

A Study on Robert C.Martin's Metrics for Packet Categorization Using Fuzzy Logic

Gurpreet Kaur and Deepak Sharma

Assistant Professor (CSE Department), DBFGOI, Moga
HOD (CSE Department), DBFGOI, Moga
ap.gurpreet@gmail.com, deepdeep2003@gmail.com

Abstract

In Object-oriented paradigm, classes are the basic units to organize the small applications. The concept of 'package' introduced when programmer found it difficult to organize the large application using classes. A good organization of classes into identifiable and collaborating packages eases the understanding, maintenance, testing and evolution of the software. In order to assess and improve the quality of an application during development process, developers and designers use various software metrics. Robert C. Martin presented a set of software metrics to identify poorly designed packages on the basis of certain features like stability, abstraction and coupling of the packages. According to Robert C. Martin, the Packages which are highly interdependent tend to be rigid, not reusable and hard to maintain. These metrics have been used to predict the quality of the packages early in the design phase and hence helps the programmers to reduce the overall cost of development. But these Software metrics neither give the exact value nor defines the sharp boundaries to categorize the packages. In this research work, a decision making system based on Fuzzy inference mechanism is used to categorize the good and bad packages that depends on various factors which are vague in nature. The proposed model is finally validated using the Distance from the Main Sequence 'D' metric intended by Robert Martin.

Keywords: *Fuzzy Logic, Vagueness, Packages, Quality, R. C. Martin Metrics*

1. Introduction

Software engineering is a profession dedicated to analysis, designing, implementing and modifying software so that we can develop software of high Quality. Accurate measurement is a prerequisite for all engineering disciplines, and software engineering is not an exception. A larger body of software quality metrics has been developed and numerous tools exist to collect metrics from program representations. Software metric is the measurement of a particular characteristic of a program's performance or efficiency [2]. Software Metrics are essential in software engineering for measuring the software complexity, estimating size, quality and project efforts [1]. There are number of Software quality metrics like the HALSTEAD metrics, Size metrics, Object Oriented Metrics and Package metrics. Package metrics are used to predict the quality of the packages early in the design phase and hence helps the programmers to reduce the overall cost of development. The focus of package metrics is in identifying packages which are hard to maintain and reuse [8], on the basis of certain features like stability, abstraction and coupling of the packages. But the results given by these metrics are uncertain and not quantifiable. So, this vagueness can be removed by using Fuzzy Logic. Fuzzy Logic tool was introduced in 1965 by Lotfi Zadeh, and is a mathematical tool for dealing with uncertainty. The theory of Fuzzy Logic is based

upon the notion of relative graded membership and the functions of mentation and cognitive processes.

2. Characteristics of Packages

Packages can be divided into different categories based on their different features. There are no. of characteristics and features of packages, but according to R.C Martin some features of packages are stability of packages, abstractness of packages and coupling between packages. These features are chosen since they are design complexity features and effect packages of object oriented design system.

2.1. Stability

Robert C. Martin demonstrates that the stability of a package means how hard it is to change the package. The lesser flexible the package is, the more stable it is. A stable package is hard to change because a modification of it may lead to refactoring of other packages.

2.2. Abstractness

Abstraction of a package means how easy it is to add the new functionality to a package. According to Robert C. martin, if a package is stable, it should also be abstract so that it can be easily extended.

2.3. Coupling

Coupling measures the interdependence between the packages. *i.e.*, the number of classes outside the packages which depend on the package or the number of classes outside the packages on which the package depends upon.

3. Robert C. Martin's Metrics

We have used five software package metrics suite proposed by Robert C. Martin [5] for measuring the quality of object oriented design in terms of the interdependence between the packages. Following are the metrics-

3.1. Afferent Coupling (CA)

This metric measure the total number of classes out-side a package that depend upon classes within that package [5]. In Figure 1 shows a package `com.acme.ascp.exception`, with an afferent coupling of 4.

