ENGINEERING JOURNAL

Article

Proposal and Validation of Usability Model for Component Based Software System

Jyoti Agarwal[1,a,*], **Sanjay Kumar Dubey**[1,b], **and Rajdev Tiwari**[2,c]

1 Amity University Uttar Pradesh, Noida, India
2 Greater Noida Institute of Engineering & Technology, Greater Noida, India
E-mail: [a]itsjyotiagarwal1@gmail.com(Corresponding author), [b]skdubey1@amity.edu, [c]rajdevtiwari@yahoo.com

Abstract. Increasing demand of rapid and cost effective development of software system has increased the demand of Component Based Software Engineering (CBSE). In CBSE, software system is developed by using existing components. These components can be in-house components or third party components. To develop a Component Based Software System (CBSS), it is important to select the suitable component in such a manner that the components of the software system do not affect each other. To increase the acceptance of the CBSS among the users and the market value of the software industries, it is important to increase the usability of the CBSS. Several usability models have been proposed for traditional and object-oriented software system (OOSS), but there is no usability model for CBSS. Existing traditional and object-oriented models can't be perfectly suitable for CBSS because of the unique characteristics of the components. This paper presents a usability model (UMCBSS) for CBSS. The proposed usability model is based on most significant usability factors. These factors are analysed from CBSS quality models. With the help of proposed model, usability is evaluated by using two different techniques i.e., centroid method and bisector method in MATLAB. Experimental results are also validated by using Center of Gravity (COG) and Mean-Max method. With the help of the proposed model, developers of the CBSS will be able to measure the usability of CBSS and to remove the usability flaws from the software system.

Keywords: Software, component, usability, model, factor, fuzzy, centroid, center of gravity, mean-max, bisector.

ENGINEERING JOURNAL Volume 21 Issue 5
Received 25 December 2016
Accepted 21 March 2017
Published 29 September 2017
Online at http://www.engj.org/
DOI:10.4186/ej.2017.21.5.243

1. Introduction

In software industry, demand is increasing for the rapid development and maintenance of cost effective software system. This is a very complex task to develop the software system from the scratch without compromising with the expected delivery time. Component Based Software Engineering (CBSE) provides solution for this problem. CBSE views a software system as a set of off-the shelf components integrated within appropriate software architecture [1]. Component Based Software System (CBSS) are developed by using existing component rather than developing the software from the scratch. CBSS have reduced the development time, cost and effort for developing the software system by using the predefined components. This perspective acts as a motivation for the software industries. Software component is defined as a unit of packaging, distribution or delivery which provides services within a data integrity or encapsulation boundary. Components can be in-house developed components or third party components. To develop a CBSS, software developers identifies which of the existing components can be reused and can be asked by the component vendors or developers. These components can be commercial-off the shelf (COTS) or open source components. Some new components are also developed by the CBSS developers and then they are added into component library for the future use [1].

Software industries are currently working on the CBSS and to increase the acceptance of the CBSS it is important the increase the quality of CBSS. Quality depends on different attributes and usability has been defined as an important attribute for the quality of CBSS. For CBSS usability can be defined as, the capability of the component to be understood, learned, used and attractive to the user, when used under specified conditions [2]. It is important to evaluate the usability of CBSS, because the success and failure of any software system depends on its usability. There are also studies for e-commerce sites that show that a 5% improvement in usability could increase revenues by 10–35% [3]. Ample research has been done on various quality attributes for CBSS but still there is lack for the research on usability of CBSS. No model has been proposed by the researchers to evaluate the usability of the CBSS.

The objective of this paper is to identify all the major factors of the usability for CBSS and to integrate them to propose a usability model suitable for CBSS. Usability is the quality attribute that is observed at run time [4], which indicates that usability is real-world issue. This makes soft computing technique ideal for estimating usability of CBSS, as soft computing deals with many uncertainties [5]. Different soft computing technique has been used by various researchers for different purposes [5, 6, 7]. In this paper, proposed model is used to measure the usability by using two different fuzzy techniques i.e., centroid and bisector method in MATLAB. Ten different input values are taken to measure the usability and to identify that which technique (centroid or bisector) is more stable. Experimental results are also validated by using Center of Gravity (COG) and Mean-Max method. The main contribution of this paper is to make developers able to evaluate the usability of CBSS, so that if any usability flaw exists in the system, it can be removed.

