Chapter - 4

Cognitive Weighted Methods Per Class
CHAPTER - 4

COGNITIVE WEIGHTED METHODS PER CLASS

4.1 Introduction

The AOP paradigm was introduced as a complement to the Object-Oriented paradigm to improve the overall quality of programs [Ivi, 13]. It aims to increase modularity by allowing the separation of cross-cutting concerns, forming a basis for aspect-oriented software development. AOP includes programming techniques and tools that support the modularization of concerns at the level of the source code, while "Aspect-Oriented Software Development (AOSD)" refers to a whole engineering discipline. AOP is a concept, and as such, it is not bound to a certain programming language or programming paradigm. Hence, it helps to overcome the shortcomings of all languages that use single, hierarchical decomposition. This may be procedural, object-oriented, or functional.

The Java implementation of AOP is called AspectJ and has been created at Xerox PARC. AOP is considered to be the best and popular choice among software professions where crosscutting features are needed to implement [Pra, 14]. The term “metrics” is used to denote a set of specific measurements taken on a particular item or process. The goal of software metrics is, to improve understanding of a product or process. Metrics attempt to measure a particular aspect of a software system. These aspects can range from traditional measurements such as the number of lines of code to the relationships created between components in a system [Alo, 12a].

The complexity measures based on cognitive informatics is in the development phase. Cognitive complexity measures represent the human effort needed to perform a
task or difficulty in understanding the software code [Ami, 14]. AspectJ has no CWMC metric to measure the different type of advice proposed by various researchers. So, there is a need for cognitive weighted, Weighted Methods per Class (WMC) for the Aspect level Advice measurement. Hence the main goal of the research is to define a CWMC metric to measure the Complexity of Advice.

4.2 Motivation

Parthipan et al. have proposed weighted Advice per Aspect (WAA) metric [Par, 14]. The value of WAA metric is calculated as the sum of the cognitive weight of all advice types in an aspect. The advice is classified into three types, namely, before, after and around advice. The weights assigned for the advice types are based on the complexity. The before and after advice are less complex compared to the around advice. Since around advice can optionally bypass the execution of a join point, the cognitive weight assigned to it has a higher value compared to other two advice types. The drawback of the WAA metric is that it is not proved statistically.

WMC counts some methods or advice in a given module proposed by Ceccato et al. [Cec, 04] and Kotrappa Sirbi et al. [Kot, 10]. It is an equivalent of the WMC metric from CK Metrics suite [Chi, 94].

4.3 AspectJ

Aspect-Oriented Programming is a way of modularizing crosscutting concerns much like object-oriented programming is a way of modularizing common concerns. AspectJ is an implementation of Aspect-Oriented Programming for Java.

AspectJ adds to Java just one new concept, a joinpoint and that adds a name to an existing Java concept. It adds to Java only a few new constructs: pointcuts, advice,
inter-type declarations and aspects. Pointcuts and advice dynamically affect program flow, an inter-type declaration (Introduction) statically affects a program's class hierarchy, and aspects encapsulate these new constructs.

- **Aspect**
  An aspect is modularized employment of a crosscutting concern, and its definition contains when, where and how to invoke a concern.

- **Joinpoint**
  A joinpoint is a well-defined point in the program flow.

- **Pointcut**
  A pointcut picks out certain join points and values at those points.

- **Advice**
  A piece of advice is a code that is executed when a join point is reached.

Joinpoint, pointcut, advice is the dynamic parts of AspectJ.

- **Inter-type declaration**
  AspectJ also has different kinds of inter-type declarations that allow the programmer to modify a program's static structure, namely, the members of its classes and the relationship between classes.

### 4.4 Advice

Advice defines crosscutting behaviour. It is defined regarding pointcuts. The code of a piece of advice runs at every join point picked out by its pointcut. Exactly how the code runs depends on the kind of advice.

AspectJ supports three types of advice as BeFore Advice (BFA), AFter Advice (AFA), and ARound Advice (ARA). The type of advice determines its interaction with the joinpoints.
- **Before Advice (BFA)** Advice that executes before a join point, but which cannot prevent execution flow proceeding to the join point (unless it throws an exception).

- **After Advice (AFA)** Advice to be executed regardless of the means by which a join point exits (normal or exceptional return).
  - **After returning advice:** Advice to be executed after a join point completes normally: for example if a method returns without throwing an exception. To use this advice, use @AfterReturning annotation.
  - **After throwing advice:** Advice to be executed if a method exits by throwing an exception. To use this advice, use @AfterThrowing annotation.

- **Around Advice (ARA)** Advice that surrounds a joinpoint such as a method invocation. This is the most powerful kind of advice. To use this advice, use @Around annotation.

