

Research Article

Empirical Evaluation of Fuzzy Synthetic Based Framework for Multifaceted Component Classification and Selection

¹Vinay, ¹Manoj Kumar, ¹Prashant Johri and ²Rajesh Kumar

¹School of Computing Science and Engineering, Galgotias University, Greater Noida (U.P.), India

²School of Mathematics and Computer Applications, Thapar University, Patiala (Punjab), India

Abstract: Component Based Software Engineering (CBSE) provides an approach to develop high quality software system at less cost by using fresh and existing software components. The quality of the software system is based on the quality of individual software component integrated. Application developer wants the good or the fittest component to assemble and improve the quality of the software product. The application developer specifies the criteria and requirements of software systems and uses them in selecting the fit components. Component classification and selection is a practical problem and requires complete and predictable input information. It is missing due to uncertainty in judgment and impression in calculations. Hence, component fitness evaluation, classification and selection are critical, multi-faceted, fuzzy and vague nature problems. There exists many component selection approaches, but these lack the repeatable, usable, exile, multi-faceted and automated processes for component selection and filtration. These approaches are not fulfilling the objectives of software industry in terms of cost, quality and precision. So, there is need of hour to devise an intelligent approach for multifaceted component fitness evaluation, classification and selection. In this study, fuzzy synthetic based approach is proposed for multi-criteria fitness evaluation, classification and selection of software component. For validation of the proposed framework, fifteen black box components of calculators are used. It helps the application developer in selecting fit or high quality component. The proposed framework reduces the cost and enhances the quality, productivity of software systems.

Keywords: Component classification, component selection, fuzzy synthetic evaluation approach, software component, software quality

INTRODUCTION

Component Based Software Engineering (CBSE) incorporates assembly of different pre-packaged software units to form a big software system. Component plays an individual role within the system software to fulfill clients' requirements. Bachman *et al.* (2000) explained that the component is an independent unit and provides a specified functionality. There is a challenge to select appropriate component from available components. Xia *et al.* (2000) pointed out that Component Based Software Development (CBSD) provides the benefit of high quality, timely delivery and cost efficient way of software development. It is important to concentrate over the selection of individual component with high quality.

Today, components are available in the open market from different vendors. To select appropriate component, decision makers cannot rely only on a single parameter of component instead of this they need to focus on various selection parameters like functionality, reliability, maintainability etc. These

parameters collectively decide the overall quality of the software. Software quality is the most important factor for success of the software in market. Nasib and Grover (2004) explained that Components are black-box in nature hence; there is no access to the internal structure of the components.

To evaluate the quality of component, decision makers consider observable properties of components. The contribution of this study is to propose a framework for software component classification, filtration and selection using Fuzzy Synthetic evaluation system. This framework is termed as multi-faceted approach as it includes various quality factors and sub-factors. Component selection requires the evaluation of quality or fitness for software components. Anil *et al.* (2009) explained that almost all researchers evaluate and select the component on the basis of internal metric but the software components are black-box and only few metrics are available to evaluate the quality of black-box components. MacCormack *et al.* (2007) shows that available component selection approaches gave importance to

Corresponding Author: Manoj Kumar, School of Computing Science and Engineering, Galgotias University, Greater Noida (U.P.), India

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

functional and non-functional requirements but it is important to understand that these requirements should be fulfilled by only good quality software components.

All quality factors don't have equal importance as the importance of quality factors changed from domain to domain. It leads another challenge to estimate the weight values of different quality factors and sub-factors according to the specified domain. In practical problems due to several conflicting objectives and uncertain statements, it is difficult to draw decisions out of it. Fuzzy logic provides a way for decision makers to draw conclusions from imprecise information. Our proposed multifaceted measurement framework is used for component classification and filtration. The proposed framework will provide a solution to reduce cost, efforts and time to select software components.

During literature review, research paper of Manoj *et al.* (2012) and Jianli and Ningguo (2007) on Fuzzy Synthetic based test case selection and software quality improvements respectively, gave us a lead to adopt Fuzzy Synthetic approach for software component classification and selection.