The structure of paper is as follows: Section 2 describes the review process. Section 3 discusses the related work. In section 4, usability model for CBSS (UMCSS) is proposed. In section 5, experimental results are shown. Section 6 deals with the validation of the experimental work. Result analysis is given in section 7 and section 8 describes the conclusion and future scope of the research work.

2. Review Process

Review of the related study is important to identify the gap in current research for future investigation and to provide the framework for the new research activities. In order to collect the evidences for usability factors of CBSS, review has been done of the secondary studies. The steps of the review process are documented below as [8, 9]:

2.1. Research Questions

After identifying the gaps in current research work following research questions is framed:

RQ1: What are the factors having great impact on usability of CBSS?

RQ2: What are the metrics for evaluating the usability factors of CBSS?

RQ3: What can be the usability model for the CBSS and how the model can be validated?

RQ4: Which technique is more suitable for measuring the usability of CBSS?

2.2. Search Process

Search is performed in electronic databases like ACM, IEEE, Springer, Google Scholar etc. by using the following string:

("software quality" OR "quality model" OR "usability of CBSS" OR "metrics of CBSS" OR "fuzzy techniques" OR "component based software system")

2.3. Inclusion and Exclusion Criteria of Research Papers

Inclusion and exclusion criteria are added to decide whether the research paper or article should be considered for current research work or not. Research papers which focus on the usability importance, usability factors for CBSS and evaluation of the usability using fuzzy technique are included. Some other papers which describe the importance of the CBSS are also referred in this paper. Research papers focusing on other quality factors like maintainability, reliability, etc. are not included in this paper as the objective of this research paper is to focus only on the usability of CBSS. Research papers not written in English are also excluded.

2.4. Quality Assessment

Research papers and articles are assessed and answered based on the following quality assessment questions [8]:

QA1: Is the search process is appropriate?
QA2: Are the inclusion and exclusion criteria is appropriate?
QA3: Did the reviewer assess the validity of the included secondary studies?
QA4: Is the basic information of included studies is provided?

2.5. Data Extraction

Following details are extracted from each study:

- Title, Source, Publication year
- Usability factors from the quality models of CBSS.
- Other useful information for the research objectives of the present paper.

3. Related Work

3.1. Usability of CBSS

Usability is defined as one of the important factor to develop efficient CBSS. For CBSS usability can be defined as the capability of the component to be understood, learned, used and attractive to the user, when used under specified conditions [10]. It is important to measure the usability of CBSS for providing fast initial response to the APIs of components, to avoid costly changes, to be able to objectively compare the usability of different components and to make usability evaluation possible for the unexperienced developers also [11].

3.2. Assessment of Usability for CBSS

To assess the usability of CBSS followings are required: a set of measurable concepts to influence usability sub-characteristics or factors, component attribute to assess these measurable concepts and direct or indirect metrics to measure component attributes [10].

In this paper firstly usability factors and metrics have been identified and then a usability model of CBSS (UMCBSS) has been proposed and validated.

3.3. Usability Factors for CBSS

After studying and analyzing research papers, it is found that usability of CBSS depends on various factors.

In this paper, these usability factors are identified and listed in Table 1 and the frequency of these usability factors are shown in Fig. 1.

Table 1. Usability factors for CBSS.

Author	Usability Factor	Source
Preiss(2001)	accessability, administrability, understandability	[4]
Bertoa(2002)	learnability, operability, understandability, complexity	[12]
Goulao(2002)	installability, learnability, configurability, operability, monitorability, compliance	[13]
Losavio(2003)	understandability, learnability, operability	[2]
Mari(2003)	learnability, efficiency, memorability, error avoidance, error handling, satisfaction	[14]
Larsson(2004)	accessibility, administrability, understandability, generality, operability, simplicity	[15]
Alvaro(2005)	understandability, configurability, learnability, operability	[16]
Rawadesh(2006)	learnability, operability, understandability, complexity	[17]
Andreou(2007)	operability, configurability, understandability, learnability,	[18]
Choi(2008)	operability, understandability, learnability,	[19]
Sharma(2008)	attractiveness, learnability, operability, understandability, complexity	[6]
Carvalho(2009)	Configurability, understandability	[20]
Kalaimangal(2010)	Learnability	[21]
Kalaimangal(2010)	Satisfaction, effectiveness	[22]
Upadhyay(2011)	Help mechanism, learnability, operability, approachability	[23]
Thapar(2014)	learnability, operability, understandability	[24]
Tiwari(2015)	installability, learnability, configurability, operability, monitorability	[1]

