CARVING AND REPLAYING FUZZY INFERENCE SYSTEM FOR PREDICTING MAINTAINABILITY OF ASPECT ORIENTED SYSTEM

Manoj Kumar , Department of Computer Application, Galgotias University, Greater Noida, India; m_pachariya1@yahoo.com;

Abstract

Software quality is an important aspect for every software manufacturer. In Aspect Oriented Programming (AOP) refers to the programming approach which isolates secondary and supporting functions from the main program's business logic. The application's modularity increased in that way and its maintenance becomes significantly easier. But there are some other non-functional factors which affect the software maintainability. In this paper, Fuzzy Inference System based Framework is proposed to predict the maintainability of Aspect Oriented Software System. The Proposed approach is empirically evaluated to classify the Aspect Oriented Software System in five classes on the basis of efforts required to maintain the AOP software system.

Introduction

Aspect-oriented software development (AOSD) is a new technique to support separation of concerns in software development [1, 2]. In recent years, AOS comes out as a better platform for building a software, this is all because the limitations which comes in object-oriented programming. It defines a new program construct, called an aspect, which is used to capture crosscutting aspects of a software application in separate modules. In Aspect Oriented Programming paradigm with breaking down a progress into cohesive area of functionality, which is called concerns, the cut across multiple in order to increase modularity. It supports for abstraction to group and encapsulate concerns in to unique entity. There are top 8 core concerns in which AOP is better solution than the Object-oriented programming. These are the top concerns which is overcome in Aspect Oriented Programming. But AOP known for crosscutting concerns, so here we discuss what is crosscutting concerns. The Details of these common concerns are given in Table 1.

Today's most of the company work on maintaining the software and AOS plays an important role while maintaining a software which is made on Aspect Oriented Programming. Another factors are Reusability and Reliability, as maintainability is crucial for software quality, reusability is also important for software quality, reusability allows us to reuse the code, or in any type of fault in software, reusability attribute helps us to recovering the code which is quite useful to reuse the code or troubleshooting when the fault is occur in the code and saves the time. Using minimum resources and gain maximum output from the software, and due to this software is decreased day by day , so main objective is quantified the software quality by using fuzzy logic approach with appropriate modification in it. We used fuzzy logic approach to measure the reliability, reusability and maintainability of Aspect Oriented System. Since this measurement is on single entity of software quality, so in last part of the paper, by the use of AHP(Analytical Hierarchy Process) to made pairwise comparison matrix of 3 software quality attribute and then find the single value of software quality by using some assumption (in generalized way).

Table1 Commonly Used Cross Cutting Concerns

S. No	Commonly Used Cross Cutting Concerns
1	Database
2	Data entitles
3	Worker
4	Result Processing
5	Process Flow Manager
6	Email/Notifications
7	Error Handling
6	Logging

Fuzzy Inference System

Fuzzy model is the best choice for managing vague, imprecise, doubtful, contradicting and diverging opinions. In Fuzzy modeling, there are four modules (Rule Base, Fuzzification, Inference Engine, Defuzzification), which transform the crisp inputs into fuzzy values. Then these values are processed by inference engine in the fuzzy domain using the rule base created by domain expert. Finally the processed output is transformed from fuzzy domain to crisp domain by defuzzification module [4]. A fuzzy model performs its operations in the following steps [4]:

2.1 Fuzzification of Inputs

Fuzzification is the first step in the Fuzzy inferencing process. This involves a domain transformation where crisp inputs are transformed into fuzzy inputs.

2.2 Apply Fuzzy Operators

Fuzzy operators are applied to compute the degree of support for each rule for the generation of a single crisp value. This value will then be applied to the output functions. The input to a fuzzy operator is two or more membership values from fuzzified input variables. The output is the single truth-value. Two commonly used fuzzy operators are MIN -MAX operators.

2.2.1 Apply implication methods.

The shaping of the consequent based on the antecedent is termed as implications. The input for the implications processes is a single number given by the antecedent, and the output is a fuzzy set. Implication is applied for each rule. Two commonly used implication methods for AND are MIN (i.e. truncation) and

PRODUCT (i.e. scaling the height of a fuzzy set). Two commonly used implication methods for OR are MAX (maximum) and probabilistic OR (algebraic sum).

