International Journal of Electrical, Electronics and Computer Engineering **1**(1): 41-46(2012)

ISSN No. (Online) : 2277-2626

Fuzzy Model to Analyze and Interpret Object Oriented Software Design

Jagmohan Mago*, Sarabjit Kaur** and Kumar Saurabh**

** Apeejay College of Fine Arts, Jalandhar, (PB)*
***Department of Computer Science Engineering, PTU Kapurthala, (PB)*
**** Department of Computer Science Engineering, RIET, Phagwara, (PB)*

(Recieved 10 April, 2012 Accepted 4 May, 2012)

ABSTRACT : Building a quality system has been most driving goal of all software engineering efforts over few decades. The lack of design guidance may affect quality of the system. To produce high quality Object Oriented applications a strong emphasis on design aspects is highly necessary. So, this research work aims to facilitate thedevelopers to appreciate design aspects of software in order to improve the quality of the software. In this paper wepropose a system based on fuzzy logic to assess the quality of OO design, uses the CK metric suite and Mamdani Inference Engine.

In this research, a decision making system, based on fuzzy inference mechanism as proposed by Mamdani, is presented. Fuzzy logic is a form of many-valued logic, designing a knowledge base modelwith six input metrics are considered with each metric being defined by three membership functions LOW MEDIUM HIGH and rule base consists of all the possible combinations i.e. 734 rules to assess the Quality of design of the software. The results proposed by the system are compared with number of industrial software tools such as Analyst4j and ViZZAnalyzer. It has been found that the results produced by the system are better than the results produced by the software tools. This was validated and verified by the human experts such as professors and the developers in the field.

Keywords: CK Metrics, Fuzzy Logic, Mamdani Model, OOD Metrics. Quality.

I. INTRODUCTION

Software engineering is a profession dedicated to analysis, designing, implementing and modifying software so that we develop software of high Quality and fast to build. Object oriented Development requires not only a different approach to design and implementation, but also require to software metrics. A larger body of software quality metrics has been developed and numerous tools exist to collect metrics from program representations. It allows the user to select the tools best suited depending upon its handling, tool support, or price.. Also, software Quality estimation model allows the software development team to track and detect design flaws in Object Oriented software systems. Such type of Quality models will helps developers in building best quality programs [19]. Software Quality is the degree to which processes a desired combination of attributes such as reliability, maintainability, efficiency, portability, usability, and reusability [13, 22]. As OOD methodologies gaining popularity, investigate OOD metrics with respect to software quality [2, 8], although various researchers have proposed many metrics suites. If the metrics are properly defined, we can avoid problems that will be more expensive to correct during latter phases of object oriented Software Development. The proposed system is developed under MATLAB development environment which can help designing interactive GUI as well as supports various tools boxes suchas Fuzzy Logic Tool Box that is employed in proposed system. The presented system employs fuzzy approach to develop a decision making system for evaluating Quality of software [11, 24]. The major benefits of proposed system are accuracy will be high enhanced; users can easily know and understand quality of software. So, this paper proposes a system based on fuzzy logic to assess the quality of OO design, uses the CK metric suite and MAMDANI INFERENCE ENGINE.

II. LITERATURE REVIEW

Object Oriented Design and Development is an interesting area of current research and many authors have done great deal of work in recent years. The number of approaches has been developed to address the problem of detecting and correcting design flaws in OO software system using metrics. Chidamber and kemerer (CK) [1, 3, 5, 20] are the researchers who mostly referred; six metrics were defined by them. They are Weighted Methods per Class (WMC), Response sets for class (RFC), Lack of Cohesion in Methods (LCOM), coupling between Object Classes (CBO), Depth of Inheritance Tree of a class (DIT) and Number of Children of a class (NOC). CK Metrics were defined for evaluating design complexity in relation to their effect on Quality factors like usability, maintainability, functionality, reliability etc. This makes them more suited to Object-oriented paradigm since, in object-Oriented Design emphasis is more on the design phase of software system.