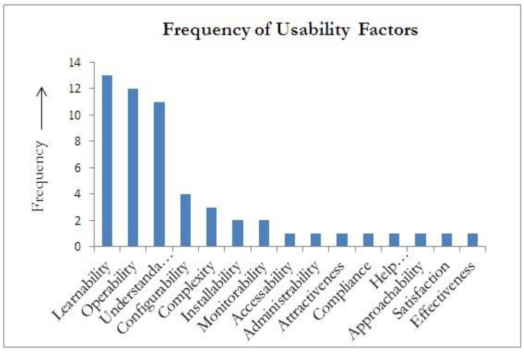

Fig. 1. Frequency of the usability factors.

3.4. Metrics for Usability Factors

Three metrics have been defined and classified to each usability factor and its attributes [16]. These metrics are defined below and Table 2 shows the classification of these three metrics.

3.4.1. Presence

This metric uses a Boolean value and a string to state whether an attribute is present in a component or not. The boolean value is used to indicates whether the attribute is present and, if so, the string describes how the attribute is implemented by the component.

3.4.2. IValues

An integer value is used to indicate exact values of the component information's and a string is used to indicate the unit (e.g. kb, mb, khtz, etc.)

3.4.3. Ratio

It is used to describe percentages. It is measured by an integer variable with values between 0 and 100.

Table 2. Attributes and metrics for usability factors of CBSS.

Usability Factors	Attributes	Metrics
Learnability	a. Time and effort to (use, configure, admin and expertise) the component	IValues
Operability	a. Complexity level	Ratio
	b. Provided Interfaces	IValues
	c. Required Interfaces	IValues
	d. Effort for operating	Presence
Understandability	a. Documentation available	Presence
	b. Documentation quality	Presence
Configurability	a. Effort for configure	Ratio

4. Proposed Usability Model for CBSS

In this paper, usability model for CBSS (UMCBSS) is proposed. To measure the usability of CBSS, usability factors are required. Table 1 shows those factors. It is identified that the average number of usability factors proposed by researchers is 3.8(≈4). So, in the proposed model four frequent usability factors are selected from Fig. 1. These four factors are learnability, operability, understandability and configurability. From these four factors, the first three factors learnability, operability and understandability are measured during life cycle of the software development and the fourth factor configurability is measured at run time of the software [16]. These for factors are integrated into a Usability model as shown in Fig. 2.

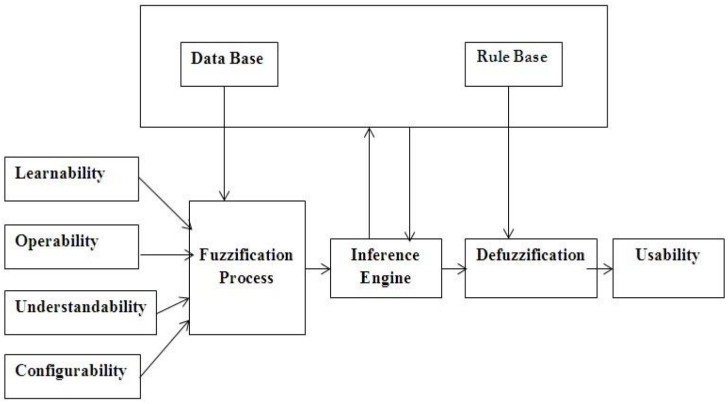

Fig. 2. Proposed usability model for CBSS.

4.1. Learnability

Learnability is defined as the capability of the component to enable the system developer to learn the application. System developer should be able to learn how to use functions and effectiveness of documentation [10]. Grossman [25] defines two type of learning i.e., formative learning and summative learning. In formative learning methodology of question asking is used to identify the learnability issues and to improve the interface. Summative learnability defines the overall usability of a system. It is used to compare the system with other competing system or to determine whether the system meets the specified requirement or not.

4.2. Operability

Operability is defined as, "the capability of the software component to enable the user (system developer) to operate and control the software component". It is important to check whether the system developers can operate the software component or not [6].

4.3. Understandability

Understandability is defined as the capability to enable the user (system developer) to understand whether the component is suitable and how it can be used for particular tasks and conditions of use [10].

