Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

An Estimation Of Measuring The Complexity Of Advice Using Fuzzy Logic Technique

Dr. G. Arockia Sahaya Sheela[1] and Dr.K.R.Martin[2*]

[1] Department of Computer Science, Holy Cross College (Autonomous), (Affiliated to Bharathidasan University), Trichy – 620002 Tamil Nadu, India.

[2*] Department of Computer Science, St. Joseph's College (Autonomous), (Affiliated to Bharathidasan University), Trichy – 620002 Tamil Nadu, India

Abstract

Aspect Oriented Software Development (AOSD) is a novel and growing paradigm that promotes the separation of concerns that are dispersed across the system. Aspect-oriented software development uses software metrics to measure desired software and software development features (AOSD). If software metrics are used throughout the development phase, maintenance costs may be kept under control. Because Aspect Oriented Software Development is a new paradigm, a thorough examination of maintainability and its associated metrics is required. The fuzzy inference system is used in this work to determine the advice complexity of an aspect oriented system. The Cognitive Weighted Method per Class (CWMC) measure is used to determine the difficulty of guidance.

Keywords Aspect Oriented Programming (AOP), Advice, metric, Weighted Methods per Class (WMC), Aspect Oriented Software Development (AOSD), Fuzzy Inference System (FIS).

I Introduction

By extending the traditional object-oriented programming (OOP) methodology, Aspect Oriented Programming (AOP) enhances code reprocess across different object hierarchies. Object-oriented programming can be utilised with AOP. AspectJ for Java is an Aspect-Oriented Programming implementation written in Java.

AspectJ introduces only one new concept to Java: a join point, which gives an old Java notion a new name. Maintainability is a software attribute that has a significant impact on service quality. The less effort/cost the software maintenance cycle consumes, the higher the highly effective of the software.

This paper discusses how fuzzy logic may be used to analyse software maintainability. For adjusting the performance of software maintainability, a novel model based on a fuzzy inference system is presented. This exam will use the Cognitive Weighted Method per Class (CWMC).

These variables are unrelated to one another and are used to assess software maintainability. These factors are subjected to a Triangular Membership Function (TriMF), in MATLAB, the Mamdani Fuzzy Inference System is described.

In this paper, we have defined and evaluated the CWMC metrics which is used for the assessment of software maintainability of aspect oriented software.

II Literature Review

A number of metrics for AOP systems have been proposed by academics. Ceccato et al. [3] and Kotrappa Sirbi et al. [4] suggest WMC as one of the measures.
It's an equivalent of the WMC metric from CK Metrics suite [5].WMC counts number of methods or advice in a given module [1]. This metric does not consider the various types of advice. Weighted Advices per Aspect (WAA) metric is proposed by Parthipan, Senthil Velan, and Chitra Babu [6] .

The total cognitive weight of all advise kinds in an aspect is used to compute the WAA metric's value. Before, after, and around counsel are the three forms of advise that are commonly given. The cognitive weights attributed to the different forms of advise are determined by their cognitive complexity. When opposed to the surrounding guidance, the before and after advice is less complicated.

Since advice is available, it's a good idea to take it the execution of a join point, the cognitive weight assigned to it has a higher value compared to other two advice types.
The WAA measure has a flaw in that they did not verify their metric using a statistical technique, therefore the data is inaccurate. The data does not fulfil the Fenton et al. [7] characteristics due to empirical data gathering.

Pradeep Kumar Singh and Om Prakash Sangwan [9], an Assessment of Software Testability using Fuzzy Logic Technique has been provided for the Aspect Oriented software based on the use of fuzzy logic to evaluate the testability of the software. For adjusting the performance of software testability, a novel model based on a fuzzy inference system is presented.

III CWMC Metric

Researchers have developed a number of measures for AOP systems. One of the metrics that has been presented by Ceccato et.al [3] and Kotrappa Sirbi et.al [4] is WMC. It's an equivalent of the WMC metric from CK Metrics suite [5].WMC counts number of methods or advice [2] in a given module. This metric does not consider the various types of advice. The proposed metric called CWMC [2], which considers the cognitive complexity of the different types of advices, such as before, after, around.