### 4.5 Formulation of the CWMC Metric

#### 4.5.1 Existing Metric

WMC captures the internal complexity of a module regarding the number of implemented functions. A more refined version of this metric can be obtained by giving different weights to operations with different internal complexity.

\[
\text{Cmpx (wmc)} = \sum_{x=0}^{X} \text{cmpx(ad)}
\]

where \( x \) is the total number of Advice in the aspect and \( \text{CMPX (ad)} \) is the advice complexity of a class. Here all types of advice are assigned by weight value 1 to represent its complexity.
4.5.2 Proposed Metric

Researchers have proposed various metrics for AOP systems. WMC is one of the known metrics proposed by Ceccato et al., [Cec, 04]. It is an equivalent of the WMC metric from CK Metrics suite [Chi, 94]. WMC counts a number of methods or advice in a given module. This metric does not consider the various types of advice. The proposed metric, CWMC considers the cognitive complexity of the different types of advice, such as before, after, around.

4.6 Experimental Design

This section discusses the CWMC metric, empirical data collection, statistics analysis, and its implication. WMC metric is selected for AO software. This metric is used to find the complexity of various types of advice using Cognitive Approach.

In this section, an experiment is conducted to assign cognitive weight to the various types of advice discussed in section 4.4. A comprehension test has been conducted for a group of students to find out the time taken to understand the complexity of aspect-oriented program concerning different types of advice. The group of students selected had sufficient exposure in analysing the Aspect-oriented programs, as they had been given special courses in AspectJ language. 60 students were selected to participate in the comprehension test.

The time taken by students to comprehend the programs was recorded after the completion of each program. The time taken for comprehension of all these programs was noted and the mean time to comprehend was calculated. Five different programs have been administered in each case (Before Advice, After Advice, and Around Advice), and totally fifteen different mean timings were recorded. Average time was
calculated for each program from the individual time taken by students, shown in Table 4.1. From Table 4.1, the calculated average comprehension time for Before Advice as 17.5833, After Advice as 23.8333 and Around Advice as 29.0167 for program1 (p1) and for program2 (p2) the average calculated comprehension time for Before Advice as 17.25, After Advice as 23.6167 and Around Advice as 28.0833 and so on.

Table 4.1 Categorized Mean Comprehension Time

<table>
<thead>
<tr>
<th>Programs</th>
<th>Average Comprehension Time (In Minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BFA</td>
</tr>
<tr>
<td>P1</td>
<td>17.5833</td>
</tr>
<tr>
<td>P2</td>
<td>17.25</td>
</tr>
<tr>
<td>P3</td>
<td>16.6833</td>
</tr>
<tr>
<td>P4</td>
<td>17.2167</td>
</tr>
<tr>
<td>P5</td>
<td>16.5333</td>
</tr>
</tbody>
</table>

The average comprehension time for programs is listed in Table 4.1. These programs are based on AspectJ.

For each type of Advice, mean was selected as a measure of central tendency and the standard deviation as a measure of dispersion. The mean and standard deviation value was calculated from Table 4.1 for BFA and AFA and getting the Mean values for BFA and AFA as 0.2842 and 0.3907. From Table 4.2, the standard deviation was calculated from the mean values and attains the average as 0.8904 and 0.4248 for BFA and AFA. Also, the mean and standard deviation value were calculated for ARA and tabulated in Table 4.3. The average value getting from the Table 4.1 was transformed. Table 4.2 and Table 4.3 illustrates the statistical
computation of different types of advice. The Figure 4.1 shows that the graph for Average comprehension time for BFA and AFA.

Table 4.2 Summary Statistics for CWBFA & CWAFA Values

<table>
<thead>
<tr>
<th>Programs</th>
<th>Before Advice (In hours)</th>
<th>After Advice (In hours)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean x(BFA)</td>
<td>Std. Dev.</td>
<td>Mean x(AFA)</td>
</tr>
<tr>
<td>P1</td>
<td>0.2931</td>
<td>1.7202</td>
<td>0.3972</td>
</tr>
<tr>
<td>P2</td>
<td>0.2875</td>
<td>1.7333</td>
<td>0.3936</td>
</tr>
<tr>
<td>P3</td>
<td>0.2781</td>
<td>1.5676</td>
<td>0.3889</td>
</tr>
<tr>
<td>P4</td>
<td>0.2869</td>
<td>1.5524</td>
<td>0.3925</td>
</tr>
<tr>
<td>P5</td>
<td>0.2756</td>
<td>1.3079</td>
<td>0.3814</td>
</tr>
<tr>
<td>Average</td>
<td>0.2842</td>
<td>0.8904</td>
<td>0.3907</td>
</tr>
</tbody>
</table>

Figure 4.1 Value of Mean Time for BFA and AFA
Table 4.2 shows the summary statistics for CWBFA and CWAFA and Fig. 4.1 shows the empirical data for a mean time both in BFA and AFA. The ratio between BFA and AFA is less than 1 (i.e., 0.7274). It indicates that mean value of the time is higher in the AFA than in the BFA.