4.4. Configurability

The component should be easy to configure. Minimum changes should be required to transfer a component to other environment. The developers verify the ability of component to be configured in order to determine the complexity to deploy the component into a certain context [16].

5. Experimental Work

In this research paper usability model is proposed to measure the usability of CBSS and for the evaluation of the proposed model fuzzy approach is used because it can handle with uncertain values and easy to implement. Mamdani fuzzy inference system is used as it is widely accepted for capturing the expert knowledge. This method is more human like manner. In Mamdani FIS, membership function is used for all the inputs and output [26]. Following steps have to be followed to conduct the experiment:

a. Identify input and output of the model: Four frequent usability factors learnability, operability, understandability and configurability are selected as input and usability is the output for the proposed usability model (UMCBSS).
b. Define membership function low, medium, high for all the inputs and very_low, low, medium, high, very_high for output. Membership function for learnability is shown in Fig.3. Similarly membership functions for other input factors are also created. Fig. 4 shows the membership function for the output factor usability.
c. Create a rule base:
   (i) Survey based on the selected usability factors: The main purpose for conducting the survey was to gather information about the fuzzy nature of input and output parameters. In questionnaire, it was mentioned that learnability, operability, understandability and configurability were independent variables and usability is dependent variable. The details about the input factors and their importance were also clearly mentioned with respect to usability. This was a close ended survey questionnaire and was distributed to 76 persons (42 were IT experts and 34 were academia) which were doing projects or research related to usability. Total 47 responses were obtained among which 26 responses were from IT experts and 21 from the academia. Finally 31 responses were selected (17 responses from IT experts and 14 from the academia). Rest of the responses were discarded due to the incomplete data.
   (ii) Based on the selected survey data rule base is created. Total 81 rules are created using different combinations of input for the four factors used for measuring usability.
d. Feed the created rule base into fuzzy inference engine.
e. Fuzzify the inputs.
f. Defuzzify the output.

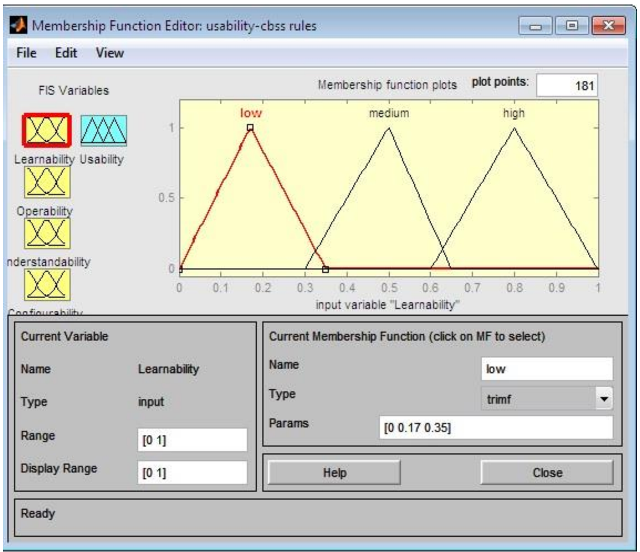

Fig. 3. Membership function for learnability.

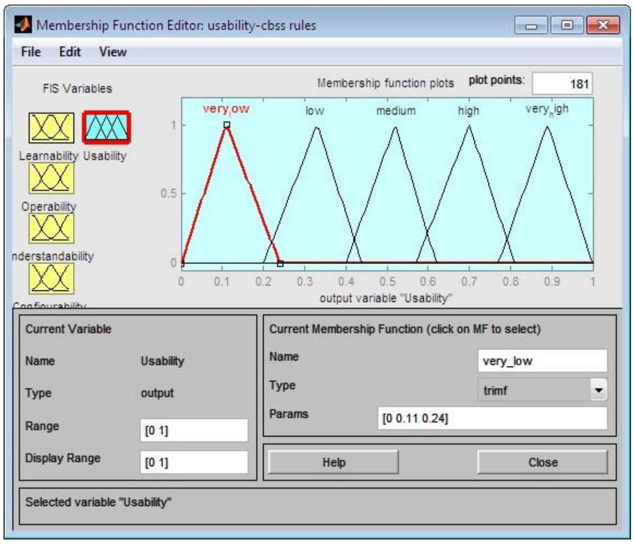

Fig. 4. Membership function for usability.