CWMC =((BFA*WFBFA)+(AFA*WFAFA)

+ (ARA*WFARA)) ——— equ 1

Where,

BFA -Before Advice

AFA -After Advice

ARA -Around Advice

WFBFA-Weighting Factor of Before Advice

WFAFA-Weighting Factor of After Advice

WFARA-Weighting Factor of Around Advice

IV Fuzzy Logic

Fuzzy logic is a systematic method for resolving issues that are quantitatively difficult to comprehend. It's a tool for dealing with ambiguity and inconsistency [8]. It's a lot less the fuzzy model may be created with minimal data and is dependent on previous data [10] [11]. Using the various membership functions, the fuzzy system takes ambiguous assertions and imprecise data and makes choices, as illustrated in Figure 5.1.

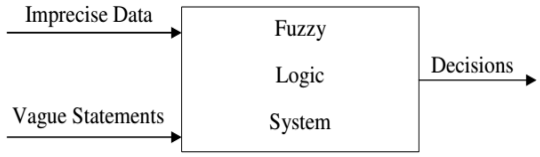

Figure 5.1 Fuzzy Logic Systems

The complexity of the AO system is a measure of different sorts of advices in this suggested work.

They are Before, After and Around. The suggested fuzzy logic takes all of these aspects into account as inputs and then uses a rule base to calculate a precise complexity value. All input values are divided into three categories: low, medium, and high. The output maintainability is divided into three categories: low, medium, and high. All possible combinations of inputs are used to construct a rule foundation. The Fuzzy Inference System (FIS) comprises the module seen in Figure 5.

1) FIS Editor: It displays information about the fuzzy inference system, which allows the system to tackle complicated issues.
2) Membership Function Editor: This programme defines the forms of all of the membership functions associated with each component.
3) Rule Editor: It is used to change the rules that define how the problem behaves.
4) Rule Viewer: This tool is used to view the rule base, or how an individual behaves.
5) Surface Viewer: This tool is used to view the graph based on the system's inputs and outputs.

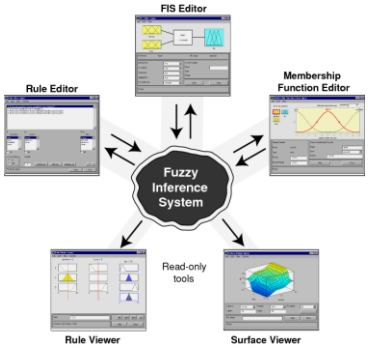

Figure 5.2 Fuzzy Inference Systems

V Simulation and Experimentation

In this section, the trained fuzzy inference system and proposed fuzzy model with inputs i.e. Before, After and Around Advice value to predict CWMC metric complexity are presented in Figure 5.3.

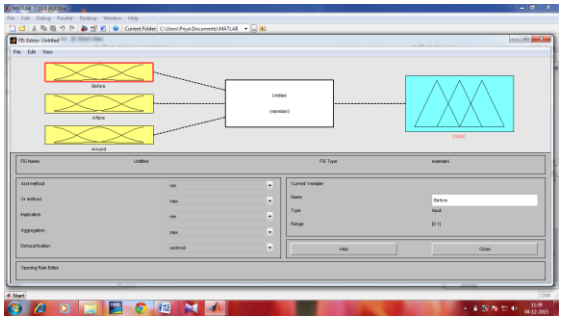

Figure 5.3 Aspect Complexity Model with 3 inputs, 1 output, 27 rules

All inputs are splitted into 3 club functions: low, medium, and high, and the CWMC output is split into 3 categories: low, medium, and high. The inputs and outputs are scaled between [0 1], and the triangular membership function is used to classify them. It is shown in the next section.

Low [0 0.2 0.4], Medium [0.3 0.5 0.7], High [0.6 0.8 1] are the inputs, whereas Low [0 0.2 0.4], Medium [0.3 0.5 0.7], and High [0.6 0.8 1] are the outputs.

All 27 rules that were generated and entered into the rule base reflect all conceivable input combinations, or 33 (27) sets. Table 5.1 contains information on the proposed fuzzy system.

Table 5.1 Details of the System used

Name of the system	Aspect_complexity
Type of model	“Mamdani”
MATLAB version	7.10.0
No. of inputs	3

No. of outputs	1
No. of rules	27
No. of MF's used in inputs	3
No. of MF's used in outputs	1
Range	[0, 1]

Figure 5.4 shows the membership function for Before Advice as input 1 in the fuzzy logic tool.