Table 1: Object-Oriented Metrics from Various Sources.

Source	*Metrics*
Chidamber *et al.* [1,3]	WMC,RFC,LCOM,CBO,DIT,NOC
Lorenz *et al.* [15]	Class Size, Class Inheritance, Class Internal
Abreu [26]	MIF, AIF, MHF, AHF, POF, COF
Rosenberg *et al.* [6]	CC, LOC, CP, WMC, RFC, LCOM, CBO, DIT, NOC
Li W. *et al.* [17]	NAC, NLM, CMC, NDC, CTA, CTM
Bansiya *et al.* [10]	ANA, CAM, CIS, DAM, DCC, MOA, MFA, NOP,DSC, NOH, NOM

III. SOFTWARE QUALITY

"Quality refers to the inherent or distinctive characteristics or property of object, process or other thing. Such characteristics or properties may set things apartfrom other things, or may denote some degree of achievement or excellence" [25].

The factors (McCall,1977) that affect software quality can be categorized into two groups [23]:

1. Factors that can be measured directly (e.g. defects recovered during testing)
2. Factors that can be measured only indirectly (Usability and maintainability)

One of the earliest software product quality models were suggested by McCall (1977) and his colleagues. They proposed a useful categorization of factors that in figure:

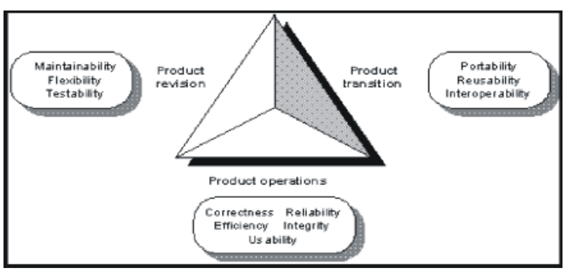

Fig. 1. McCall's Quality factors.
Source: (Pressman, Fifth Edition) [21].

IV. METRICS

The importance of knowing Quality of the design phase there is need of metrics [16] that measures the goodness of design and it provides the designer with improved insight that leads to a higher level of quality. We categorized metrics into two groups: project based and design based metrics [17, 23]. Project based metrics are used to predict project needs, such as staffing levels and total efforts. Design based metrics contain traditional metrics and object oriented metrics [4, 7, 12, 14, 18, 23]. Chidamber and Kemerer [1, 3] are considered the best OO design metrics.

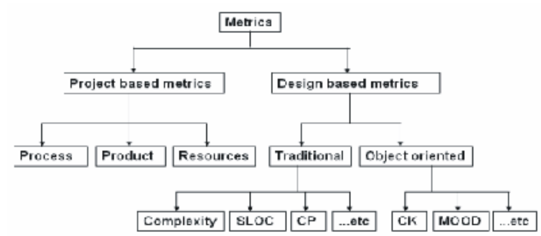

Fig. 2. Metrics Hierarchy.

Table 2: Relationship between CK Metrics and Software Quality concepts.

CBO	Number of couplings between a certain class and all other classes	Efficiency and Reusability
RFC	Number of method that can be performed by a certain class regarding a received message	Understandability, Maintainability and Testability
WMC	Number of methods of certain class without inherited methods	Understandability, Maintainability and Reusability
DIT	Maximal depth of certain class in an inheritance structure	Efficiency, Reusability and Testability
NOC	Number of direct subclasses of certain class	Efficiency, Reusability and Testability
LCOM	Number of disjunctive method pairs of a certain class	Efficiency and Reusability

V. PROPOSED SYSTEM

Software quality must be addressed during the whole process of software development [11]. All of the six input metrics CBO, RFC, WMC, DIT, NOC, and LCOM measures the different aspects of Software Quality. The CK metric suite is growing in the software industry. This is reflected in the increasing number of industrial software tools, that enables automated computation of these metrics.