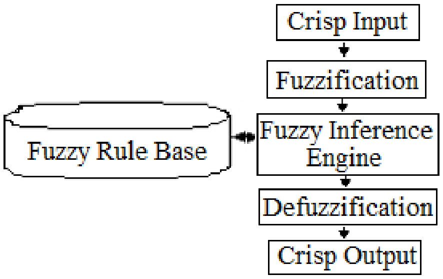

Figure 1 Block Diagram of Fuzzy Model

2.3Aggregate all outputs

Aggregation is a process by which fuzzy sets are combined in desirable way to produce a single fuzzy set. It is obtained by combining all the fuzzy sets that represents the output of each rule into a single fuzzy set. Aggregation occurs only once for each output variable. For our model the input of the aggregation processes is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable.

2.4 Defuzzification

Defuzzification transforms the fuzzy values to crisp values. The input of the defuzzification process is a fuzzy set and the output is a single fuzzy number. Given a fuzzy set that encompasses a range of output values; one number needs to be returned thereby moving from a fuzzy set to crisp.

Table2 Input Parameters and Expected Output Value

S. No	Input Parameters	Expected Output Value
1	Analyzability	Low, Medium, High
2	Changeability	Low, Medium, High
3	Stability	Low, Medium, High
4	Testability	Very Low, Low, Medium, High, Very High
5	Adaptability	Low, Average, High

Proposed Model

The first step in fuzzy inference step is to fuzzify inputs. The input parameters to fuzzy inference system are fuzzy, vague and imprecise value. In proposed fuzzy inference system, five characteristics are used as input parameters to predict the maintainability of Aspect Oriented Software System. The defuzzification process of proposed approach provide the predicted value of maintainability of Aspect Oriented Software System. The input parameters to proposed Fuzzy Inference System are described in Table 2. Then these fuzzy sets are represented by a membership function. This can be best achieved by deciding numerical range for the fuzzy input/output variables.

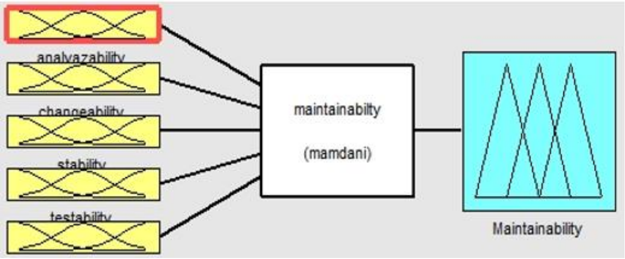

Figure 2 : Fuzzy Inference system for Maintainability

Table 3 Ranges of Fuzzy Input Parameters Used for Predicting Maintainability

S. No.	Input Parameters	Linguistic variable	Numerical Range
1	**Analyzability**	Low	0-3
		Medium	3-6
		High	6-10
2	**Changeability**	Low	0-3
		Medium	3-6
		High	6-10
3	**Stability**	Low	0-4
		Medium	3.5-7.5
		High	7-10
4	**Testability**	Very Low	0-3
		Low	2-4
		Medium	4-6
		High	6-8
		Very High	8-10
5	**Adaptability**	Low	0-4
		Medium	4-8
		High	7-10

Table 4 Range Values of Fuzzy Output Variable Maintainability.

Output Parameter	Linguistic Variable	Numerical Range
Maintainability	Very Low	0-2.5
	Low	2.5-4.5
	Medium	4-6.5
	High	6.2-8.2
	Very High	7.5-10

In proposed approach, five different key input parameters are used to predict the value of maintainability of Aspect Oriented Software System. The five different input parameters have different values and different number of classes. In Proposed Fuzzy Inference System, the ranges of input parameters Analyzability, Changeability, Stability, Testability and Adaptability are described in Table 3. In Proposed Fuzzy Inference System, similarly the output value of maintainability of Aspect Oriented

Software is described in Table 4. After the fuzzification of input parameters, several rules of fuzzy inference system are designed and applied to various input parameters to obtain output variable. In this study, Triangle Membership function is used for calculating the fuzzy value of input and output variables. In this study, 395 fuzzy inference rules are designed and applied to get optimum value of output variable maintainability.