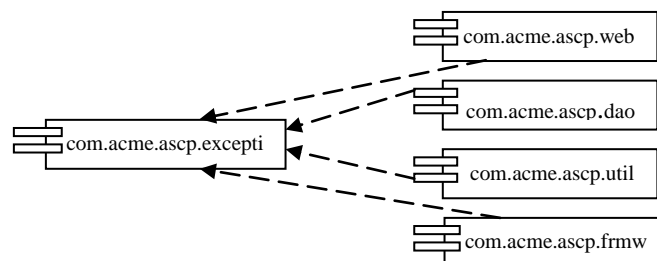


Figure 1. Afferent Coupling in the Exception Package

This metric is highly related with portability. Packages with higher CA are bad packages because they are harder to be replaced since they have a lot of other packages that depend upon them.

3.2. Efferent Coupling (CE)

Metric measures the total number of classes' in-side a package that depend upon classes outside this package [5]. For example, Figure 2 depicts the com.acme.ascp.dao package, which has an efferent coupling or $C_e = 3$.

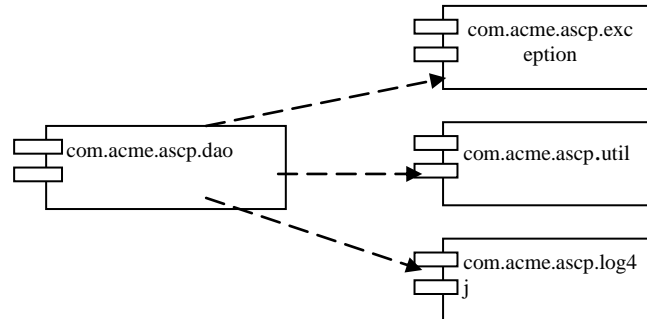


Figure 2. Efferent Coupling in the Dao Package

Packages with high CE are bad packages because they will negatively influence package reusability and maintainability, since it's harder to understand and isolate all the components necessary to reuse and maintain the package.

3.3. Instability (I)

This metric measures the instability of packages, where stability is measured by calculating the effort to change a package without impacting other packages within the application [5, 7]. Metric measures the ratio between CE and CE+CA.

$$\text{Instability } I = C_e / (C_a + C_e)$$

This metric produces results in the range [0, 1]. A value $I=0$ indicates a maximally stable package that depends upon nothing and $I=1$ indicates a total instable package that has no incoming dependencies but depends upon other packages. So instability negatively influences re-usability, maintainability and portability.

3.4. Abstractness (A)

This linear unit metric measures how abstract a package is, in that it will calculate the ratio between the number of abstract classes or interfaces and the total number of classes inside a package [5, 7]. One can measure the abstractness of a package (A) using the following formula:

$$A = N_a / N_c$$

Where:

- N_a is the number of abstract classes and interfaces in the package
- N_c is the number of concrete classes and interfaces in the package

This results in a value in the range [0, 1]. $A=0$ means that all classes in this package are non-abstract while $A=1$ shows a package that only consists of abstract classes and interfaces. More abstract packages are the good packages since they are easy to maintain and reuse.

3.5. Distance from Main Sequence (D)

Martin claims that every stable package should be abstract because stable packages are hard to change but they should be abstract so that they can be extended [5]. On the other hand instable package should be concrete so that their code can be easily changed. With

these principals, Martin [7, 8] defines the relationship between A and I presented in Figure 3.

To categorize the Packages Martin [6] introduced the fifth metric distance from main sequence (D) metric. The normalized distance D has a value in the range [0, 1] and is calculated by the formula

$$D = |(A + I) - 1|$$

Where A= abstraction, I= instability.

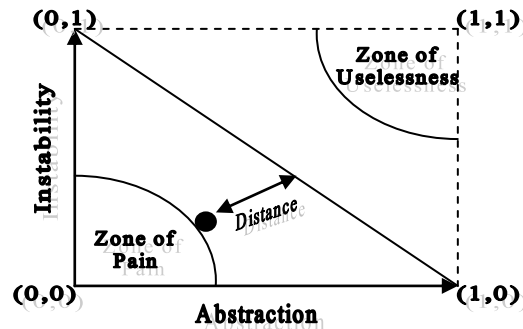


Figure 3. The Distance from the Main Sequence

Based on the value of D there are three cases.

