text{m1})| = |R_s(\text{m2})| = |R_s(\text{m4})| = 1$ and $|R_s(\text{m3})| = 0$. We can calculate as $PCC = \max[1/(4-1), 0/(4-1)] = 1/3$.

3 Analytical Evaluation of Metrics

Briand, Morasca and Basili have proposed a mathematical framework (BMB framework) including properties to be satisfied by several types of software metrics[10]. The supported types of metrics are 'size', 'length', 'complexity', 'coupling' and 'cohesion'. Note that BMB framework provide necessary conditions of software metrics, because the framework does not include the all properties to be satisfied by those metrics.

In BMB framework, a software is represented by a graph model in which vertexes are corresponding to software components, and edges are corresponding to coupling relationships between the components. BMB framework have suggested the following four properties to be satisfied by cohesion metrics. For the sake of convenience, we will write the cohesion of class C as $\mu(C)$.

Property 1 : $\forall C \ \mu(C) \in [0, max]$, where max is a positive constant number.

Property 2 : Let $G(V, E)$ be the graph model of a class C. Then $E = \phi \implies \mu(C) = 0$.
Property 3 : Consider two classes C and C' whose models are $G(V, E)$ and $G(V, E')$, respectively. Then $E \subseteq E' \implies \mu(C) \leq \mu(C')$.
Property 4 : Consider two classes C_1 and C_2 whose models are $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, respectively. Let C_{12} be a class whose model is $G_{12}(V_1 \cup V_2, E_{12})$, where $E_1 \cup E_2 \subseteq E_{12}$, i.e., C_{12} is composed of C_1 and C_2. Then $(E_1 \cup E_2 = E_{12}) \wedge (E_1 \cap E_2 = \phi) \implies \max[\ \mu(C_1),\ \mu(C_2)\ ] \geq \mu(C_{12})$.

3.1 OCC

OCC satisfies the above four properties. We now show the proof for the property 4 only, for the lack of space; the proofs for the remaining properties could be derived from the definitions of *OCC* and R_w easily.

Proof for Property 4 :

Let C_1 has n_1 methods m_i^1 $(i = 1, \ldots, n_1)$, C_2 has n_2 methods m_i^2 $(i = 1, \ldots, n_2)$, and C_{12} has n_{12} methods m_i^{12} $(i = 1, \ldots, n_{12})$, respectively. Let $R_w^1(m_i^1)$ be the set of reachable methods by m_i^1 on $G_1(V_1, E_1)$, $R_w^2(m_i^2)$ be the set of reachable methods by m_i^2 on $G_2(V_2, E_2)$, and $R_w^{12}(m_i^{12})$ be the set of reachable methods by m_i^{12} on $G_{12}(V_1 \cup V_2, E_{12})$, respectively.

(i) Case $n_1 = n_2 = 1$: From the definition and property 2, $OCC(C_1) = OCC(C_2) = OCC(C_{12}) = 0$. Thus, $\max[OCC(C_1), OCC(C_2)] = OCC(C_{12})$.

(ii) Case $n_1 > 1$ or $n_2 > 1$: This case may be $n_1 + n_2 \geq n_{12}$. Let $n_1 + n_2 - n_{12} = p\ (\geq 0)$. We will describe $m_{n_1-p+k}^1 = m_k^2$ $(k = 1, \ldots, p)$, without loss of generality. In constructing C_{12} from C_1 and C_2, we add no edge to their weak-connection graphs. Thus

$$R_w^{12}(m_i^{12}) = \begin{cases} R_w^1(m_i^1), & (i = 1, \ldots, n_1 - p), \\ R_w^1(m_i^1) = R_w^2(m_{i-n_1+p}^2), & (i = n_1 - p + 1, \ldots, n_1), \\ R_w^2(m_{i-n_1+p}^2), & (i = n_1 + 1, \ldots, n_{12}). \end{cases}$$

Since $n_1, n_2 \leq n_{12}$,

$$\frac{|R_w^{12}(m_i^{12})|}{n_{12} - 1} \leq \frac{|R_w^1(m_i^1)|}{n_1 - 1}, \quad (i = 1, \ldots, n_1), \quad \text{and}$$
$$\frac{|R_w^{12}(m_i^{12})|}{n_{12} - 1} \leq \frac{|R_w^2(m_j^2)|}{n_2 - 1}, \quad (i = n_1 - p + 1, \ldots, n_{12};\ j = i - n_1 + p).$$