5.1. Rule Base for Proposed Usability Model

Total 81 rules are created by taking all the possible values of the input values (low, medium, high) and usability is classified as very_low, low, medium, high and very_high. These rules are fed into fuzzy inference engine. Some of the rules are:

Rule 1: (if learnability is high) and (if operability is high) and (if understandability is high) and (if configurability is high) then usability is very_high.

Rule 13: (if learnability is medium) and (if operability is high) and (if understandability is medium) and (if configurability is high) then usability is high.

Rule 24: (if learnability is low) and (if operability is medium) and (if understandability is high) and (if configurability is high) then usability is medium.

Rule 75: (if learnability is low) and (if operability is medium) and (if understandability is low) and (if configurability is medium) then usability is low.

Rule 80: (if learnability is low) and (if operability is low) and (if understandability is medium) and (if configurability is low) then usability is very_low.

To find the value of usability input values are given to the four usability factors. For this experimental work 10 input values are used. These input values are shown in the input values are given in Table 3.

Table 3. Input values for usability factors.

Input	Learnability	Operability	Understandability	Configurability
1	0.25	0.35	0.4	0.3
2	0.35	0.25	0.25	0.31
3	0.29	0.3	0.35	0.35
4	0.3	0.4	0.35	0.31
5	0.2	0.3	0.35	0.35
6	0.35	0.25	0.35	0.31
7	0.35	0.3	0.2	0.25
8	0.32	0.38	0.33	0.28
9	0.25	0.33	0.39	0.3
10	0.36	0.28	0.3	0.29

When input 1 is applied then the value of usability is 0.283 using centroid method and 0.3 using bisector method. 0.3 which is shown in Fig. 5 and Fig. 6 respectively.

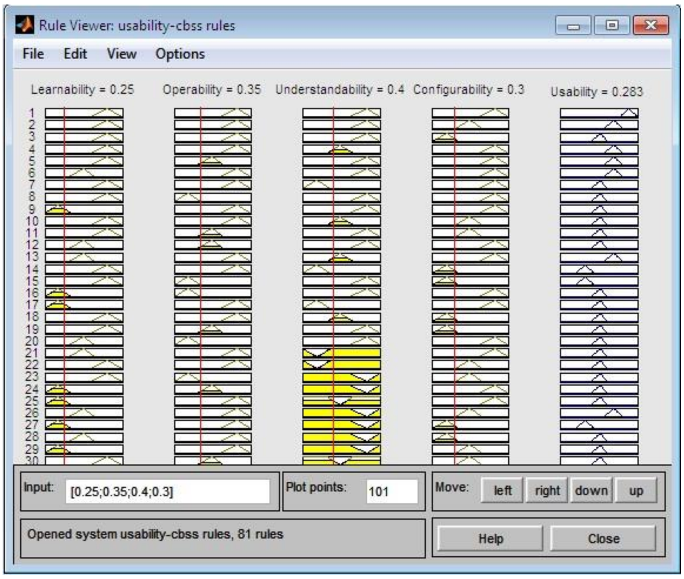

Fig. 5. Rule viewer for proposed usability model using centroid method.

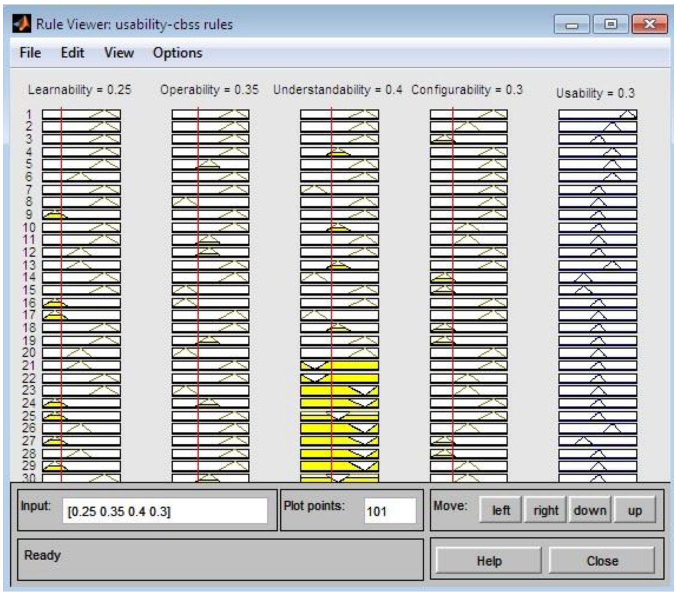

Fig. 6. Rule viewer for proposed usability model using bisector method.