Table 4.3 Summary Statistics for CWBFA & CWARA Values

<table>
<thead>
<tr>
<th>Programs</th>
<th>Before Advice (In hours)</th>
<th>Around Advice (In hours)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean x(BFA)</td>
<td>Std. Dev.</td>
<td>Mean x(ARA)</td>
</tr>
<tr>
<td>P1</td>
<td>0.2931</td>
<td>1.7201</td>
<td>0.4836</td>
</tr>
<tr>
<td>P2</td>
<td>0.2875</td>
<td>1.7333</td>
<td>0.4681</td>
</tr>
<tr>
<td>P3</td>
<td>0.2781</td>
<td>1.5676</td>
<td>0.4669</td>
</tr>
<tr>
<td>P4</td>
<td>0.2869</td>
<td>1.5524</td>
<td>0.4589</td>
</tr>
<tr>
<td>P5</td>
<td>0.2756</td>
<td>1.3079</td>
<td>0.4561</td>
</tr>
<tr>
<td>Average</td>
<td>0.2842</td>
<td>0.8904</td>
<td>0.46672</td>
</tr>
</tbody>
</table>

Figure 4.2 Value of Mean Time for BFA and ARA
Table 4.3 shows the summary statistics for CWBFA and CWARA and Figure 4.2 shows the empirical data for a mean time both in BFA and ARA. The ratio between BFA and ARA is less than 1 (i.e., 0.6089). It indicates that mean value of the time is higher in the ARA than in the BFA. The Figure 4.2 shows that the graph for Average comprehension time for BFA and ARA.

Types of advice are compared by mean and standard deviation. The ratio between the mean time for CWBFA and CWAFA and the mean time for CWBFA and CWARA are used for calculation. When the value of this metric greater than 1, it indicates that the mean value of the metric is higher in the aspects than in the class.

4.7 Cognitive Weighted Methods per Class

The proposed metric called CWMC, which considers the cognitive complexity of the different types of advice such as before, after and around. The existing WMC metric proposed by Ceccato et al. [Cec, 14] and Kotrappa Sirbi et al. [Kot, 10] uses the count of methods. It is an equivalent of the WMC metric from CK Metrics suite [Chi, 94]. This metric does not consider the various types of advice. CWMC can be calculated by using the Equation as follows,

\[
\text{CWMC} = ((\text{BFA} \times \text{CW}_{\text{BFA}}) + (\text{AFA} \times \text{CW}_{\text{AFA}}) + (\text{ARA} \times \text{CW}_{\text{ARA}})) \quad \text{... (4.1)}
\]

where,

- BFA - Before Advice
- AFA - After Advice
- ARA - Around Advice
- \(\text{CW}_{\text{BFA}}\) - Cognitive Weight of Before Advice
- \(\text{CW}_{\text{AFA}}\) - Cognitive Weight of After Advice
- \(\text{CW}_{\text{ARA}}\) – Cognitive Weight of Around Advice
The Cognitive Weight Factor of each type of Advice is calibrated in Table 4.4 using the method discussed in the Empirical Metric Data Collection. The weight value is calculated based on the mean value. Average mean value of each type of advice is, BFA = 0.2842, AFA = 0.3907, ARA = 0.4667. To normalize the mean value to get appropriate weight value. Since the values are less than one, it is multiplied by 10. BFA = 2.842, AFA = 3.907, ARA = 4.667 then the values are rounded as BFA=3, AFA=4, ARA=5. Finally, the weight value is calculated by dividing the values by 3 to reduce the range of values and getting the Cognitive Weight of BeBefore Advice (CW_{BFA}) as 1, Cognitive Weight of AFter Advice (CW_{AFA}) as 1.33, and Cognitive Weight of ARound Advice (CW_{ARA}) as 1.66. The weight value of each type of advice is given as,

<table>
<thead>
<tr>
<th>Advice</th>
<th>Cognitive Weight Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CW_{BFA}</td>
<td>1</td>
</tr>
<tr>
<td>CW_{AFA}</td>
<td>1.33</td>
</tr>
<tr>
<td>CW_{ARA}</td>
<td>1.66</td>
</tr>
</tbody>
</table>

The following section explains the calculation of CWMC metric using a case study.