MULTIFACETED MEASUREMENT FRAMEWORK FOR COMPONENT SELECTION

In the process of component classification, filtration and selection, it is important to estimate the numeric values of classification criteria. Fuzzy-multifaceted approach for component classification and selection is a new approach. It helps stakeholders to take decisions for filtration and classification. It requires an appropriate scheme for weight value calculation. It is realized that not all quality factors and sub-factors have equal importance, so it requires prioritizing these factors and sub-factors (Table 1).

Weight values calculation: To establish the facts, we conducted a survey on software professionals who have rich experience in banking domain and e-Commerce software development. The survey form consists of different comparative questions and professionals are requested to assign values using Satty Scale. Different responses were collected and analysis was done using Satty Scale. Weight values that comes out of this process help decision makers to classify and select components. Thomas (2008) proposed a scale which is used to take decisions where problems are difficult to quantify the intangible factors. We designed survey form to quantify weight values using bottom to top approach. Different quality sub-factors are compared with each other to conclude their importance or weight values and in turn using these values we will conclude the weight values of major five factors. MS EXCEL is used to analyze and process generated data. The sum of all weight values is 1. We use the evaluated weight

Table 1: Quality factors and weight values

First layer index/factors	Second layer index
Functionality X1 (0.44)	Suitability X11 (0.163)
	Accuracy X12 (0.286)
	Self-defined X13 (0.118)
	Reusability X14 (0.43)
Reliability X2 (0.06)	Maturity X21 (1)
	Understandability X31 (0.677)
Usability X3 (0.236)	Learn ability X32 (0.322)
	Resource behavior X41 (1)
Efficiency X4 (0.11)	Changeability X51 (1)
Maintainability X5 (0.15)	

values in table to develop the membership degree for components in ascertaining sets.

MULTILAYER FUZZY SYNTHETIC VALUATION APPROACH

Cai *et al.* (1998) gave an approach which uses an index system for component classification. Different steps of this algorithm are as follows:

Ascertain valuation set: $Y = \{\text{Good, Average, Bad}\}$.

Ascertain evaluation factor: Evaluation factors are divided into two index layers i.e., first index layer and second index layer. Evaluation factor set of first layer consists of major quality factors denoted by $\{X1, X2, X3, X4, X5\}$ similarly evaluation factor set of the second layer consists of corresponding quality sub-factors. For example evaluation factor set of X1 is $\{X11, X12, X13, X14\}$.

Ascertain the weight of each index layer: Weight values of each index layer are calculated using Satty Scale and discussed in Table 1.

Ascertain fuzzy evaluation matrix: It is required to ascertain the fuzzy evaluation matrix and it is related to two index layer factors of the software component evaluation system. Fuzzy evaluation matrix values are membership degree values of each factor which is correlative with each index researched and developed by USA RADC. These values are given under the boundary interval of $[0, 1]$, here "0" signify worst and "1" signify excellence. Arun *et al.* (2008) gave different metrics which are used here to calculate numeric values, Here for limiting the length, we just take $X11 = 1 - (\text{No. Of Operations not suitable} / \text{total number of operations provided})$, $X12 = \text{Number of operations having required accuracy} / \text{Total number of operations}$, $X13 = \text{Functions provided by the component itself without external support} / \text{Total function provided by the component}$, $X14 = \text{Number of Customizable properties} / \text{Total Number of properties}$. The evaluation of the second layer index refers to component classification. Generally, membership degree values for second layer index factors are distributed under

Table 2: Value interval for second class index/factors

Second layer index/factors	Good	Average	Bad
X11	0.70, 1	0.40, 0.70	0, 0.40
X12	0.72, 1	0.45, 0.75	0, 0.45
X13	0.75, 1	0.45, 0.75	0, 0.45
X14	0.80, 1	0.50, 0.80	0, 0.50
X21	0.77, 1	0.48, 0.77	0, 0.48
X31	0.74, 1	0.49, 0.74	0, 0.49
X32	0.73, 1	0.47, 0.73	0, 0.47
X41	0.75, 1	0.50, 0.75	0, 0.50
X51	0.76, 1	0.40, 0.76	0, 0.40