Similarly usability is measured for all another input values. Table 4 shows the usability value obtained for ten different input values from centroid and bisector method. Co-efficient of variance (CV) is also calculated to identify that which technique is more stable.

Table 4. Usability for 10 different input values.

Input	Usability	
	Centroid Method(x)	Bisector Method(y)
1	0.283	0.30
2	0.220	0.22
3	0.243	0.26
4	0.245	0.26
5	0.243	0.26
6	0.257	0.28
7	0.220	0.22
8	0.247	0.21
9	0.215	0.21
10	0.220	0.22
Mean	0.2391	0.244
Standard Deviation(S.D)	0.0211	0.032
CV(S.D/mean)	0.088	0.131

Table 4. shows that the CV(x)<CV(y), which means centroid method is more stable than bisector method, which means the result obtained from centroid method are more consistent.

6. Validation of Results

To validate the result of experimental work two methods i.e. Center of Gravity (COG) and Mean-Max methods are used. Center of Gravity (COG) is most prevalent and widely used defuzzification method [27] and Mean-max method gives the most plausible or acceptable results [28].

6.1. Center of Gravity (COG) Method for Validation of the Result of Centroid Method (Obtained from MATLAB

6.1.1. Membership values for input values

The procedure of finding membership value for the given input value is shown in Fig. 7. In this figure membership value for learnability is 0.55 for the given input 0.25. Similarly membership value is obtained for the input values of operability, understandability and configurability. The membership values for all the four input values are shown in Table 5.

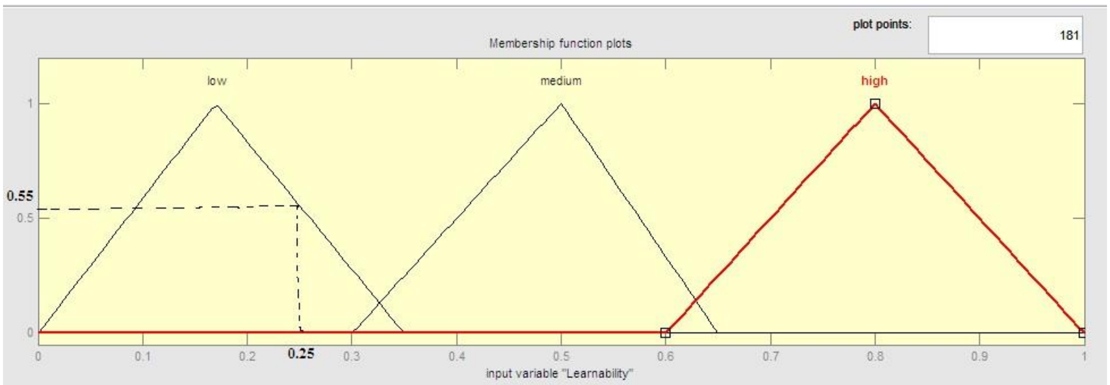

Fig. 7. Membership value of learnability.

Table 5. Membership values for input values for usability factors.

Input Factors	Learnability	Operability	Understandability	Configurability
Input Value	0.25	0.35	0.4	0.3
Membership Value	Low=0.55	Low=0.05 Medium=0.25	Medium=0.45	Low=0.35

6.1.2. Output computation of usability using COG

From Table 5, we get membership values as {0.55, 0.05, 0.45, 0.35} and {0.55, 0.25, 0.45, 0.35}. To create computation graph for usability two valued are obtained by taking minimum values of the membership values.

min{0.55, 0.05, 0.45, 0.35}=0.05

min{0.55,0.25,0.45,0.35}=0.25

By using these two values computation graph for usability is created in Fig. 8.