Case 1: Desirable and Balanced Packages (D close to 0): Martin says that stable and abstract packages should be at the upper left corner at (0, 1). The instable and concrete packages should be at the lower right corner (1, 0). These are desirable packages [5]. Martin claims that practically only half of the packages are desirable. Packages have degrees of stability and abstraction. The line joining the points (1, 0) to (0, 1) represents the packages that have the right proportions of abstraction and stability. This line is called the main sequence. According to Martin the Packages that lie on or close to the main sequence is not “too abstract” for its stability, nor is “too instable” for its abstractness. Such packages are “Balanced Packages” as well as “good packages” having the value of D close to 0.

Case 2: Undesirable and Very Bad Packages (D close to 1): Martin argues that the coordinates (0, 0) represents the highly stable and concrete packages that are difficult to be extended and hard to change (Zone of Pain). The coordinate (1, 1) represents the packages that are highly abstract but don't have clients. These kinds of packages are useless (Zone of Uselessness) [5, 8]. And the packages far away from the main sequence *i.e.*, D close to 1 are very bad packages.

Case 3: Packages whose Value of D Neither Close to 0 nor Close to 1: The packages whose value of D is neither close to 0 nor close to 1 are not well defined by Robert. C.Martin [5, 7, 8]. Some of these Packages can have properties of balanced packages or bad packages.

4. Fuzzy Approach to Categorize the Packages

4.1. Proposed Model

According to Robert Martin the Packages that are close to or on the main sequence are considered as balanced packages (good packages). The Packages having D=0 lie on the main sequence but the value of D for the packages that are close to the main sequence is

not defined by Robert Martin, so here the uncertainty arises that what would be the range of value D metric for balanced packages. Thus we proposed a fuzzy model to remove this uncertainty.

Fuzzy logic is a captivating field of research these days as it considers the fuzzy value instead of binary values [1-3, 7]. The benefit of using fuzzy logic is that the fuzzy logic models can be built even with little or no data. In this paper, we proposed fuzzy model to categorize the packages that depend upon various factors which are fuzzy (vague) in nature.

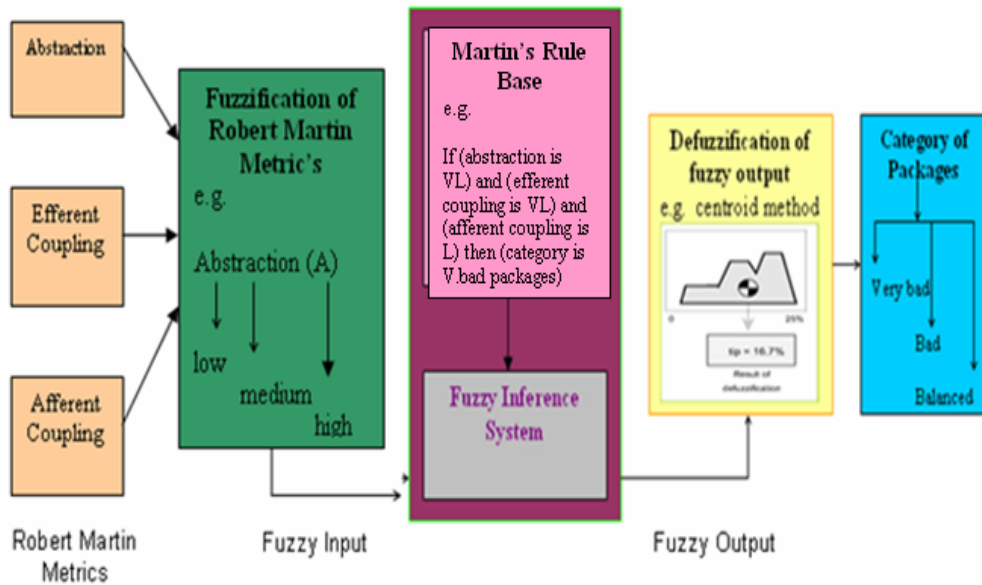


Figure 4. Block Diagram of FUZZY MODEL

4.2. Working of the Model

In the proposed model the crisp values of metrics abstraction, efferent coupling and afferent coupling are converted into fuzzy values. The fuzzy values for abstraction, afferent coupling and efferent coupling can be 'very low', 'low', 'medium', 'high' and 'very high'. This process of converting crisp values in to fuzzy set values is termed Fuzzification. Fuzzy inference system (FIS) uses fuzzy logic to map the input to output. Mamdani fuzzy inference method is used.