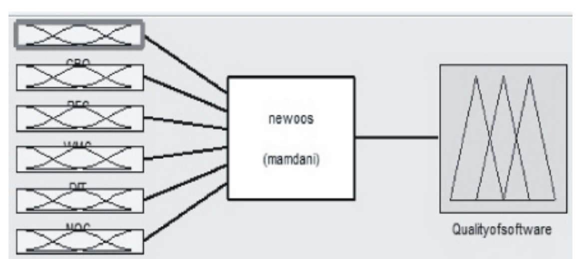

Fig. 3. Proposed Fuzzy Logic based Framework.

Fuzzy logic is a form of many-valued logic or probabilistic logic; it deals with reasoning that is approximate rather than fixed and exact [9]. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false.Furthermore, when linguistic variables are used, these degrees may be managed by specific functions. The input variables in a fuzzy control system are in general mapped into by sets of membership functions similar to this, known as “fuzzy sets”. The process of converting a crisp input value to a fuzzy value is called “fuzzification” [9, 24]. For a fuzzy system whole final output need to be in crisp (no fuzzy) form, a step is needed to convert the finally combined fuzzy conclusions into crisp onethis is called defuzzification. The working of fuzzy logic can be summarized into four steps:

Step 1 : Fuzzification defines membership functions for linguistic terms namely: CBO,RFC, WMC, DIT, NOC and LCOM.

Step 2 : Designing the knowledge base by using the fuzzy rules using an if-then format. If (RFC is HIGH) and (CBO is low) and (DIT is MEDIUM) and (NOC is MEDIUM) and (WMC is MEDIUM) and (LCOM is MEDIUM) . . . then (QUALITY is MEDIUM).

Step 3 : In Rule Evaluation,several rules need to be fired out of the 734 rule knowledge base on the basis of input presented by the designer. This leads to decisions regarding quality of design of software.

Step 4 : Defuzzification, finding the crisp outcome for output variables since the proposed system needs to inform the designer, level of quality of design as LOW MEDIUM HIGH should be performed.

A. Working of System

In this system, all the crisp values of six input metrics are considered, each input metric defined by three membership function [9] as LOW MEDIUM HIGH. Each of the inputs is mapped to a membership value in the interval [0, 1]. The value zero is used to indicate the complete non membership and value one indicate the complete membership and value in between are used to represent intermediate degrees of membership. The first step is the fuzzification of all inputs by transforming the crisp values into fuzzy values. All the input metrics are considered and combined with AND operator. The MIN/MAX membership operator helps to determine the degree of membership in Mamdani inference Engine. The technique of defuzzification is Centroid which transforms the fuzzy values to linguistic variable.

The types of membership function for all input metrics are trapezoidal membership function.Trapezoidal-shaped built-in membership function.

Syntax

y = trapmf [x, (a b c d)]

Description

The trapezoidal curve is a function of a vector, x and depends on four scalar parameters a, b, c, and d, as given by

$$f(x;a,b,c,d) = \left\{ \begin{array}{ll} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \frac{d-x}{d-c}, & c \le x \le d \\ 0, & d \le x \end{array} \right\}$$

The parameters a and d locate the “feet” of the trapezoid and the parameters b and c locate the shoulders.

Coupling between Object Classes (CBO)

CBO	*a*	*b*	*c*	*d*
Low	0	0	5.712	8
Medium	5.76	9	25.740	30
High	26	30.596	50	50

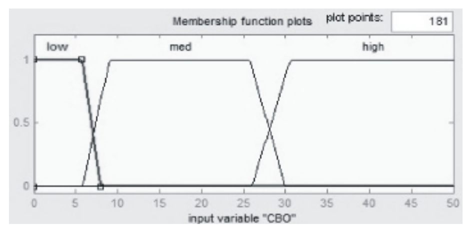

Fig. 4. Membership function for CBO.

