International Journal of Mathematics and
Computer Science, **17**(2022), no. 3, 1227–1240

Application of Fuzzy Logic Model for Software Quality Measurements

Safwan O. Hasoon[1], **Fatima Mohammed Rafie Younis**[2]

[1]Department of Software
Faculty of Computer Science and Mathematics
University of Mosul
Mosul, Iraq

[2]Department of Mathematics
Faculty of Basic Education
University of Mosul
Mosul, Iraq

email:Dr.safwan1971@uomosul.edu.iq, fatimamsw17@uomosul.edu.iq

(Received February 1, 2022, Accepted April 6, 2022)

Abstract

Software quality is very important for the software engineering life cycle in the design stage. In this paper, the EA program draws UML and converts them to XMI format with software quality metrics applied. The Metrics for Object-Oriented Design (MOOD) involve Coupling Factor, Polymorphism Factor, Inheritance Factor Method, Hiding Factor Method, and Hiding Factor Attribute could have been measured in addition to inheritance factor methods and attributes. The intelligent techniques are used through fuzzy logic to determine the level of software quality depending on metrics. The application of fuzzy logic in software is flexible and updated according to system requirements. Software quality is very important for the software engineering life cycle in the design stage.

-

Key words and phrases: Fuzzy Logic, Design Phase, Quality Assurance, MOOD.
AMS (MOS) Subject Classifications: 68T27.
ISSN 1814-0432, 2022, http://ijmcs.future-in-tech.net

1 Introduction

The role of the design stage is to define the representation and function formula for the system to appear aesthetically and functionally. In the software development life cycle, the primary role of the design stage consists of a set of related objects for each object that has one or more responsibilities. The main goal of creating the design is to determine how the system works; in addition, it should be easy to understand by the user [4].

Definition 1.1. *Quality assurance (QA) means Agreements, criteria, and guidelines that must meet determined requirements [3].*

Without a measure, errors cannot be known early until after the completion of the product and, for this, measures are necessary during the development of the product [10].

2 Class-Oriented Metricsthe MOOD Metrics Suite

The MOOD Metrics defined by Harrison, Counsell, and Nithi [Har98b] are intended to develop a brief of the inclusive quality of an object-oriented project. The MOOD metrics suite involves 6 metrics [10], basic periods for MOOD Model table 1.

2.1 Method Inheritance Factor (MIF)

The MIF is calculated by way of a fraction. The numerator denotes the number of methods inherited by a class. The denominator is the entire number of existing methods (locally defined and, in addition, inherited) for all classes [10].

$$MIF = \frac{\sum_{i=1}^{Ac} N_i(E_i)}{\sum_{i=1}^{Ac} N_a(E_i)} \tag{2.1}$$

Anywhere,
$N_a(E_i) = N_d(E_i) + N_i(E_i)$
$N_d(E_i) =$ No. of methods defined in a class
$N_a(E_i) =$ No. of available methods in Ei.
$N_i(E_i) =$ No. of inherited methods.
$AC =$Total no. of classes.

2.2 Attribute Inheritance Factor (AIF)

The MIF is calculated by way of a fraction. The numerator denotes the number of attributes inherited by a class. The denominator is the entire number of existing attributes (locally defined and, in addition, inherited) for all classes [10].

$$AIF = \frac{\sum_{i=1}^{Ac} A_i(E_i)}{\sum_{i=1}^{Ac} A_a(E_i)} \tag{2.2}$$

Anywhere,
$A_a(E_i) = A_d(E_i) + A_i(E_i)$
$A_d(E_i)$ = No. of attributes defined in a class
$A_a(E_i)$ = No. of available attributes in Ei.
$A_i(E_i)$ = No. of inherited attributes.
AC = Total no. of classes.

2.3 Method Hiding Factor (MHF)

MHF is calculating the hiddenness methods in classes. The hiddenness method denotes the ratio of entire classes from which the method is hidden [2].

$$MHF = \frac{\sum_{i=1}^{Ac} \sum_{i=1}^{Nd(ei)} (1 - V(N_{mi}))}{\sum_{i=1}^{Ac} N_d(E_i)} \tag{2.3}$$

Anywhere:
$N_d(E_i)$ = No. of methods defined inE_i
Ac =Total no. of classes.
$V(N_{mi})$ = Visibility value of the method for all Ei, a method value private=1, public=0, protected = Size of the inheritance tree / Number of classes.