Table 3: Value constant m and c for second class index/factors

Second layer index/factors	(m, c)		
	Good	Average	Bad
X11	(0.850, 0.180)	(0.550, 0.180)	(0.200, 0.240)
X12	(0.860, 0.168)	(0.585, 0.162)	(0.225, 0.270)
X13	(0.875, 0.150)	(0.600, 0.180)	(0.225, 0.270)
X14	(0.900, 0.120)	(0.650, 0.180)	(0.250, 0.300)
X21	(0.885, 0.138)	(0.625, 0.174)	(0.240, 0.289)
X31	(0.870, 0.156)	(0.615, 0.150)	(0.245, 0.295)
X32	(0.865, 0.160)	(0.600, 0.156)	(0.235, 0.283)
X41	(0.875, 0.150)	(0.625, 0.150)	(0.250, 0.300)
X51	(0.880, 0.144)	(0.580, 0.210)	(0.200, 0.240)

Table 4: Membership degree for second class index/factors

Second layer index/factors	μ (x)		
	Good	Average	Bad
X11 = 0.90	0.93	0.02	0
X12 = 0.80	0.88	0.17	0.01
X13 = 0.77	0.61	0.41	0.01
X14 = 0.88	0.98	0.20	0.01
X21 = 0.25	0.00	0.01	0.99
X31 = 0.66	0.16	0.92	0.13
X32 = 0.77	0.71	0.31	0.03
X41 = 0.94	0.83	0.01	0
X51 = 0.88	0.99	0.14	0

different intervals shown in Table 1. The interval values considered in this study for second class index are given in Table 2.

Computation of membership function for valuation set is done by using Eq. (1):

$$\mu (x) = e^{-\left(\frac{x-m}{c}\right)^2} \tag{1}$$

Jianli and Ningguo (2007) explained that “m” and “c” are constant values for Eq. (1), as x = m, the value of μ (x) reaches to the maximum level that is μ (x) = 1. For ascertain set “excellent” interval is [0.85, 1] and m = (0.85 + 1) / 2 = 0.925.

Manoj *et al.* (2012) shows that “x” is a boundary point for two neighbor intervals and therefore membership degree values are same for corresponding two successive comments. So, μ (x = boundary point) = 0.5. The constant “c” is computed by using Eq. (2):

$$c = \frac{X_r - X_l}{1.66} \tag{2}$$

Here X_r and X_l are the right endpoint of the interval are given in Table 2. All “m” and “c” values are given in Table 3.

Fuzzy membership degree values are computed for second layer index factors by taking data of Table 1 and Eq. (1) for corresponding comments and in different intervals.

We are maintaining fuzzy membership degree values only for second layer index items as per their corresponding interval and comment sets in Table 4. The same method is used to calculate all second layer index factors. To derive the results we must extract the desired information from the component, this information is basically observable properties of a component. Different observable properties and information are extracted by using the trial version of the component available from the internet or some other component repositories.

To use fuzzy synthetic evaluation algorithm, we first compute fuzzy evaluation matrix for second layer index factors and then we move toward first layer index factors. It provides a relationship using RADC measurement algorithm. The Fuzzy evaluation matrix for the first layer index factors can be calculated as:

$$R_n = A_n * R_{nm} \tag{3}$$

In the same order fuzzy evaluation matrix is obtained for first layer index factors. At the end, in turn we get the fuzzy synthetic evaluation factor. As per example, we gave membership degree values for X11, X12, X13, X14 and these values are given in Table 4. We will get fuzzy evaluation matrix for second layer index factors with same method:

$$R_{11} = \begin{pmatrix} 0.93 & 0.02 & 0 \\ 0.88 & 0.17 & 0.01 \\ 0.61 & 0.41 & 0.01 \\ 0.98 & 0.20 & 0.01 \end{pmatrix}$$

Matrix R₁₁ is used to calculate fuzzy evaluation matrix for first layer Index factors by using Eq. (3). A₁, A₂, A₃, A₄ and A₅ are the weight values of the second layer index factors that are A₁= (0.16, 0.286, 0.118, 0.43), A₂= (1), A₃ = (0.677, 0.32), A₄ = (1) and A₅= (1):

$$R_1 = (0.16, 0.286, 0.118, 0.43) * \begin{pmatrix} 0.93 & 0.02 & 0 \\ 0.88 & 0.17 & 0.01 \\ 0.61 & 0.14 & 0.01 \\ 0.98 & 0.20 & 0.01 \end{pmatrix} = (0.90, 0.18, 0)$$