Responsefora Class (RFC)

RFC	*a*	*b*	*c*	*d*
Low	0	0	26.13	35
Medium	26.9	36.7	60.5	71.65
High	63.4	72.201	100	100

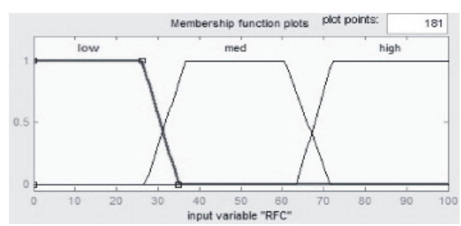

Fig. 5. Membership function for RFC.

Weighted Methods per Class (WMC)

WMC	*a*	*b*	*c*	*d*
Low	0	0	12.8	16.721
Medium	13.5	17.2	33.23	36.1
High	33.2	36.61	51.1	51.1

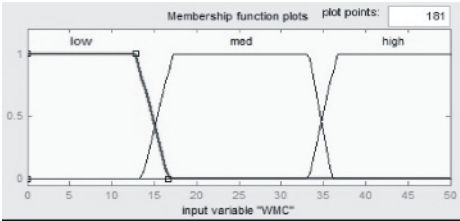

Fig. 6. Membership function for WMC.

Depth of Inheritance (DIT)

DIT	*A*	*b*	*c*	*d*
Low	0	0	5.009	5.94
Medium	5.3	5.98	7.69	8.54
High	7.59	8.66	10.11	10.22

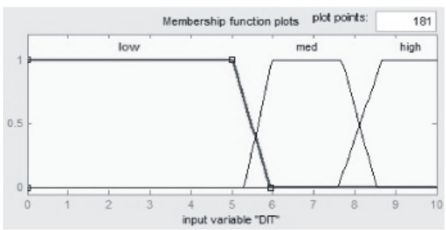

Fig. 7. Membership function for DIT.

Number of Children (NOC)

NOC	*A*	*b*	*c*	*d*
Low	0	0	3.955	6
Medium	4.18	6.2	15.361	18.9
High	14.9	19.273	31.6	31.6

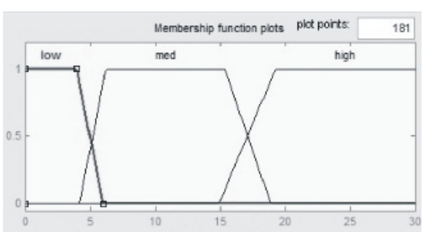

Fig. 8. Membership function for NOC.

Lack of Cohesion of Methods (LCOM)

LCOM	*A*	*b*	*c*	*d*
Low	0	0	0.2985	0.499
Medium	0.28	0.5	0.72	0.89
High	0.75	0.87	1	1

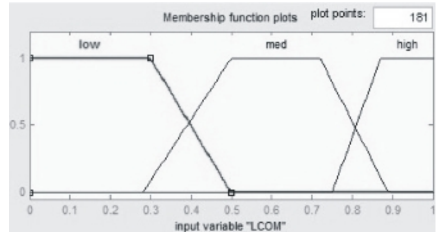

Fig. 9. Membership Function for LCOM.

The curves for all input metrics are defined by threshold values .Each input metric is further classified as LOW, MEDIUM AND HIGH.

Metrics	*WMC*	*DIT*	*NOC*	*CBO*	*RFC*	*LCOM*
Threshold values	0-50	0-10	0-30	0-50	0-100	0-1

Threshold values for CK metric suite

B. Grading of Membership Functions

As inputs are given to the fuzzification module and after fuzzification of given values, we need to try to find out which fuzzy set they actually belongs as LOW, MEDIUM, HIGH as $L = 3$, $M = 2$, $H = 1$. For this we need to calculate average values of trapezoidal curves for each input metric.

Table 3: Average values of Input Metrics.