2.4 Attribute Hiding Factor (AHF)

MHF is the measure of the hiddenness attributes in classes. An attribute's hiddenness denotes the ratio of the entire classes from which the attribute is hidden [1].

$$AHF = \frac{\sum_{i=1}^{Ac} \sum_{i=1}^{Ad(ei)} (1 - V(A_{mi}))}{\sum_{i=1}^{Ac} A_d(E_i)} \tag{2.4}$$

Anywhere:
$A_d(E_i)$ = No. of attributes defined in E_i
Ac =Total no. of classes.

$V(A_{mi})$ = Visibility value of the attribute for all Ei, a attribute value private=1, public=0, protected = Size of the inheritance tree / Number of classes.

2.5 Coupling Factor (CF)

The coupling factor (CF) is calculated by way of a fraction. The numerator denotes the amount of non-inheritance couplings. The denominator is the number of classes squared subtracts the number of classes [10].

$$CF = \frac{\sum_{i=1}^{AC}[\sum_{j=1}^{Ac} is - cleint(E_i, E_i)]}{Ac^2 - Ac} \tag{2.5}$$

Anywhere:
AC = Total no. of classes.
$is - client$ = 1, if each class contains one or more links except inheritance; otherwise =0, if it has no or only inheritance links.

2.6 Polymorphism Factor (PF)

PF is calculated by way of a fraction. The numerator denotes the total amount of overridden methods. The denominator is the number of new methods multiplied by descendants for all classes [1].

$$PF = \frac{\sum_{i=1}^{AC} N_o(E_i)}{\sum_{i=1}^{AC}[Mn(E_i) \times DC(E_i)]} \tag{2.6}$$

Anywhere,
$Nn(E_i)$ = No. of new methods.
$No(E_i)$ = No. of overriding methods.
$DC(E_i)$ = No. of subclasses for Ei.
AC = Total no. of classes.

3 Fuzzy logic

Fuzzy logic was defined in 1965 by Dr. Lotfi Zadeh as the fuzzy set theory as a form of probabilistic logic that deals with approximate values ranging between 0 and 1 when using linguistic variables, in contrast to binary logic whose values are either 0 or 1 [6].

Table 1: Standard periods for MOOD Model [2]

Metrics	Minimum	Mean	Maximum
MHF	12.7%	17.3%	21.8%
AHF	75.2%	81.9%	100%
MIF	66.4%	72.4%	78.5%
AIF	52.7%	59.5%	66.3%
POF	2.7%	6.1%	9.6%
CF	4.0%	7.6%	11.2%

3.1 Membership Function

This must be determined first and used to calculate the degree of affiliation of the input value. Its range is [0, 1] with the value zero being the entry which does not belong to the fuzzy group and the value one being the entery belonging entirely to the fuzzy group. There are several types of affiliation functions, the most common being the triangular function [6]. The Fuzzy logic contains three basic operations [9]:

1. Fuzzification
   This is considered the first step in fuzzy logic where the inputs are assigned to the groups they belong to, called fuzzy groups.

2. Fuzzy linguistic inference
   The fuzzy inference engine includes the rule base created by mapping fuzzy associations of the input's fuzzy subgroups to the output's fuzzy subgroups.

3. Defuzzification
   This is considered the last step in fuzzy logic, the final product being a clear numerical value.

4 Related Work

1. In 2015, Laheeb Ibrahim and Khalil Ibrahim [8] designed three tools of object-oriented design metrics quality used in MOOD and MEMOOD. The first tool is KDM which incomes the XMI document for the UML class diagram built by EA as entering. The second tool is KRS which incomes the XML document generated by KDM entering, analyzing, and providing a metrics document report. The third tool is KDB which

is used to generate similar XML documents. The XML document is entered for analysis and supplies metrics in the database can be used as old data.