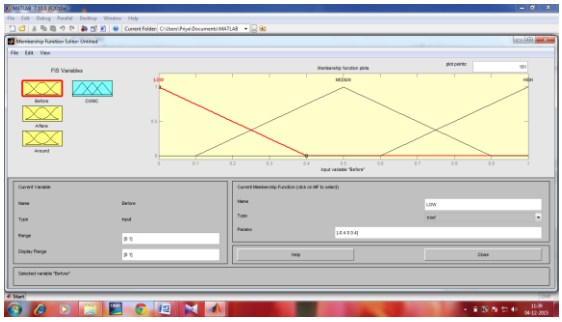

Figure 5.4 Membership function for Before Advice

Figure 5.5 shows the membership function for After Advice as input 2 in the fuzzy logic tool.

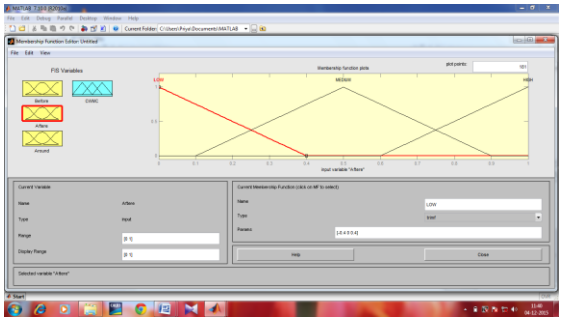

Figure 5.5 Membership function for After Advice

Figure 5.6 indicates the club feature for Around Advice as use 3 withinside the fuzzy logic tool.

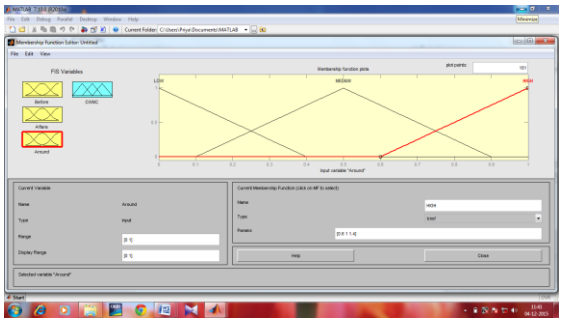

Figure 5.6 Membership function for Around Advice

Figure 5.7 shows the membership function for the CWMC metric value as output 1 in the fuzzy logic tool.

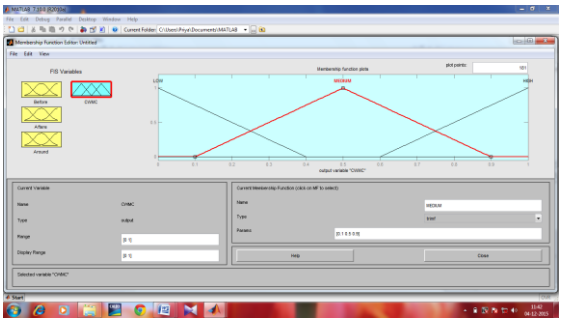

Figure 5.7 Membership function for CWMC metric

As illustrated in Figure 5.8, all 27 rules are constructed, and a rule base is formed that represents all conceivable combinations of inputs, i.e. (27) sets.

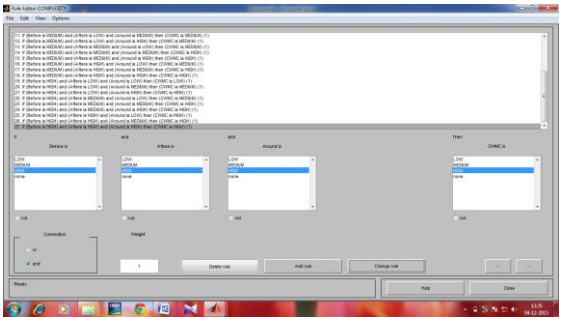

Figure 5.8 Possible Combinations of Inputs (27 Rules)

CWMC is calculated using three input factor values using a rule viewer (see Figure 5.9).

