Study of evaluation models for aspect-oriented software maintainability according to Fuzzy-based methods

Zeynab Samdaliri¹, Fatemeh Yousefian², Reza Tavoli³

*Zeynab Samdaliri, Fatemeh Yousefian, Reza Tavoli

¹Dept. of Computer Engineering, Shafagh Institute of Higher Education, Tonekabon, Iran
²Dept. of Computer Engineering, Islamic Azad University, Central Tehran Branch, Tehran, Iran
³Dept. Math, Islamic Azad University, Chalous, Iran

Abstract

The software maintenance is important and critical features in software evaluation; therefore maintainability measurement is considerable issues for the capabilities of software maintenance in aspect-oriented programming. In this paper, various fuzzy methods and fuzzy-based model is used to calculate maintainability. According to the studied paper, in this study, fuzzy XNOR model is the best method to calculate fuzzy maintainability base on available input parameters. Although fuzzy-based model will be described which input parameters of maintainability produce in its output by receiving values. This model could be applicable and important for software developers and consumers.

Key words: maintainability, fuzzy, Aspect oriented, maintenance.

1. INTRODUCTION

Aspect-oriented programming is a programming paradigm, which is used to increase the modularity degree in a system. Oriented aspects will help developers to maintain and manage system easier [1].

Aspect-oriented programming language first was introduced by Gregor Kiczales in 1991 [2]. Production of aspect-oriented software was developed due to the importance and value of Modularization. The separation of concerns in modules forms is the principal tasks of this programming language so that they have minimal overlapping. Aspects–oriented Programming provides the design and construction ability of high quality system and optimal maintenance capabilities to users. Maintenance software is a set of activities that are given to system software with the aim of making changes to respond to some reasons. These reasons may be anything like: resolving error, augment efficacy or acclimatize to a changed environment in terms of hardware or operating system [3].
One of the key factors in the assessment of software quality is the maintenance and the software development costs. Main software development costs are devoted to its Maintenance [3-6]. Software maintenance is one of the most expensive activities that consume about 50–70 percent of the development cost [6]. Hence, attention and further studies are needed to identify factors affecting the maintenance of aspect-oriented software. The goal of this paper is comparison of models and factors that there are for maintenance. Also, identification of the best model to find the most important factors in the maintenance among the proposed fuzzy methods is which one have recently been proposed. In the following sections, I will a brief describe briefly about the aspect-oriented, its important parameters and full description of its maintenance and affected factors. Finally, conclusions will be presented in Section 5.

2. BASIC CONCEPTS

2.1. Aspect Oriented

Aspect-oriented is a technology for software development in which are created common concerns in the system to prevent a repetition in the modules form, these modules are implemented in the system automatically. In software development by aspect-oriented programming, the core logic of program is isolated from other sectors using Modularization and created individually as aspects.

Aspect Oriented Software Development (AOSD) is a promising paradigm which allows concerns separated from each other. Concern as part of a problem that can be solved as a conceptual unit. These concerns called the cross-cutting concerns [4]. According to the done research, it can be said, aspect-oriented systems are complement of object-oriented systems [7, 4, 5, and 3].

In Aspect-oriented systems, change of static OOP model and new code creation leads to satisfy the needs and easy use for developers. So this conclusion is obtained that aspect-oriented programming is complementary of object-oriented programming and should be considered to replace object-oriented [4]. The available standards in object-oriented programming can be used for aspect-oriented systems [7]. One of most important feature of aspect-oriented programming is less complex to program, so it has more flexibility. Also understanding and development of the system is easy in it. On the other hand, the change in aspect-oriented system models is easier than object-oriented systems.

2.2. Maintainability

Software maintenance is as a main criterion for quality evaluation that Software must have this criterion according to the ISO standards [8]. The definition of maintainability as per IEEE is defined as: "The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment is maintainability. Maintainability is a system that can evolve from its current state to its future desired state. It is termed as the most difficult and costliest activity due to it is inherent involvement in making predictions about the future" [3, 9]. Maintenance model is a system that can be used to assess the maintenance of its software. This model ensures standards and
guidelines that a software system can be produced by following it. The system software needs to less maintenance activities. To create a maintenance model of the statistical approach or advanced machine learning methods are used [5].

2. 3. Affected factors in maintenance

There are many affected factors to predict maintenance for object-oriented and aspect-oriented systems. Some of these factors include the following: Coupling, Cohesion, Traceability, Class complexity, size, Modularity, Readability, Programming Language, Standardization, Level of Validation and Testing, Separation of concerns (SOC), Changeability, Stability, Testability and etc [3-8,10]. Here are some common factors in the research are explained.

• Coupling
This feature represents the association degree between components and or modules and in fact, this shows dependencies between features and some classes in the system. According to the principles of software engineering, less coupling is optimal for the quality of the systems design and maintenance [3, 4].