4.8 Illustration

The proposed CWMC metric given by Equation 4.1 is evaluated with AspectJ programs are given in Appendix A. From the program given in Appendix A, the existing and proposed metric values be calculated and shown below.
WMC

\[ \text{Cmpx (wmc)} = \sum_{x=0}^{n} \text{cmpx(ad)} \]

where \( x \) is the total number of Advice in the aspect and CMPX (ad) is the advice complexity of class. Here all types of advice are assigned by weight value 1 to represent its complexity. The program is given in Appendix A having 3 advices.

Then,

\( x = 3 \)

So,

\[ \text{CMPX (WMC)} = 3 \]

CWMC

\[ \text{CWMC} = ((\text{BFA*CW}_{\text{BFA}}) + (\text{AFA*CW}_{\text{AFA}}) + (\text{ARA*CW}_{\text{ARA}})) \]

\[ \text{CWMC} = ((1*1) + (1*1.33) + (1*1.66)) \]

\[ \text{CWMC} = 1 + 1.33 + 1.66 = 4 \]

Table 4.5 Advice Complexity metric value for the sample program

<table>
<thead>
<tr>
<th>Program#</th>
<th>Existing Metric Value (WMC)</th>
<th>Proposed Metric Value (CWMC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

4.9 Theoretical Validation

Fenton’s properties are used for theoretical validation of CWMC, as it has been proposed in the evaluation framework developed by [Fen, 97]. Those properties were used for the data collection process and described as follows:
• **Accuracy** The higher the difference between the actual data and measured data and the lower is the accuracy and vice-versa. It implies that the metric value for WMC is lower than the proposed CWMC value.

• **Replicability** means that the analysis can be done at different times by different people using the same setting. Data are taken from rural and urban PG students at different times.

• **Correctness** According to the metrics definition data was collected. The value of CWMC is collected and calculated through the WMC metric.

• **Precision** Data is expressed by some decimal places. The less decimal place shows a lower accuracy. The decimal place of the data is high (i.e. 0.6089). So it shows a higher accuracy.

• **Consistency** It counts the differences in the metric values when collected using different tools by different people. Accordingly, we found the difference between existing metric - WMC and proposed metric – CWMC by giving different programs by different students.

### 4.10 Comparative Study

The comparison of the proposed metric, Cognitive Weighted Method per Class (CWMC) was made with the most widely accepted metric, WMC proposed by Ceccato et al. [Cec, 04]. WMC is defined as a total number of methods or advice in the aspect. The proposed metric, CWMC, is one step ahead of the existing metric, WMC, as it includes the complexity that arises due to the different types of Advice. A comprehension test with the rural area and urban area degree students was conducted to calculate the cognitive weights. Five different programs in AspectJ were given to
the students. The time taken by the students in comprehending the programs was recorded. From that experiment, the average time taken was calculated. After the comprehension test, metric values were calculated using the proposed metric and WMC. The complexity values revealed are shown in Table 4.6.

From the Table 4.6, it is found that resulting values of CWMC is larger than the WMC were, the weight of each advice was assumed to be 1 while in CWMC, the cognitive weights for different advices were considered in computing the complexity. Thus, it provides more realistic complexity by considering the different types of Advice.

<table>
<thead>
<tr>
<th>Program#</th>
<th>Existing Metric Value (WMC)</th>
<th>Proposed Metric Value (CWMC)</th>
<th>Mean Comprehension Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>13</td>
<td>13.5</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>12</td>
<td>13.6</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>14</td>
<td>16.7</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>19</td>
<td>21.5</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>25</td>
<td>23.2</td>
</tr>
</tbody>
</table>

A correlation analysis is performed using Table 4.6 between WMC Vs Mean Comprehension Time which yielded $r = 0.851339$ and CWMC Vs Mean Comprehension Time which yielded $r = 0.948259$. The corresponding graph is shown in Figure 4.3. This confirms that CWMC is more positively correlated to Mean Comprehension Time than WMC. From the result, it is concluded that CWMC is a better indicator of the complexity of the classes with different types of Advice.
4.11 Conclusion

A CWMC metric for measuring the class level complexity has been formulated. The complexity of the class includes the methods (Advices) complexity of the class. CWMC includes the cognitive complexity due to different types of Advice. CWMC has proven that the complexity of the class getting affected, which is based on the cognitive weights of the various types of Advice. The assigned cognitive weight of the various types of Advice is validated using the comprehension test and found that the cognitive load to understand the ARA is greater than BFA and AFA. The metric is evaluated through an experimental and proved to be a better indicator of the class level complexity. Having discussed the proposed CCMS-AOP framework that is used to measure the complexity of AO software concerning the cognitive complexity, the following chapter attempts to define and validate the proposed Cognitive Weighted Coupling on Advice Execution (CWCAE) metric.