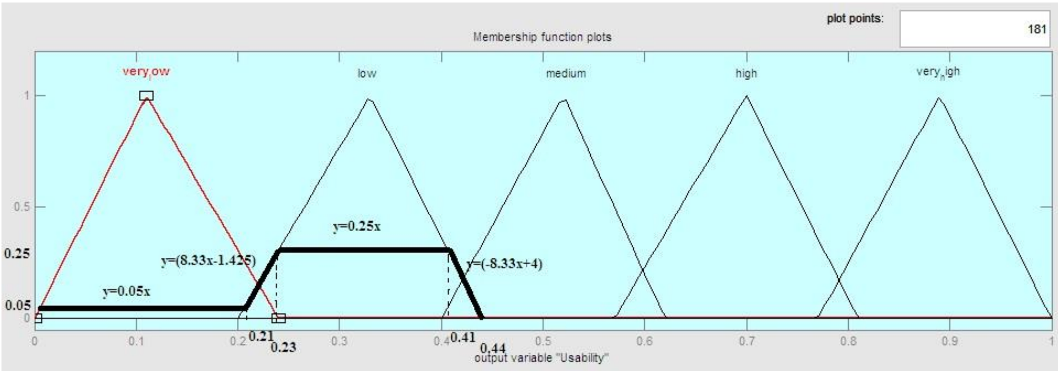

Fig. 8. Output computation of usability.

Defuzzification of the above output is obtained by finding the Center of Gravity of the above fuzzy output.

$$\text{Usability}=\frac{\int_{0}^{0.21} 0.05\,x\,dx+\int_{0.21}^{0.23}(8.33x-1.425)x\,dx+\int_{0.23}^{0.41}0.25\,x\,dx+\int_{0.41}^{0.44}(-8.33\,x+4)x\,dx}{\int_{0}^{0.21} 0.05\ dx+\int_{0.21}^{0.23}(8.33x-1.425)dx+\int_{0.23}^{0.41}0.25\ dx+\int_{0.41}^{0.44}(-8.33\,x+4)\,dx}$$

=0.30

6.2. Mean of Maximum(MOM) Method for validation of the result of Bisector Method (obtained from MATLAB

z*=(a+b)/2

Here a=upper limit of maxima and b= lower limit of maxima

Usability= (0.41+0.23)/2

=0.32

7. Result Analysis

The value of usability calculated for input 1 by using the COG formula is 0.3 which is very close to the value of usability (0.283) obtained from MATLAB experiment. The difference between both values is 0.017 which is less than standard significant error (α=0.05). Similarly the value of usability calculated by using mean-max method is 0.32 which is approximately equal to the value obtained by bi-sector method (0.3) in MATALB. The difference between both values is 0.02 which is also less than standard significant error. It proves that the proposed usability Model for CBSS (UMCBSS) model is correct and valid. It is also find out that co-efficient of variance (CV) for centroid method is less than CV of bisector method, which means centroid method is more stable.

8. Conclusion and Future Scope

The demand for CBSS is increasing rapidly. It is a challenge for software developers to develop usable software to reduce the failure rate and to increase the acceptability of the software systems. So there is need to develop a usability model. This paper proposes and validates a usability model for CBSS (UMCBSS). The results are obtained by centroid and bisector method using MATLAB R2013a software. Experimental results are also validated by using Center of Gravity (COG) and Mean-Max method. This paper also identifies that the centroid method is more stable for finding usability of CBSS. The proposed model will be helpful for the software developers to measure and compare the usability of software system, which will help to increase the acceptability and popularity of the software. In future, more efficient model can also be developed by using more efficient techniques.

References

[1] A. Tiwari and P. S. Chakraborty, "software component quality characteristics model for component based software engineering," in *IEEE International Conference on Computational Intelligence and Communication Technology*, pp. 47-51, 2015.

[2] F. Losavio, L. Chirinos, and N. Lévy, "Quality characteristics for software architecture," *Journal of Object Technology*, vol. 2, no. 2, pp. 133-150, Mar. 2003.

[3] N. Juristo, A. M. Moreno, and M. I. Sanchez-Segura, "Analysing the impact of usability on software design," *Journal of Systems and Software*, vol. 80, no. 9, pp. 1506-1516, Sep. 2007.

[4] O. Preiss, A. Wegmann, and J. Wong, "On quality attribute based software engineering," in *27th Euromicro Conference, IEEE*, 2001, pp. 114-120.

[5] K. Tyagi and A. Sharma, "An adaptive neuro fuzzy model for estimating the reliability of component-based software systems," *Applied Computing and Informatics*, vol. 10, no. 1, pp. 38-51, Jan. 2014.

[6] A. Sharma, R. Kumar, and P. S. Grover, "Estimation of quality for software components: An empirical approach," *ACM SIGSOFT Software Engineering Notes*, vol. 33, no. 6, pp. 1-10, 2008.