• Cohesion
This factor determines belong degree of the elements of a module and power of internal module. If the amount of Cohesion is more, the system quality will be higher [3,5].

• Size
This feature is related to the program duration and static source code [3.11]. Measurement criterion is size LOC (line of code) [11,12].

• Class complexity
The complexity is introduced as one of the important and affected factors in several aspects of maintainability [13]. This feature shows the complexity of certain operations in the modules [5].

• Separation of Concern
This feature is the definition of introduced complexities by aspect-oriented programming [5]. Foundation of aspect-oriented programming is separation of concerns [4]. SoC is ability to encapsulate, manipulate and identify the part of the software that is related to a specific concern [3.11].

3. DONE RESEARCH

Ananthi and Roby presented a fuzzy XNOR system in order to evaluate the maintainability-based in aspect-oriented system in 2016. They assessed target model using fuzzy logic based Mamdani engine inference and using effective four-factor model in maintenance models.
Four factors of maintainability are: Flexibility, Understandability, Complexity and Modularity that are entered the login fuzzy as input, the system produces the optimal amount for maintenance. According to these four parameters, input variables are three membership functions ranging from low, medium and high. Input ranges are shown in Table 1.

### Table 1. Fuzzy membership functions of input variables

<table>
<thead>
<tr>
<th>Membership function</th>
<th>Input variables</th>
<th>Output variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Range</td>
<td>Range</td>
</tr>
<tr>
<td>Low</td>
<td>0   0.185 0.37</td>
<td>0  0.212 0.422</td>
</tr>
<tr>
<td>Medium</td>
<td>0.31 0.495 0.68</td>
<td>0.31 0.495 0.68</td>
</tr>
<tr>
<td>High</td>
<td>0.63 0.815 1</td>
<td>0.63 0.815 1</td>
</tr>
</tbody>
</table>

Ananthi and Roby have used values of factor such as Coupling, Cohesion, size, class complexity and SOC for input parameters that CDA (Crosscutting Degree of an Aspect) criterion in order to SOC measurement, CFA (Coupling on Field Access) criterion and CAE(Coupling on Advice Execution) criterion to evaluate coupling, from WOM (Weighted Operations in a Module) criterion and RFM (Response for) criterion in order to class complexity, LCOO (Lack of Cohesion in Operations) criterion to evaluate cohesion and LOCC is utilized to measure the size. Mapping of Maintainability metrics is shown in Figure 1.

![Figure 1. Mapping of metrics to maintainability [5]](image)

Output for Fuzzy Inference System (FIS) is maintenance of aspect-oriented software that is classified in three ranges of low, medium and high. Outputs range shown in Table 1. In [5], three synthetic model are used included fuzzy XNOR-OR combination, AND-OR combination, fuzzy Prod-sum combination. Based on the results presented in this paper, fuzzy XNOR-OR combination has the best performance based on the amount of maintenance which can be seen in Table 2.
TABLE 2. Input-Output values for fuzzy XNOR-OR combination [5]

<table>
<thead>
<tr>
<th>understandability</th>
<th>Flexibility</th>
<th>Modularity</th>
<th>Complexity</th>
<th>Input Name</th>
<th>Output Value for Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.826</td>
<td>0.826</td>
<td>0.846</td>
<td>0.208</td>
<td>Ip1</td>
<td>0.219</td>
</tr>
<tr>
<td>0.503</td>
<td>0.503</td>
<td>0.503</td>
<td>0.254</td>
<td>Ip2</td>
<td>0.589</td>
</tr>
<tr>
<td>0.128</td>
<td>0.3864</td>
<td>0.159</td>
<td>0.8538</td>
<td>Ip3</td>
<td>0.731</td>
</tr>
</tbody>
</table>

The values presented in Table 2, the result is obtained, when Understandability, Flexibility and Modularity are high and complexity is low, minimum value for Maintenance is obtained which is indicative of high quality of software. Momeni and Zahedian evaluated aspect-oriented software maintenance using ANFIS. In this paper, values of 4 parameter CAE, CDA, WOM and CFA for input variables are used to assess Maintainability. input range shown in Table 3.

TABLE 3. fuzzy membership functions of input variables [6]

<table>
<thead>
<tr>
<th>Range of Membership function</th>
<th>Membership Function</th>
<th>Range</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0 - 0.27</td>
<td>Low</td>
<td>0-1</td>
<td>CAE</td>
</tr>
<tr>
<td>0.25 - 0.52</td>
<td>Medium</td>
<td>CDA</td>
<td></td>
</tr>
<tr>
<td>0.50 - 0.77</td>
<td>High</td>
<td>CFA</td>
<td></td>
</tr>
<tr>
<td>0.75 - 1.00</td>
<td>Very High</td>
<td>WOM</td>
<td></td>
</tr>
</tbody>
</table>

Output is classified in five classes( low, very low, medium, high and very high) which outputs range can be seen in Table 4.