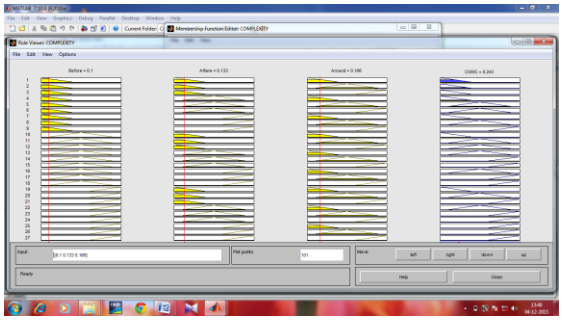

Figure 5.9 Rule Viewer

The CWMC metric value of AO Software is discussed in this article. It analyses the elements that influence the CWMC metric value and establishes a connection between these factors and the CWMC metric value.

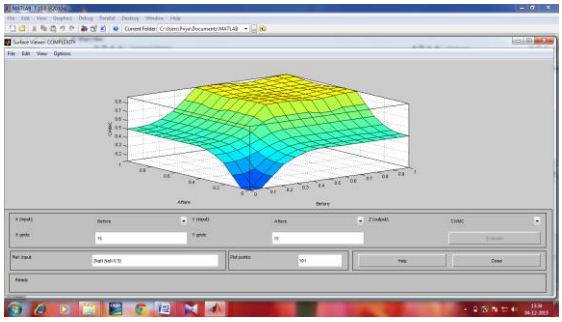

Figure 5.10 surface views with inputs as Before & After and output as CWMC

Figure 5.10 depicts a three-dimensional representation of the surface view that maps from the Before and After metrics on the input axes (X and Y) to the CWMC on the output axis (Z). More 3D figures, similar to Figure 5.11, can be produced by considering another set of inputs and outputs (CWMC).

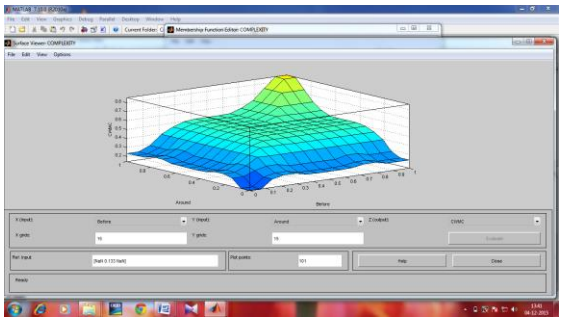

Figure 5.11 surface views with inputs as Before & Around and output as CWMC

VI Conclusion & Future Work

In this article, a rule-primarily based totally fuzzy version for predicting the CWMC complexity degree of aspect-oriented software program is given. First, we diagnosed the variables influencing software program maintainability for aspect-oriented software program, after which we advanced a connection among the diagnosed elements and maintainability. This connection is used to forecast external quality factors such as maintainability.

Based on the input factors, this relationship is used to forecast the external quality aspect, such as maintainability. In a fuzzy inference system, the process of measuring CWMC is automated by employing three essential factors: before, after, and around. This model predicts the CWMC metric value automatically for a module as well as for the entire software, which aids in the assessment that based on the provided set of design criteria for AO software, determine the amount of software maintenance. It would encourage software companies to create software with fewer aspects.

VII References

[1] A.Aloysius and G.Arockia Sahaya Sheela, “Aspect Oriented Programming Metrics – A Survey”, in the “International Journal of Emerging Trends in Computing and Communication Technology (IJETCCT)”, Volume 1, No 3, August 2015.

[2] G.Arockia Sahaya Sheela and A.Aloysius, "Analysis of Measuring the complexity of Advice using a Cognitive Approach", "International Journal of Applied Engineering Research (IJAER)", Vol. 10, No.82, 2015.

[3] Ceccato M, and Tonella P,"Measuring the Effects of Software Aspectization", WARE, 2004.

[4] KotrappaSirbi and Prakash Jayanth Kulkarni,"Metrics for Aspect Oriented Programming-An Empirical Study", IJCA, 2010, pp 17-23.

[5] Chidamber S.R., Kemerer, C.F., "A metrics suit for object oriented design", IEEE, Trans. Software Engineering, 1994, vol 20, pp 476-498.

[6] Parthipan, SenthilVelan and Chitra Babu, "Design Level Metrics to Measure the Complexity Across Versions of AO Software", IEEE, 2014.

[7] K.R.Martin, Dr.E.Kirubakaran and Dr.E.George Dharma Prakash Raj (2017). Extended Metric of cognitive weighted coupling on Method Call, International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), vol.2, Issue 5, September.