Thereby we get

$$\max_{i=1,\ldots,n_1} \left[\frac{|R_w^{12}(m_i^{12})|}{n_{12} - 1} \right] \leq \max_{i=1,\ldots,n_1} \left[\frac{|R_w^1(m_i^1)|}{n_1 - 1} \right] = OCC(C_1), \quad \text{and}$$
$$\max_{i=n_1-p+1,\ldots,n_{12}} \left[\frac{|R_w^{12}(m_i^{12})|}{n_{12} - 1} \right] \leq \max_{\substack{i=n_1-p+1,\ldots,n_{12} \\ (j=1,\ldots,n_2)}} \left[\frac{|R_w^2(m_j^2)|}{n_2 - 1} \right] = OCC(C_2).$$

From these,

$$\begin{aligned} OCC(C_{12}) &= \max_{i=1,\ldots,n_{12}} \left[\frac{|R_w^{12}(m_i^{12})|}{n_{12} - 1} \right] \\ &= \max \left\{ \max_{i=1,\ldots,n_1} \left[\frac{|R_w^{12}(m_i^{12})|}{n_{12} - 1} \right], \max_{i=n_1-p+1,\ldots,n_{12}} \left[\frac{|R_w^{12}(m_i^{12})|}{n_{12} - 1} \right] \right\} \\ &\leq \max \{\ OCC(C_1),\ OCC(C_2)\ \}, \end{aligned}$$

for all classes such that $E_1 \cup E_2 = E_{12}$ and $E_1 \cap E_2 = \phi$. ■

3.2 PCC

In the above proof, replace '*OCC*' with '*PCC*', 'R_w' with 'R_s', and, 'weak-connection graph' with 'strong-connection graph', respectively. Then we can get the proof for *PCC*.

4 Conclusions and Future Works

We have proposed two metrics for class cohesion, *OCC* (*Optimistic Class Cohesion*) and *PCC* (*Pessimistic Class Cohesion*). If two or more methods access to one attribute directly or indirectly, those methods seem to share the attribute. Such sharing is a kind of connections among methods. *OCC* quantifies the maximum extent of such connections within a class. This would be the maximum size of cohesive part of the class.

When methods access to attributes, those accesses include data-readings and data-writings. By focusing on such differences in accesses, we can consider dependent relationships among methods, which would be strong connections among methods. *PCC* quantifies the maximum extent of such dependent relationships within a class. This would be the maximum size of highly cohesive part of the class.

According to BMB framework, we have showed that *OCC* and *PCC* satisfy necessary conditions of cohesion metrics.

The future works include the followings : (1) to study relationships among *OCC*, *PCC* and existing metrics using many experimental data; (2) to discuss the usefulness of BMB framework in the software engineering discipline.

References

[1] G.J. Myers, Software Reliability-Principles and Practices, John Wiley & Sons, Inc. (1976).

[2] M. Page-Jones, The Practical Guide to Structured Systems, 2nd ed., Yourdon Press (1988).

[3] E. Yourdon and L. Constantine, Structured Design, Englewood Cliffs, Prentice Hall (1979).

[4] L.C. Briand, J.W. Daly and J. Wüst, A Unified Framework for Cohesion Measurement in Object-Oriented Systems, Empirical Software Eng. : An Int'l J. **3**(1) (1998) 65–117.

[5] S.R. Chidamber and C.F. Kemerer, A metrics suite for object-oriented design, IEEE Trans. on Software Eng. **20**(6) (1994) 476–493.

[6] M. Hitz and B. Montazeri, Chidamber and Kemerer's metrics suite : a measurement theory perspective, IEEE Trans. on Software Eng. **22**(4) (1996) 267–271.

[7] J.M. Bieman and B.-K. Kang, Cohesion and reuse in an object-oriented system, Proc. ACM Symp. Software Reusability (1995) 259–262.

[8] B. Henderson-Sellers, Software Metrics, Prentice Hall (1996).

[9] Y.-S. Lee, B.-S. Liang, S.-F. Wu and F.-J. Wang, Measuring the coupling and cohesion of an object oriented program based on information flow, Proc. International Conference on Software Quality (1995).

[10] L.C. Briand, S. Morasca and V.R. Basili, Property-based software engineering measurement, IEEE Trans. on Software Eng. **22**(1) (1996) 68–86.

[11] D.F. Stanat and D.F. McAllister, Discrete mathematics in Computer Science, Prentice-Hall (1977).

[12] H. Aman, H. Yamada and M.T. Noda, A Proposal of Class Cohesion Metrics focusing on Sizes of Cohesive Parts, Technical Report of IEICE **KBSE2001–19** (2001).