2. In 2012, Mago and Kaur [5] applied fuzzy logic to assess quality early in the design stage of the software engineering life cycle, using the CK metric suite. They explained the six metrics WMC, RFC, LCOM, CBO, DIT, and NOC and used thetTriangular membership function and the MAMDANI fuzzy inference engine.

3. In 2020, Agina, Muchiri, and Muthoni [7] used Fuzzy logic to foretell the reusability of Object-Oriented software Abstraction, Cohesion, and Coupling at the source-code level existing legacy systems to produce advanced systems that meet customer requirements. This model was verified using the AHP framework.

4. In 2013, Aggarwal, Verma, and Mishra [13] emphasized a group of Object-Oriented metrics. The quality and efficiency of an Object-Oriented design can be measured using Object-Oriented metrics.

5. In 2010, Ahmed, Hassan, Soliman, and Sewisy [11], [12] worked on two parts:
First, taking the XMI document files for the UML class diagram and computing three different quality metrics sets (CK, LK, and MOOD). Secondly, displaying vulnerabilities and forcing the three sets and proposing a combination metrics HCLKM set which is a combination of the three packages. HCLKM consisted of eight class-level metrics work (AC, MC, CBO, RFC, LCOM and DIT).

5 Proposed system

This paragraph explains the steps and mechanisms used to build the software engineering tool at the design stage. It can calculate the software quality of the UML-Class Diagram model of the design phase. Using Enterprise Architect 7.5 for converting to XMI.

1. Using UML-Class Diagram by EA Designer.

2. Convert it into a file with an XMI extension.

3. Open the file for reading.

4. Read one line of text at a time.
5. Lines are divided into words and symbols.
6. Specified name class.
7. Initial works a counter to count No. of classes.
8. Define No. of attributes and operations for each class.
9. Specify the type of operations and attributes, whether public or private or semi-public.
10. Define hereditary classes by relationship generalization.
11. Extract the subtype and supertype value.
12. No. of attributes and processes to inheritance and non-inheritance for each class is limited.
13. If the classes are not finished, return to step 4.
14. MHF and AHF are accounted.
15. MIF and AIF are accounted.
16. We define the standard functions between subtype and supertype and the new functions of the class.
17. We determine the No. of children per super.
18. We calculate the POF.
19. If each class has more than one correlation (except inheritance), is-client = 1; otherwise 0, when no correlation or is only correlated to inheritance.
20. CF accounted.

5.1 Examples

In this section, quality standards MOOD Metrics have been applied to an illustrative example as shown in figure 1.

The table 2 metrics results of example.