After the fuzzification process is completed, we take the fuzzy sets for output variables that require Defuzzification. For Defuzzification the input will be fuzzy set and output will be a singleton value. The centroid method which gives centre of area under curve is most commonly used for Defuzzification [4].

There are many types of membership functions but we have used trapezoidal membership function. Trapezoidal-shaped is built-in membership function [6].

Syntax

$$y = \text{trapmf} [x, (a \ b \ c \ d)]$$

Description

$$f(x; a, b, c, d) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & d \leq x \end{cases}$$

The trapezoidal curve is a function of a vector, x and depends on four scalar parameters a , b , c , and d . The parameters a and d locate the “feet” of the trapezoid and the parameters b and c locate the shoulders.

4.3. Membership Function for Inputs and Outputs

To classify the packages of an object oriented system the values of three metrics, abstraction, efferent coupling and afferent coupling is taken as input as shown in Figure 5, 6 and 7 respectively. The membership function values for these metrics are – ‘very low’, ‘low’, ‘medium’, ‘high’ and ‘very high’. The range of input for abstraction is [0, 1] and for efferent coupling and afferent coupling is [0, 100]. The membership function values for output (category of packages) are – ‘very bad’, ‘bad’ and ‘balanced packages’ as shown in Figure 8. The range for output is [0, 1].