Average value	*Low*	*Medium*	*High*
CBO	X<7.15	7.15<=X<=28.07	X>28.07
WMC	X<15.06	15.06<=X<=34.78	X>34.78
RFC	X<31.4	31.4<=X<=67.03	X>67.03
NOC	X<5.08	5.08<=X<=17.11	X>17.11
DIT	X<5.57	5.57<=X<=8.12	X>8.12
LCOM	X<0.394	0.394<=X<=0.81	X>0.81

C. Rules

The structure of the fuzzy rule can be divided into two parts: an if-part (also referred as the antecedent part) and a then-part (also referred as the consequent part) IF <antecedent> THEN <consequent>. The antecedent describes a condition whereas consequent describe a conclusion.

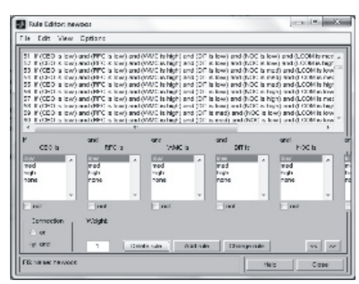

Fig. 10. Rules.

In this systemall six input metrics are considered with each metric being defined by three membership functions as Low Medium High and rule based inference unit consists of all possible combinations *i.e.* 734 rules. Qualify the quality of software by assigning the values to LOW MEDIUM HIGHand divide the sum by six to get average values.

QUALITY = values of (CBO + RFC + WMC + DIT + NOC + LCOM)/6

Lower the value of the parameter, Higher will be quality of design of software. Higher the value of parameter, lower will be Quality of design of software.

MID POINT	***Low***	***Medium***	***High***
Quality of software	X<1.67	1.67<=X<=2.27	X>2.27

VI. DEFUZZIFICATION

To determine the proper metric suite for object oriented software. It is required that the system produce a crisp result, *i.e.* precise decision rather than vague interpretation. For this reason, we use Centroid method (COA) which calculates the weighted average of a fuzzy set.

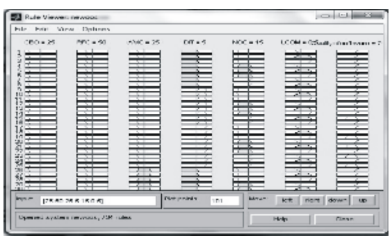

Fig. 11. Defuzzification.

VII. OUTPUT

Fig. 12. Output.

Fig. 13. Output.

Fig. 14. Output.

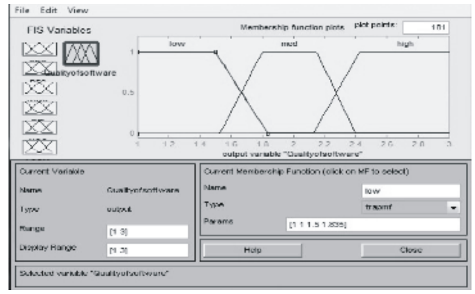

Fig. 15. Output membership function.

VIII. CONCLUSION

In this paper, proposed a software Quality system based on fuzzy Logic for carrying out the quality estimation early at design phase of Software Development Life Cycle. It is a clear solution to one of the major problem of analysis and hence side lining the point that needs to be deeply considered for production of ideally desired software.The system clearly inclines the reliability to know the quality of the software on the basis of the defined object oriented metrics. After the analysis of Quality of software a lot of work can be done on idealizing it. In future, we can planan extensive validation of our metric suite on a variety of different industrial environment and collect data to analyze the reliability of the proposed system.

REFERENCES

[1] Chidamber, Shyam and Kemerer, Chris, "A Metric Suite for Object Oriented Design", *IEEE Transaction on Software Engineering*, June, (1994), pp. 476-492.

[2] Hudli, R., Hoskins, C., Hudli, A., "Software metrics for Object Oriented Designs", IEEE, (1994).

[3] Chidamber, S.R., and kemerer, C.F. "Towards a Metrics Suits for Object oriented Design," Proc.Conf. Object oriented programming systems, *Languages, and applications*, vol. **26**, no. 11, (1991), pp. 197-211.

[4] S. Jarmali, "Object Oriented Metrics", Sharif University of Technology, Jan (2006).