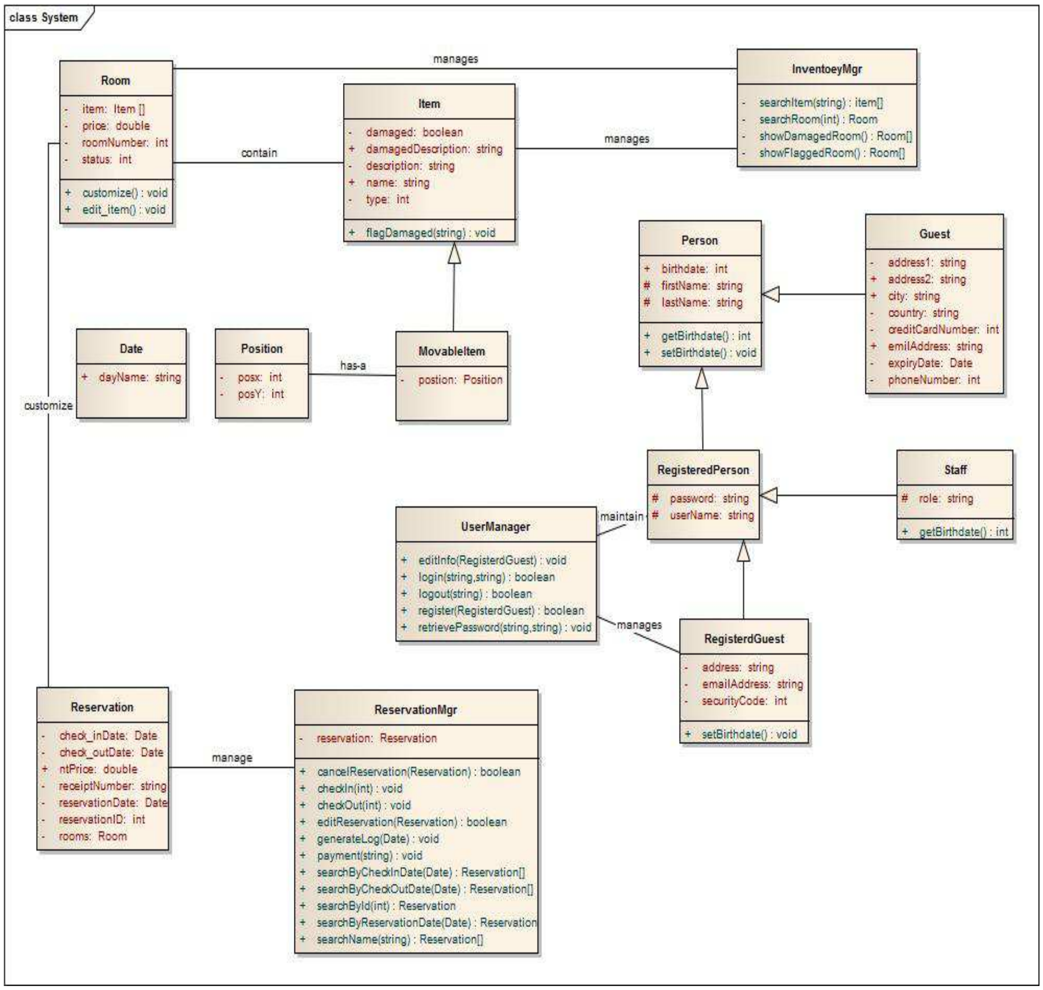

Figure 1: Class hotel management

Table 2: The metrics of class hotel

Metrics	Value
MHF	14.81%
AHF	65.79 %
MIF	64.29 %
AIF	47.73 %
POF	40 %
CF	5.49 %

5.2 Fuzzy Logic Model

In this model, the fuzzy logic toolbox in Matlab language was used to measure the system's quality.

- Fuzzification
  There are six input scales. Each input is expressed in linguistic terms, low, medium, and high. The output is expressed as poor, acceptable, average, good, very good, and excellent.

- Membership Functions
  The Triangular membership function for both input and output variables shown in figure 2, the input metric MHF, AHF, MIF, AIF, POF, and CF divided into three phases low, medium, and high; the range of the input metrics shown in table 3.

Table 3: The range of three possible values

Metrics	low	medium	High
MHF	10.4-15%	15-19.5%	19.5-24.1%
AHF	71.9-79%	73-91%	91-109%
MIF	63.4-69.4%	69.4-75.4%	75.5-81.5%
AIF	49.3-56.1%	56.1-62.9%	62.9-69.7%
POF	1-4.4%	4.4-7.8%	7.8-11.3%
CF	2.2-5.8%	5.8-9.4%	9.4-13%