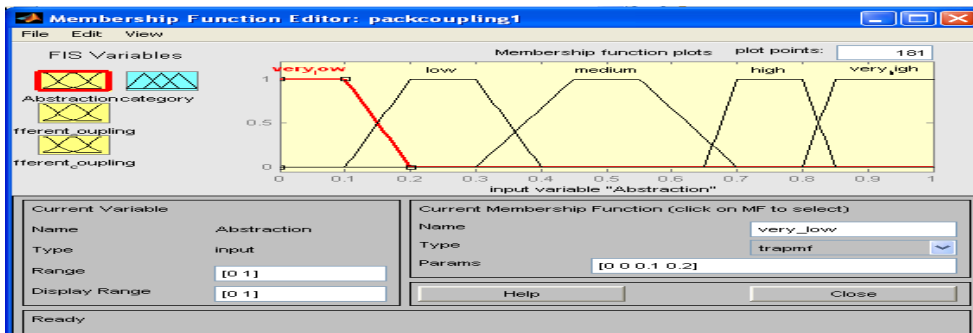


Figure 5. Membership Function for Abstraction

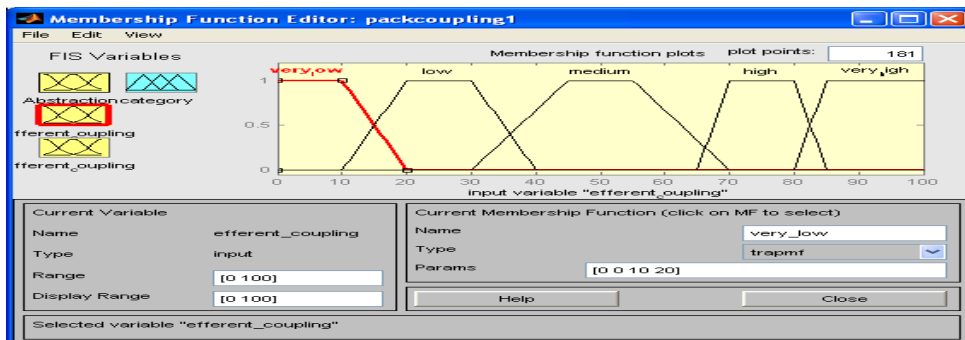


Figure 6. Membership Function for Efferent Coupling

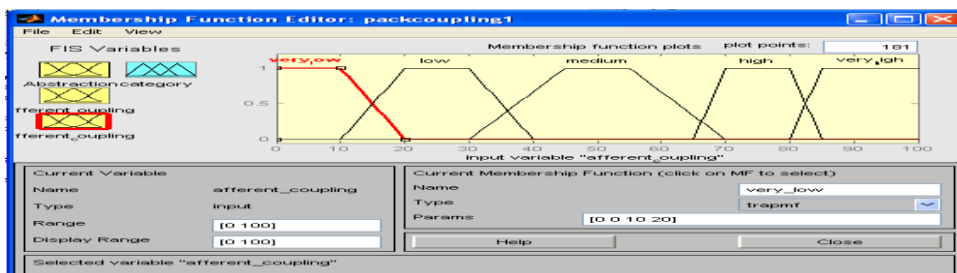


Figure 7. Membership Function for Afferent Coupling



Figure 8. Membership Function for Category

4.4. Robert Martin’s Rule Base

Robert Martin’s Rule Base has combination of 122 rules for all the inputs. Fuzzy rules are usually expressed in the form: IF variable IS property THEN action *e.g.*, If (Abstraction is very low) and (Efferent Coupling is very low) and (Afferent Coupling is very low) then (Category is Very Bad Packages). The antecedent part of rule has three input metrics, abstraction, efferent coupling and afferent coupling with linguistic variables- ‘very low’, ‘low’, ‘medium’, ‘high’ and ‘very high’. The consequent part consists of output (category of packages) with linguistic variables- ‘very bad’, ‘bad’ and ‘balanced’. For all 122 combinations packages can be classified as Desirable and Balanced Packages or Undesirable and Very Bad Packages as shown in Case-1 and Case-2. A survey is taken from n experts including project managers, software developers, research scholars, packages experts to finalize the set of rules.

Table 1. Rules for Fuzzy Model

| S. No | Abstraction | Efferent Coupling | Afferent Coupling | Category of Packages |
|-------|-------------|-------------------|-------------------|----------------------|
| 1. | VL | VL | VL | V. Bad |
| 2. | VL | VL | M | V. Bad |
| . | . | . | . | . |
| 5. | VL | L | M | Bad |
| 6. | VL | L | H | Bad |
| . | . | . | . | . |
| 18. | VL | M | L | Balanced |
| 19. | VL | H | L | Balanced |
| . | . | . | . | . |
| 25. | L | VL | L | Bad |
| . | . | . | . | . |
| 32. | L | L | L | Balanced |
| . | . | . | . | . |
| 45. | L | L | VL | Bad |
| . | . | . | . | . |
| 53. | M | L | M | Balanced |
| . | . | . | . | . |
| 66. | M | M | L | Bad |
| . | . | . | . | . |
| 73. | H | VL | L | Balanced |
| . | . | . | . | . |
| 80. | H | L | L | Bad |
| . | . | . | . | . |
| 97. | VH | VL | L | Balanced |
| . | . | . | . | . |
| 104. | VH | L | L | Bad |
| . | . | . | . | . |
| 114. | VH | M | L | V. Bad |
| . | . | . | . | . |
| 122. | VH | VL | VL | Balanced |

5. Results of Proposed Model

The categorization of the packages is difficult to analyze when the results are subjective in nature. Therefore a model is proposed in this research work based on the fuzzy logic to get the quantitative results. Finally this model is validated by comparing its results with the ‘D’ (distance from the main sequence) metric proposed by Robert martin.

5.1. Evaluation Metric and Quantitative Results

5.1.1. ‘D’ Metric: Martin claims that every stable package should be abstract because stable packages are hard to change but they should be abstract so that they can be extended. On the other hand instable package should be concrete so that their code can be

easily changed [5]. With these principals, Martin defines the relationship between A and I presented in Figure 9.