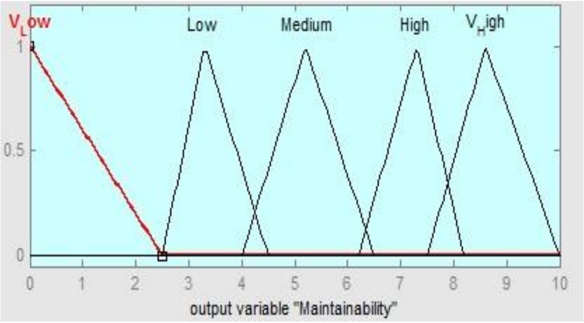

Figure 3 Fuzzification for Output Variable Maintainability

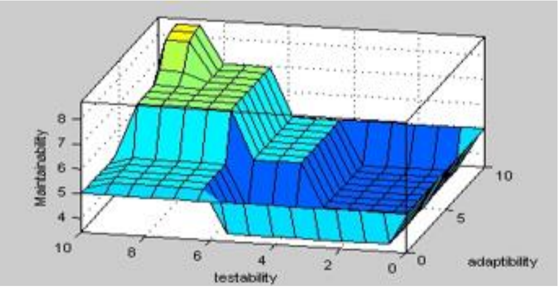

Figure 4 Surface view between Testability and Adaptability

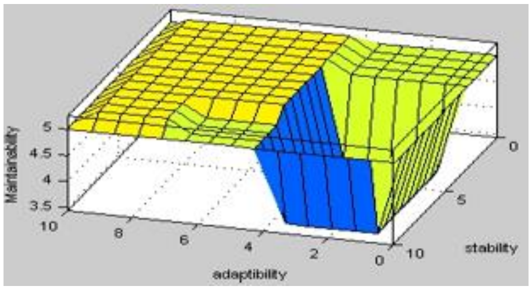

Figure 5 Surface view between

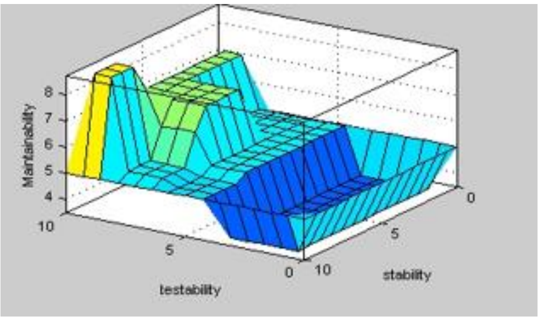

Figure 6 Surface view between Testability and Stability

The proposed study presents the persistency of maintainability in Aspect Oriented Software System while concurrent consideration of above-mentioned sub-characteristics of maintainability. After that the defuzzification process is carried to know the crisp value of output variable maintainability. In this study, Centre of Gravity method is used for defuzzification process [12].

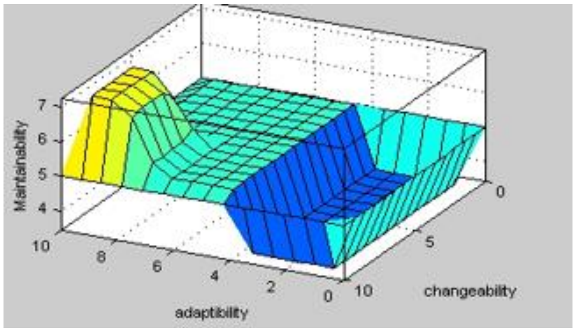

Figure 7 Surface view between Adaptability and Changeability

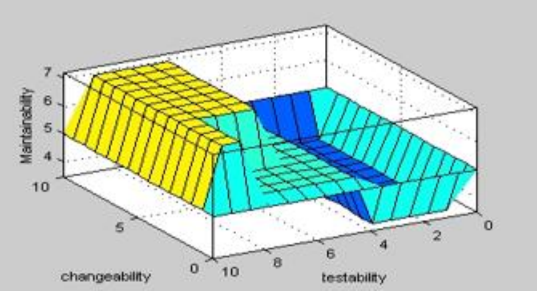

Figure 9 Surface view between Changeability and Testability

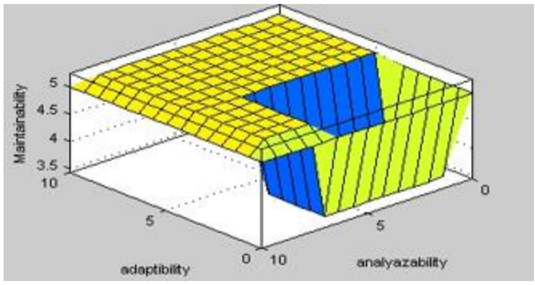

Figure 8 Surface view between Adaptability and Analyzability

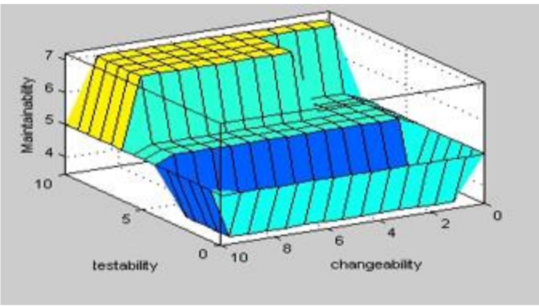

Figure 10 Surface view between Testability and Changeability

Result & Discussion

Suppose we have the following crisp values at input: Analyzability (4.091), Changeability (6.39), Stability (7.36), Testability (7.55), Adaptability (6.39) then output for this system is observed as 6.59 which is Medium as well as High which means that maintainability in AOS is above average and maintain ability is easy for AOS. we can describe as if Analyzability of system is either low or medium, Changeability is high, Stability is either medium or high, Testability is either high or very high and Adaptability is either average or high then we conclude that system is subject to maintain.