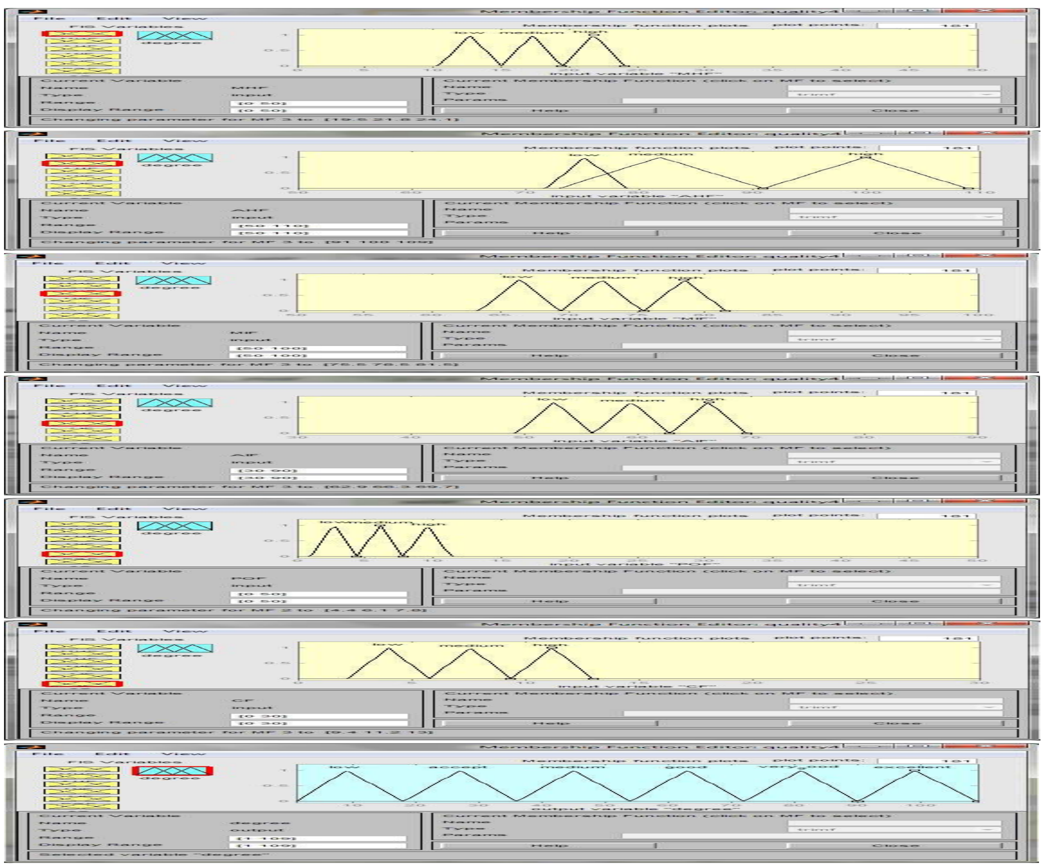

Figure 2: Membership function for input and output

- Rules
  The model consists of 6 entries. Each entry has 3 functions defined. The knowledge base contains 729 rules using the MAMDANI inference-type as shown in figure 3.