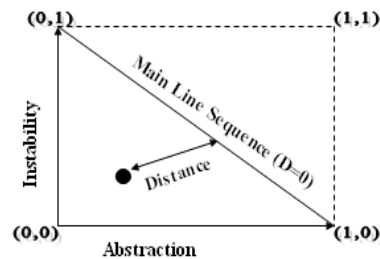


Figure 9. Distance from the Main Sequence

To categorize the Packages Martin introduced the fifth metric distance from main sequence (D) metric. The normalized distance D has a value in the range [0, 1] and is calculated by the formula:

$$D = |(A + I) - 1|$$

Where A= abstraction, I= instability.

Based on the value of D there are three cases.

Case 1: Desirable and Balanced Packages (D close to 0)

Martin says that stable and abstract packages should be at the upper left corner at (0, 1). The instable and concrete packages should be at the lower right corner (1, 0). These are desirable packages [5]. Martin claims that practically only half of the packages are desirable. Packages have degrees of stability and abstraction. The line joining the points (1, 0) to (0, 1) represents the packages that have the right proportions of abstraction and stability. This line is called the main sequence. According to Martin the Packages that lie on or close to the main sequence is not “too abstract” for its stability, nor is “too instable” for its abstractness. Such packages are “Balanced Packages” as well as “good packages” having the value of D close to 0.

Case 2: Undesirable and Very Bad Packages (D close to 1)

Martin argues that the coordinates (0, 0) represents the highly stable and concrete packages that are difficult to be extended and hard to change (Zone of Pain) [5]. The coordinate (1,1) represents the packages that are highly abstract but don't have clients. These kinds of packages are useless (Zone of Uselessness). And the packages far away from the main sequence *i.e.*, D close to 1 are very bad packages.

Case 3: Packages whose Value of D neither Close to 0 nor Close to 1

The packages whose value of D is neither close to 0 nor close to 1 are not well defined by Robert Martin [5]. Some of these Packages can have properties of balanced or bad packages.

5.2. Quantitative Results of the Proposed Model

In order to classify the packages the values of the metrics abstraction, efferent coupling and afferent coupling are given as input to the Fuzzy Model and the crisp value of Desirable, Balanced Packages as well as Undesirable and Very Bad Packages is obtained as shown in Figure 10 and 11.

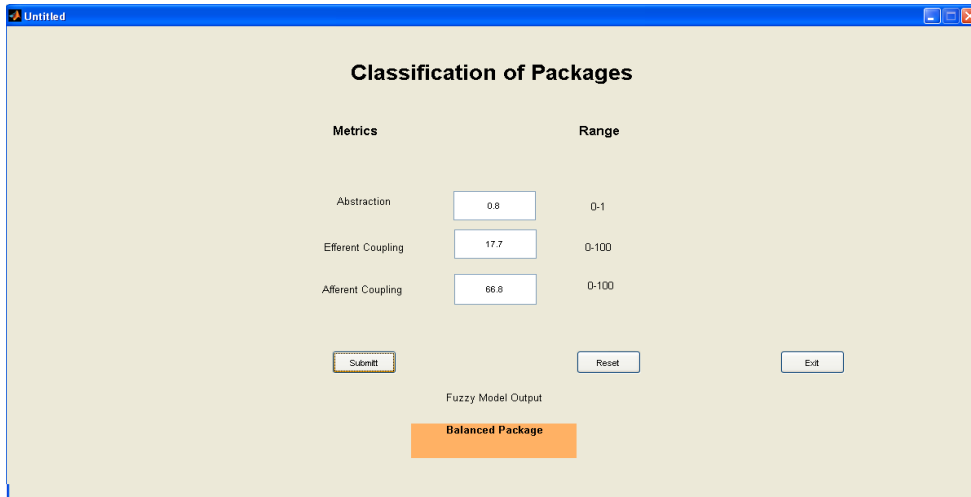


Figure 10. Interface of the Proposed Model

Figure 10 shows the interface designed for the proposed model in MATLAB. Figure 11 demonstrates the results using the Rule Viewer of Fuzzy toolbox.