[8] Aloysius, "A cognitive complexity metrics suite for Object-oriented design" Thesis, 2012

[9] G. Arockia Sahaya Sheela and Dr. A. Aloysius (2018). Aspect Oriented Programming - Cognitive Complexity Metric Analysis Tool, International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN No. 2456-3307, vol.3, Issue 1. (UGC Approved Journal)

[10] Amalarethinam, DI George, and PH Maitheen Shahul Hameed (2016). Validation of Object Oriented Metrics Using Open Source Software System: Fuzzy Logic Technique, ISSN: 0974-5572, I J C T A, 9(27), 2016, pp. 165-173

[11] G. Arockia Sahaya Sheela and Dr. A. Aloysius (2017). Maintenance Effort Prediction Model using Aspect Oriented Cognitive Complexity Metrics, International Journal of Advanced Research in Computer Science (IJARCS), ISSN No. 0976-5697, vol. 8, Issue 9, pp. 1-4, September. (UGC Approved Journal)

[12] N.Fenton, S.P.fleeger, "Software Metrics: A Rigorous and Practical Approach". PWS Publishing Company, 1997.

[13] Sivanandam, S.N., Sumathi, S., Deepa, S.N., ―Introduction to Fuzzy Logic using MATLAB‖, Springer, Heidel-berg, 2007.

[14] G.Arockia Sahaya Sheela, (2017). Design and Analysis of Aspect Oriented Metric CWCOAR using Cognitive Approach, IEEE, ISBN: 978-1-5090-5573-9, pp. 195-197.

[15] G.Arockia Sahaya Sheela, (2016). Statistical Analysis of Pointcut Complexity Metric using Cognitive Approach, International Journal of Control Theory and Applications (IJCTA), ISSN: 0974-5572, 9(27), 2016, pp. 29-34.

[16] G.Arockia Sahaya Sheela and A.Aloysius (2015). Analysis of Measuring the complexity of Advice using a Cognitive Approach, International Journal of Applied Engineering Research (IJAER), Vol. 10, No.82.

[17] G.Arockia Sahaya Sheela, and A.Aloysius (2016). Design and Analysis of Aspect Oriented Metric CWCAE using Cognitive Approach, International Journal of

Engineering Research & Technology (IJERT), ISSN: 2278 – 0181, Vol. 5 Issue 04, April. (Scopus Indexed)

[18] P.K. Singh, O. P. Sangwan, A. Pratap, A. P. Singh, An Analysis on Software Testability and Security in Context of Object and Aspect Oriented Software Development, International Journal of Security and Cybercrime, Romania, Vol. 3, Issue 1, pp. 17-28, 2014.

[19] Mac Donell S.G., Gray A.R., Calvert J.M., ―Fuzzy Logic for Software Metric Practitioners and Researchers, Published in Proceedings of the 6th International Conference on Neural Information Processing ICONIP, Perth, pp. 308-313, 1999.

[20] Ryder J., ―Fuzzy Modeling of Software Effort Prediction‖, Published in Proceedings of IEEE Information Technology Conference, pp. 53-56, Syracuse, New York, 1998.

[21] http://www.mathworks.in/help/fuzzy/ building-systems-with-fuzzy-logic-toolbox-software.html (last accessed on 12/02/14).

Dr. G. Arockia Sahaya Sheela is working as Assistant Professor in Department of Computer Science, Holy Cross College (Autonomous), Tiruchirappalli, Tamil Nadu, India. She has obtained the Master of Computer Science degree in 2005 and Master of Philosophy degree in 2007 from Bharathidasan University and Doctor of Philosophy in Computer Science degree in 2019, from Bharathidasan University, Trichy. She has 16 years of experience in teaching Computer Science and Research Experience. Her research areas are Sensor Networks, Cloud Metrics and Software Metrics. She has published many research articles in the National/ International Conferences, and Journals.

Dr. K R Martin is working as Assistant Professor in Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has obtained the Master of Computer Applications degree in 1999, Master of Philosophy degree in 2007 from Bharathidasan University and Doctor of Philosophy in Computer Science degree in 2021 from Bharathidasan University, Trichy. He has 24 years of experience in teaching Computer Science and Research Experience. His research areas are Software Engineering, Cloud Metrics and Software Metrics. He has published many research articles in the National/International Conferences and Journals.