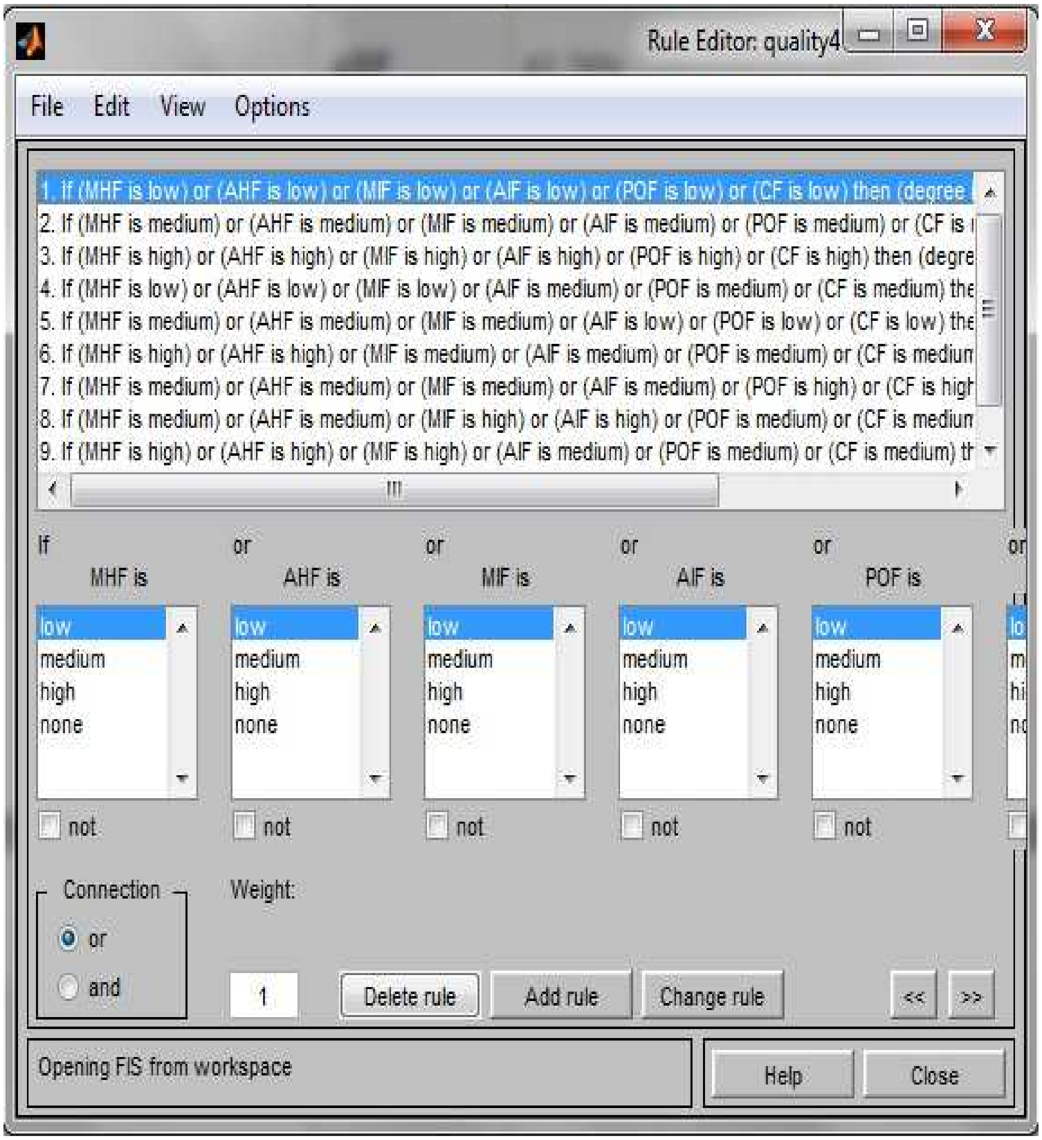

Figure 3: The Rules

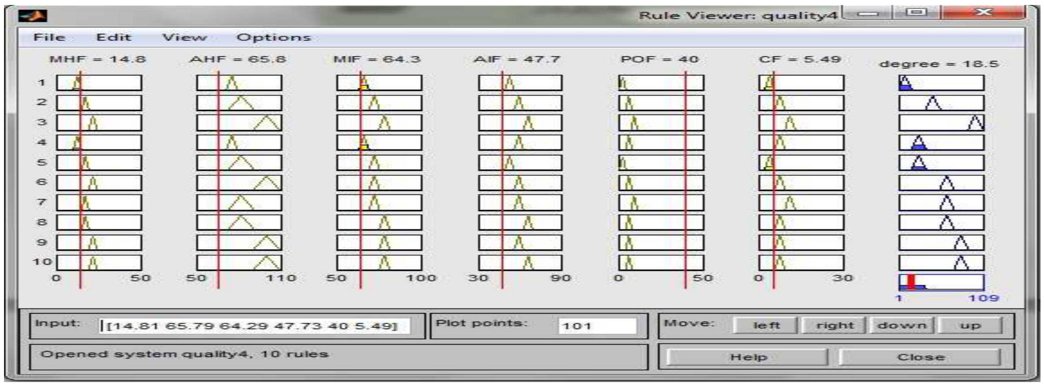

Figure 4: The Rules Viewer

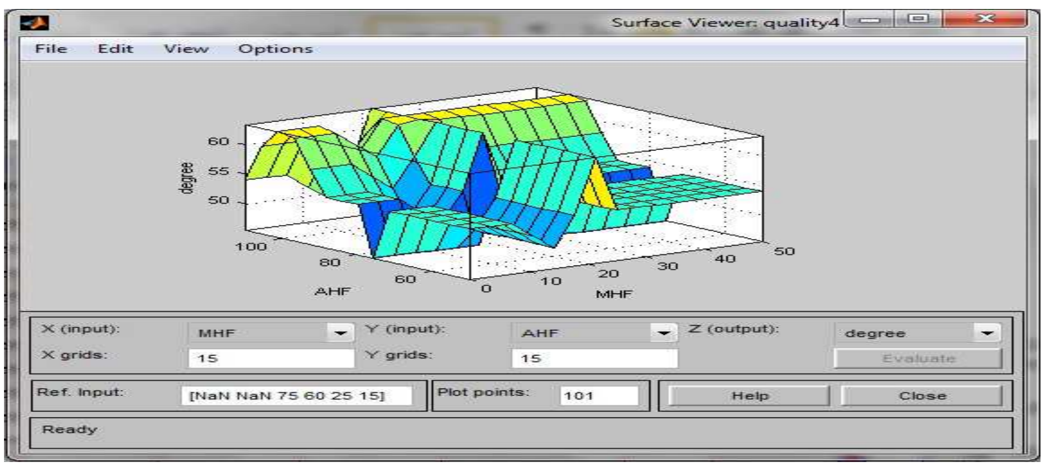

Figure 5: The 3D surface

- Defuzzification
  In this model, the output of the proposed system divided into low, acceptable, medium, good, very good, and excellent using the centroid technique relying upon the MAMDANI fuzzy model as shown in figure 4 showing the curve 3D surface viewer in figure 5.

6 Conclusion

In this study, the artificial intelligence technique was used by applying fuzzy logic technology in software engineering by finding software evaluation at the design stage using MOOD standards, where if the evaluation is weak, acceptable, or average, the designer can change the model in terms of encapsulation, inheritance of functions and data, a diversity of forms and interconnectedness between the classes, the new evaluation model was introduced again to show a final evaluation of the system to produce a strong system.

References

[1] Daniel Rodriguez, Rachel Harrison, An overview of object-oriented design metrics, 2001.

[2] Fernando Brito e Abreu, Design metrics for object-oriented software systems, 1995.

[3] Ian Sommerville, Software engineering, 9th Edition, 2011.

[4] Ivan Marsic, Software engineering, 2012.

[5] Jagmohan Mago, Parwinder Kaur, Analysis of quality of the design of the object oriented software using fuzzy logic, International conference on recent advances and future trends in information technology, Citeseer, (2012), 21–25.