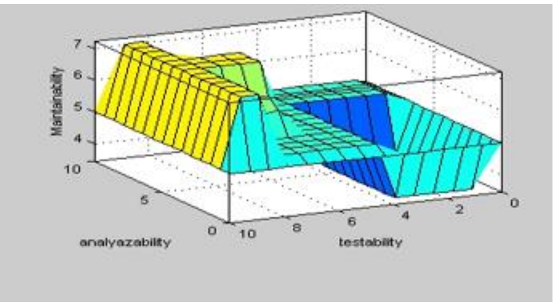

Figure 11 Surface view between Analyzability and Testability

Conclusion & Future Scope

In this paper, fuzzy logic approach is designed and implemented to predict the value of maintainability of Aspect Oriented Software System. This study also shows the effectiveness, importance and contribution of sub characteristics in calculating the maintainability of Aspect Oriented Software System but this study, Quality of Aspect Oriented Software System is not explored. In future, Fuzzy Inference System will be developed for predicting the single of quality of aspect-oriented software system. In Future, Fuzzy AHP approach may be explored for predicting the single of quality of aspect-oriented software system.

References

[1]. G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J. M. Loingtier, and J. Irwin, "Aspect-Oriented Programming, " *Proceedings of the 11th European Conference on Object-Oriented Programming*, pp.220-242, LNCS, Vol.1241, Springer-Verlag, June 1997.

[2]. G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J. M. Loingtier, and J. Irwin, "An Overview of AspectJ," *Proc. 13th European Conference on Object-Oriented Programming*, pp.220-242, LNCS, Vol.1241, Springer-Verlag, June 2000.

[3]. Hema Kashyap "Process to identify the crosscutting concerns in changing requirements through Aspect-Oriented Software Engineering", IJCA, April 2014

[4]. Ravinder Kumar, Kiran Khatter, Arvind Kalia " Measuring Software Reliability : A Fuzzy Model ",ACM SIGSOFT Software Engineering Notes , November 2011

[5]. Reena Dadhieh, Bhavesh Kumar, "Measuring Reliability of an aspect-oriented software using Fuzzy Logic Approach."

[6]. S. W. A. Rizvi, R. A. Khan," Maintainability Estimation Model for Object-Oriented Software in Design Phase "

[7]. J.H. Hayes and L Zhao, "Maintainability Prediction: a Regression Analysis of Measures of Evolving Systems," Proc. 21st IEEE International Conference on Software Maintenance, 26 - 29 Sept. 2005, pp. 601 -604, 2005.

[8]. Avdhesh Kumar, Rajesh Kumar, PS Grover," An Evaluation of Maintainability of Aspect-Oriented Systems: a Practical Approach"

[9]. Daesung Park, Sungwon Kang, Jihyun Lee," Design Phase Analysis of Software Qualities Using Aspect-Oriented Programming"

[10]. Avdhesh Kumar, Rajesh Kumar, PS Grover "A Fuzzy Logic Approach to Measure Complexity of Generic Aspect-Oriented Systems." IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990.

[11]. R. Laddad, "Aspect Oriented Programming will improve Quality", 2003,published by IEEE Computer Society 0740-745.

[12]. Pradeep Kumar Singh, Om Prakash Sangwan, Amar Pal Singh, Amrendra Pratap,"A Framework for Assessing the Software Reusability using Fuzzy Logic Approach for Aspect Oriented Software", IJITCS, vol.7, no.2, pp.12-20, 2015. DOI: 10.5815/ijitcs.2015.02.02

[13]. Alessio Ishizaka and Ashraf Labib, "Review of the main developments in the Analytic Hierarchy Process"

[14]. Mehmet Yüksel, "Evaluating the Effectiveness of the Chemistry Education by Using the Analytic Hierarchy Process"

[15]. D. Shukla, S. Fell, and C. Sells. (2002) Aspect-Oriented Programming Enables Better Code Encapsulation and Reuse. MSDN Magazine. Available: http://msdn. microsoft.com/enus/maGazine

[16]. O. Papapetrou and G. A. Papadopoulos, "Aspect Oriented Programming for a component-based real-life application: a case study," presented at the Proceedings of the2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004.