[5] Dr. M.P. Thapaliyal, G. Verma, "Software Defects and Object Oriented Metrics-An Empirical Analysis", *International Journal of Computer Applications*, vol. **9**, Nov. (2010).

[6] L.H. Rosenberg and L. Hyatt, "Software Quality metrics for Object Oriented Environments", Crosstalk journal, (1997).

[7] Shyam R.Chidamber and Chris F. Kemerer "Metrics suite for Object Oriented Design", *IEEE Transaction on Software Engineering*, vol. **20**, no.6, June (1994).

[8] Sanjay Kumar Dubey,andProf. (Dr.) Ajay Rana "A Comprehensive Assessment of Software Systems Using metric approach", *International Journal on Computer Science and Engineering*, Vol. **02**, No. 08, (2010), 2726-2730.

[9] Fuzzy Logic: Theory and applications, prentice Hall of India, New Delhi.

[10] J. Bansiya and C.G Davis, "A Hierarchical model for object oriented Design Quality Assessment", *IEEE Transactions on software Engineering*, Vol. **28**, No. 1, (2002).

[11] C. Serban and H.F. Pop, "Software Quality Assessment using a Fuzzy Clustering approach", INFORMATICA, Nov. (2008).

[12] Y. Zhou and H. Leung "Empirical Analysis of Object Oriented design metrics for predicting High Low severity Faults", *IEEE Transactions on Software Engineering*, Vol. **32**, No. 10, October (2006).

[13] McCall, J. A., Richards, P.K., and Walters, G.F., "Factors in software quality", *Information Service*, No. vol. **1, 2 and 3**, (1997).

[14] R.A. Khan, K. Mustafa, I Ahson, "AN Empirically Validation Of Object Oriented design Quality Metrics, Journal king Saud university, *Computer and Information Science*, Vol. **19**, pp. 1-16, Riyadh (1427H/2007).

[15] Lorenz, Mark and Kidd, Jeff, object Oriented Software Metrics, Prentice Hall Publishing, (1994).

[16] N.E. Fenton, Software metrics: A rigorous Approach, Chapman and Hall, (1991).

[17] W. Li, Another metric suite for Object Oriented Programing, *Journal of System and Software*, **44**: 155-162, (1998).

[18] Basili, V.R., L.C. Briand and W.L. Melo, A Hierarchical Model for Object Oriented Design Quality Assessment", *IEEE transactions on Software engineering*, vol. **28**, no. 1, (2002). "Validation of Object Oriented Design Metrics as Quality Indicators,"*IEEE trans. Software Engineering*, vol. **22**, (1996), pp. 751-761.

[19] Krishnan, M.S., Kriebal, C.H., Kekre, S. and Mukhopadhyay, T. "An Empirically Analysis of Productivity and Quality in Software products," *Management Science*, vol. **46**, (2000), pp. 745-749.

[20] Mayer, T., and Hall, T. "Critical Analysis of Current OO Design Metrics", *Software Quality Journal*, **8** (1999), pp.97-110.

[21] Pressman, R. "A practitioners approach to Software Engineering," McGraw-Hill Publications, (2001), pp. 658-662.

[22] J. Bansiya and C.G. Davis, "A Hierarchical Model for Object Oriented Design Quality Assessment", *IEEE Transactions on Software engineering*, vol. **28**, No.1, January (2002), pp. 4-17.

[23] A. Shaik, C.R.K. Reddy, B. Mandra, Prakashini, C, Deepthi. K, "An Empirical Validation of Object Oriented Design Metrics in Object Oriented System", *Journal of Emerging Trends in Engineering and Applied Sciences*, pp. 216-224, (2010).

[24] N. kaur, M. Singh, "*Fuzzy logic Approach to Measure Precise Testability Index of Software*," vol. **3** no. 2, pp. 857-865, (2011).

[25] http://en.wikipedia.org/wiki/Quality.

[26] F.B. Abreu and R. Carapua," Candidate metrics for OOS within Taxonomy framework", *Journal of System and Software*, vol. **26**, no. 1, July (1994).