TABLE 4. The range of linguistic variables of output [6]

<table>
<thead>
<tr>
<th>Function</th>
<th>Range</th>
</tr>
</thead>
</table>

2604
In this study, Gaussian membership function is used for the output variable and the law 256 is used for inference engine. In Momeni and Zahedian studies, fuzzy logic approach and compatible neuro-fuzzy inference system (ANFIS) been proposed to evaluate maintainability of aspect-oriented systems. Then, they compared the RMSE values obtained by the two approaches together. In the classification results by fuzzy system based on Mamdani inference engine, obtained RMSE value is 0.180 which shows the classification accuracy is not desirable in this manner.

The used second method is a neuro-fuzzy system based on Sugeno inference engine. In this method, the output nodes of the first layer is the degree of linguistic variables that membership functions bell was used in this layer.

The second layer in the fuzzy neural network is law layer which the part of law condition is calculated by the operator of the fuzzy min. And the result is obtained as the degree of implementation of laws. In this layer, learning activities is the gain of the best grade based on implementing laws regard to the training data that is injected into the network.

In the third layer of linear combination, the result of the law rate is used to determine the degree of membership in a particular category. In the fourth layer, membership Sigmund function is used. Membership functions are linear for input, triangle and output variables. The number of available samples in the training data set is 126 and the number of training courses is 20.

Structure of Artificial fuzzy neural network is shown in Figure 2.

![Figure 2. Structure of fuzzy artificial neural network](image)

The results of the research by Zahedian and Momeni show that the RMSE value was equal to 0.780 and fuzzy neural network was equal to 0.21 in the fuzzy method. These results indicate that the fuzzy method just is weaker than Adaptive Fuzzy Neural Network to categorize and
estimate maintainability of aspect-oriented software and also forecast error is higher. Maintenance software refers to changes in which are created after Initial Release, setup and operation. Several studies also have shown that maintenance has more than 40% of the total cost of the software. In a paper presented by Sineq et.al, major affected factors divided aspect-oriented maintenance software into four categories Separation of Concern, Cohesion, Coupling and size. They have proposed a fuzzy model to predict software maintenance based on these four metrics (as input). Presented phase model is shown in Figure 3.

![Figure 3. Fuzzy model for aspect oriented software maintainability](image)

In proposed fuzzy model by Sineq, fuzzy inference Mamdani method is used. Linguistic variables also are for all inputs (low, medium, high) and linguistic variables for output (very low, low, medium, high, very high). Total inputs is of this model are considered in spectrum [1 0] and all values set of possible input are obtained with 81 law for the model, some of them are shown in table 5. In here, low amount of membership degree to maintain is optimal for better quality of software and requires less maintenance efforts.

<table>
<thead>
<tr>
<th>Rule</th>
<th>SO</th>
<th>Cohesion</th>
<th>Coupling</th>
<th>Size</th>
<th>Membership Level</th>
<th>Membership grade of software maintainability</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Very High</td>
<td>[0.1742;0.1769;0.8258;0.7923]=[0.879]</td>
</tr>
<tr>
<td>77</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
<td>[0.9542;0.9542;0.3458;0.5]=[0.225]</td>
</tr>
</tbody>
</table>

In sum, according to the results in Sineqan and et.al paper can said that the proposed model is prediction of automated maintainability software for a module and also for the whole software, which is provided using the values of the input parameters. In fact, it helps to assess the level of maintenance of the software to design AO. In [8], fuzzy logic as an approach for Software Maintainability Assessment (SMA) systems Aspect-Oriented has been used in which an algorithm based on fuzzy logic are presented for SMA aspect-oriented systems. Maintenance software is determined by both qualitative and quantitative data. Effective metrics used for determining maintainability include: coupling, size and cohesion and complexity. This criterion
in fuzzy system has three levels, low, medium and high which based on the existing rules in the knowledge base; output values are included low, medium, high and very high. Proposed algorithm is provided in Antani and Ruby article on pseudo-code below.

```plaintext
Pseudocode to AOFMM()
// input: complexity, coupling, size and cohesion values
// output: Maintainability value based on metrics.
begin
1. /* create the fuzzy rules using if-then statements */
2. /* Calculate membership value for premise parameter */
   For i = 1 to n
   Output(i) = \mu_a(x)
   where
   \mu_a(x) = \exp\left(-\frac{(c-x)^2}{2q^2}\right)
3. /* Enthusue and normalize strength of rule */
   For i = 1 to n
   Assign \sum w = 1
   F'(A_1, A_n) = \sum w B_i
   B_i to B_n is decreasingly ordered nature of fuzzy numbers
   A_1 to A_n so that B_i is the i-th largest element.
4. /* find consequence of the rule through the combination of outputs in step 2 and step 3 */
   Calculate (a_1 x_1 + a_2 x_2 + \ldots + a_n x_n + t)
   where a_1, a_2, \ldots, a_n are consequent parameters
5. /* Combine the output to an output distribution */
   Use fuzzy OR to combine the outputs of all fuzzy rules to obtain a single fuzzy output distribution
6. /* Defuzzify the output */
   Use mean of maximum method to find a sharp value for Output
\[
   m = \sum_{j=1}^{n} \frac{z_j}{m}
\]
end
```

This approach tries to build a maintainability model for aspect-oriented system. The output of above pseudo-code implementation; quality offers maintainability in aspect-oriented systems.