[7] W. Daosud, K. Jariyaboon, P. Kittisupakorn, and M. A. Hussain, "Neural network based model predicting control of batch extractive distillation process for improving purity of acetone," *Engineering Journal*, vol. 20, no. 1, pp. 47-59, 2016.

[8] B. Kitchenham, "Procedure for undertaking systematic reviews," Computer Science Department, Keele University (TRISE-0401) and National ICT Australia Ltd (0400011T. 1), Joint Technical Report, 2004.

[9] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering—A systematic literature review," *Information and Software Technology*, vol. 51, no.1, pp.7-15, 2009.

[10] M. F. Bertoa and A. Vallecillo, "Usability metrics for software components," in *8th International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'2004)*, Oslo, Norway, June 2004.

[11] T. Scheller and E. Kuhn, "Measurable concepts for the usability of software components," in *37th EUROMICRO Conference on Software Engineering and Advanced Applications*, pp. 129-133, Aug. 2011.

[12] M. Bertoa and A. Vallecillo, "Quality attributes for COTS components," in *6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, 2002.

[13] M. Goulao and F. B. Abreu, "Towards a components quality model," in *Work in Progress Session of the 28th Euromicro Conference (Euromicro)*, Dortmund, Germany, Sep. 2002.

[14] M. Mari and N. Eila, "The impact of maintainability on component-based software systems," in *29th Euromicro Conference,* IEEE, pp. 25-32, Sep. 2003.

[15] M. Larsson, "Predicting quality attributes in component-based software systems," Mälardalen University, Mar. 2004.

[16] A. Alvaro, E. S. Almeida. and S. L. Meira, "Quality attributes for a component quality model," *in 10th WCOP/19th ECCOP,* Glasgow, Scotland, pp. 31-37, 2005.

[17] A. Rawashdeh and B. Matalkah, "A new software quality model for evaluating COTS components," *Journal of Computer Science*, vol. 2, no. 4, pp. 373-381, 2006.

[18] A. S. Andreou and M. Tziakouris, "A quality framework for developing and evaluating original software components," *Information and Software Technology*, vol. 49, no. 2, pp. 122-141, 2007.

[19] Y. Choi, S. Lee, H. Song , J. Park, and S. Kim, "Practical S/W component quality evaluation model," in *IEEE 10th International Conference on Advanced Communication Technology (ICACT)*, vol. 1, pp. 259-264, Feb. 2008.

[20] F. Carvalho, S. R. L. Meira, B. Freitas, and J. Eulino, "Embedded software component quality and certification," in *35th Euromicro Conference on Software Engineering and Advanced Applications,* IEEE, pp. 420-427, Aug. 2009.

[21] S. Kalaimagal and R. Srinivasan, "Q'Facto 10-A commercial off-the-shelf component quality model proposal," *Journal of Software Engineering*, vol. 4, no. 1, pp. 1-15, 2010.

[22] S. Kalaimagal and R. Srinivasan, "Q'Facto 12: An improved quality model for COTS components," *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 2, pp. 1-4, Mar. 2010.

[23] N. Upadhyay, B. M. Despande, and V. P Agarwal, "Towards a software component quality model," in *Advances in Computer Science and Information Technology*, Springer, Berlin Heidelberg, Jan. 2011, pp. 398-412.

[24] S. S. Thapar, P. Singh, and S. Rani, "Reusability based quality framework for software components," *ACM SIGSOFT Engineering Notes*, vol. 39, no. 2, pp. 1-5, Mar. 2014.

[25] T. Grossman, G. Fitzmaurice, and and R. Attar, "A survey of software learnability: Metrics, methodologies and guidelines," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, pp. 649-658, Apr. 2009.

[26] A. Kaur and A. Kaur, "Comparison of Mamdani-type and Sugeno-type fuzzy inference systems for air conditioning system," *International Journal of Soft Computing and Engineering (IJSCE*), vol. 2, no. 2, pp. 323-325, May 2012.

[27] [Online]. Available: http://www.csee.wvu.edu/classes/cpe521/presentations/DEFUZZ.pdf [Accessed: 18 Jan. 2017].

[28] [Online]. Available: http://zone.ni.com/reference/en-XX/help/370401H-01/lvpidmain/io_meanmax/[Accessed: 18 Jan. 2017].