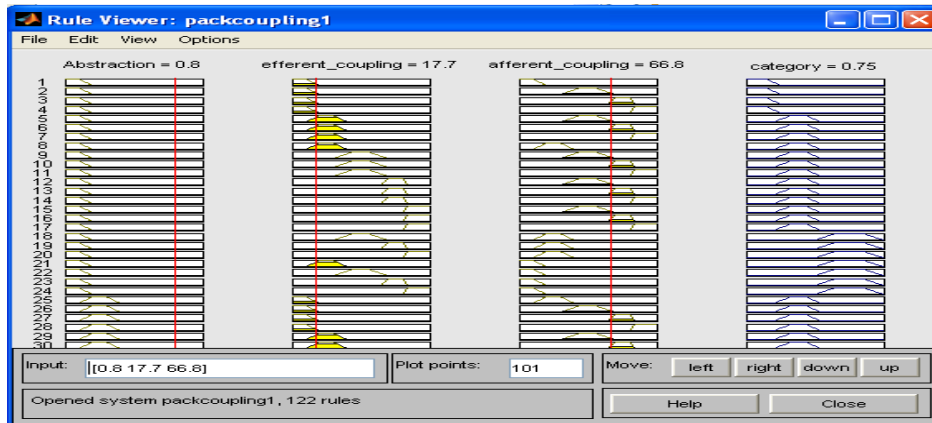


Figure 11. Category (Crisp Value) of Packages Obtained using Fuzzy Logic Toolbox

6. Conclusion

The proposed fuzzy model categorizes the packages of object-oriented software systems. The inputs for the proposed model are Abstraction, Efferent coupling and Afferent coupling on which the quality of the packages depends. Based on expert's Knowledge the vagueness in the results proposed by Robert Martin have been eliminated by designing the fuzzy rule base that contains 122 rules to classify the packages. The model is finally validated using the fifth metric D (Distance from main line sequence) given by Robert Martin. The designed model will help the software developers to design the packages that can be easily extended, maintained and reused early in the design phase of Software Development Life Cycle.

References

- [1] S. K. Dubey, A. Rana and Mridu, "Fuzzy Model for Quantifying Usability Of Object Oriented Software System", IJCSIS, vol. 10, no. 4, (2012), pp. 79-83.
- [2] A. Desai and A. Ganatra, "Fuzzy Based Refactoring Cost Resemler (FRCR) Model for Object Oriented Systems", International Journal of Computer Theory and Engineering, vol. 4, no. 2, (2012), pp. 251-258.

- [3] S. K. Dubey, A. Rana and A. Sharma, "Usability Evaluation of Object Oriented Software System Using Fuzzy Logic Approach", International Journal of Computer Applications, vol. 43, no. 19, (2012), pp. 1-6.
- [4] X. Liu, "Parametrized Defuzzification with maximum entropy weighting function- Another view of the Weighting function expectation model", Mathematical and Computer Modelling, vol. 45, (2007), pp. 177-188.
- [5] R. Martin, "OO Design Quality Metrics An analysis of dependencies", (1994).
- [6] J. Mago, S. Kaur and S. Kumar, "Fuzzy Model to Analyze and Interpret Object Oriented Software Design", International Journal of Electrical, Electronics and Computer Engineering, vol. 1, no. 1, (2012), pp. 41-46.
- [7] P. M. Shanthi and K. Duraiswamy, "An Empirical Validation of Software Quality Metric Suites on Open Source Software For Fault Proneness Prediction in Object Oriented system", European Journal of Scientific Research, vol. 51, no. 2, (2011), pp. 168-181.
- [8] S. Hyrnsalmi and V. Leppanen, "A Validation of Martin's Metric", Proceedings of 11th Symposium on Programming Languages and Software Tools and 7th Nordic Workshop on Model Driven Software Engineering, (2009), pp. 87-101.

Authors



Gurpreet Kaur, Assistant Professor (CSE, Department) at Desh Bhagat Foundation Group of Institutions, Moga (Punjab).



Deepak Sharma, HOD (CSE, Department) at Desh Bhagat Foundation Group of Institutions, Moga (Punjab).