4. EVALUATION AND COMPARISON

In conducted studies by Ananthi Sheshasayee and Ms. Roby Jose in 2016, a fuzzy XNOR system is presented to evaluate criteria based-maintainability in aspect-oriented system. Input parameters of the model are Flexibility Understandability, Complexity and Modularity, the system produces the optimal value for maintenance based on these four input values. The results
of this study show that when understandability, flexibility and modularity are high and low complexity, the optimal amount is assessed for maintenance. In a study by Hossein Momeni and Shiva Zahedian in 2014, a neuro-fuzzy inference system is proposed to assess the maintainability of aspect-oriented systems.

The input parameters of model are CAE, CDA, WOM, CFA which the results from this study, the obtained root-mean-square error (RMSE) is decreased from the fuzzy neural network rather than fuzzy approach. In research conducted by Pradeep Kumar Singh and et.al in 2014, a fuzzy model was developed to predict the maintenance software in aspect-oriented systems that input parameters of this model include: Separation of Concern, Cohesion, Coupling and size. Their proposed model is automated prediction of the maintainability software for module as well as for all applications. Ananthi Sheshasaayee and Roby Jose in 2076 developed fuzzy logic-based algorithms to evaluate Maintainability software in aspect-oriented systems. They use the input parameter, coupling, size, cohesion and complexity for model maintainability. The resulted output to run the proposed algorithm is maintainability quality in the oriented-aspects systems. Table 6 shows the results of each method with input and output parameters. Fuzzy approach has great potential to show ambiguous data, which is used in phenomena with uncertainty; so in this article, several models have been investigated which have studied the different factors to maintain aspect-oriented systems based on fuzzy approach.

Because the aspect-oriented paradigm is developing, there is no complete set of criteria for evaluating maintainability of an aspect-oriented system. Finding important metrics of maintainability has an important role in assessing the quality of maintainability.

**TABLE 6. Results of each method with input and output parameters**

<table>
<thead>
<tr>
<th>Authors</th>
<th>Method</th>
<th>Input</th>
<th>Output</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ananthi Sheshasaayee and Ms. Roby Jose</td>
<td>fuzzy XNOR-OR combination</td>
<td>Understandability</td>
<td>Flexibility</td>
<td>With the increasing understandability, modularity, flexibility, reduced complexity. Achieving the optimal value for maintenance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Complexity</td>
<td>ModularitY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Complexity</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hossein Momeni and Shiva Zahedian</td>
<td>Adaptive Neuro Fuzzy Inference System (ANFIS)</td>
<td>CAE</td>
<td>CDA</td>
<td>obtained RMSE value is 0.270 that decreased compared to fuzzy 0.510</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WOM</td>
<td>CFA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pradeep Kumar Singh and et.al</td>
<td>A new model using fuzzy inference system</td>
<td>SOC</td>
<td>Cohesion</td>
<td>Create a model to predict automatically the maintainability of aspect-oriented software using input parameter values</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Coupling</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Size</td>
<td></td>
</tr>
</tbody>
</table>

2608
A Fuzzy Approach for the Maintainability Assessment of AOS

Coupling Size

Maintainability quality in Aspect-Oriented system

Build a maintainability model for aspect-oriented systems using fuzzy logic-based algorithm

Thus, one of the main issues to evaluate maintainability of aspect-Oriented systems is, evaluating criteria that in reviewed studies by Ananthi Sheshasaayee; Ms. Roby Jose using fuzzy XNOR can be achieved. After extracting the important and effected factors in evaluating the maintainability, the estimated value of aspect-oriented software is very important. According to results of done research by Momeni and Zahedian, it can be said that fuzzy neural network operates more successful than categorization of estimation of maintainability and offers favorable results. They didn’t consider a comprehensive set of criteria that show proposed approach with limited criteria to evaluate maintenance of aspect-oriented system. Designing a model can play a significant role to calculate the maintainability for developers and consumers. Therefore scholars like Sineq and et.al were able to design a model based on the fuzzy inference engine that calculates the amount of software maintainability using the values of parameters, Separation of Concern, Cohesion, Coupling and size.

5. REFERENCES