[6] Mohsen Kahani, Expert System & Knowledge Engineering in Wikipedia, 2012.

[7] Kevin A. Onyango, Geoffrey M. Muketha, Elyjoy Muthoni Micheni, A Metrics-Based Fuzzy Logic Model for Predicting the Reusability of Object-Oriented Software, Blue Eyes Intelligence Engineering and Sciences Publication, 2020.

[8] Laheeb Mohammed Ibrahim, Khalil Ahmed Ibrahim, Constructing an Add-in Tool for Enterprise Architect v 7.5 to Measure the Quality of Object Oriented Design (Class Diagram), International Journal of Computer Science and Information Security, **13,** no. 7, LJS Publishing, (2015), 72.

[9] Michael Negnevitsky, A guide to intelligent systems, 2005.

[10] Roger S. Pressman, Software engineering: a practitioner's approach, 2005.

[11] Saddam H. Ahmed, Taysir Hassan A. Soliman, Adel A. Sewisy, A hybrid metrics suite for evaluating object-oriented design, International Journal of Software Engineering, **6,** no. 1, (2013), 65–82.

[12] Maad Shatnawi, Anas Shatnawi, Zakarea AlShara, Ghaith Husari, Symptoms-based Fuzzy-Logic Approach for COVID-19 Diagnosis, International Journal of Advanced Computer Science and Applications, **12,** no. 4, (2021).

[13] Mansi Aggarwal, Vinit Kumar Verma, Harsh Vardhan Mishra, An Analytical Study of Object-Oriented Metrics (A Survey), International Journal of Engineering Trends and Technology, Citeseer, **6,** no. 2, (2013), 76–83.