Values for R₂, R₃, R₄ and R₅ are calculated in the same way as explained above. These values form

Table 5: Membership degree of calculators

Component name	Good	Average	Bad
Calculator 7	0.71	0.27	0.083
Calculator pro	0.30	0.33	0.380
1 st Calc	0.28	0.13	0.540
Microsoft Cal 2	0.60	0.33	0.150
Calc ME 4.1	0.25	0.59	0.180
Calc with paper roll	0.18	0.64	0.240
Smart maths	0.44	0.48	0.200
Complex calculator	0.26	0.68	0.150
Calc 1 tape	0.23	0.74	0.130
Desktop Calc	0.24	0.33	0.490
Multipurpose Calc	0.27	0.21	0.430
Add'em up Calc	0.36	0.60	0.100
See and Calc	0.26	0.51	0.250
ESB Calc pro	0.30	0.63	0.070
B&G Calc	0.27	0.16	0.510

another and matrix will be multiplied with the weight values of first layer index factors:

$$[0.44, 0.06, 0.236, 0.11, 0.15]*$$

$$\begin{pmatrix} 0.90 & 0.18 & 0 \\ 0 & 0.01 & 0.99 \\ 0.33 & 0.72 & 0.10 \\ 0.83 & 0.01 & 0 \\ 0.99 & 0.14 & 0 \end{pmatrix}$$

Fuzzy Synthetic Evaluation Value of the evaluated component = [0.71, 0.27, 0.083]. Jianli and Ningguo (2007) proposed a computation method, by this computation method, the fuzzy evaluation matrix for first layer index factors comes out. Finally, we get the fuzzy synthetic evaluation value of the evaluated software component. We adjust fuzzy synthetic evaluation value and distribute it in the intervals [0, 100]. We divide the interval [0, 100] in five sub intervals corresponds to five comment sets. In the same way we compute the degree of membership for 15 different calculators in Good, Average and Bad comment set (Table 5).

Empirical study:

Software components used: To conduct the empirical study, we select scientific calculators as a component. These calculators are available freely and with restricted license terms and conditions. For our study and implementation, we use 15 different calculators which are available for evaluation under restricted license terms.

Experimental setup: We perform evaluations of our proposed methodology for software component selection and classification on 15 different software components. Metric values are calculated for all the components. The validation of the proposed framework is done using C++ programming language on windows 7, Home Basic, Service Pack 1 with Intel (R Core TM) -i3-2310M CPU @ 2.10 GHz Pentium, 4.0 GB RAM, 64 bit operating systems.

Criteria used: Major quality factors and sub-factors for component classification and selection are as follows:

Functionality: Suitability, Self-defined, Accuracy and Reusability

Reliability: Maturity

Efficiency: Resource Behavior

Usability: Understandability and Learnability

Maintainability: Changeability

RESULTS AND DISCUSSION

We validated proposed framework to derive a membership degree belonging to three different classes for 15 calculators. We select all the calculators from open sources and web repositories (CNET/Softpedia).

We conducted an evaluation of all the components to extract information regarding properties, functions and operational support. Membership degrees of all the components are computed using Fuzzy Synthetic Evaluation System. In the present study, software components with degree of membership 0.48 to 0.55 are considered as “ambiguous” components. Components with a membership degree of 0.60 or more from “Good” category are liable for selection and the remaining components are “filtered-out”. The proposed framework categorizes components in three categories given in Table 6 as follows:

Table 6 shows the final classification of calculator components in three different categories. Only two calculators are came under “Good” category, six calculators are rated “Average” and seven calculators belongs to “Bad” category. Final selection of components is carried out as follows:

- **“Selected” calculators:** Calculator 7, Microsoft Cal 2
- **“Filtered out” calculators:** Calc ME 4.1, Calc with paper roll, Complex Calculator, Calc tape 1, Add'em up Calc, ESB Calc Pro Calculator Pro, 1st Calc, Desktop Calc, Multipurpose Calc and B and G Calc
- **“Ambiguous” calculators:** Smart Math, See and Calc

Results and observation shows that above mentioned classification is flexible and user defined and can be modified as per the requirements of stakeholders. Table 7 shows that our proposed framework selects 2 components and filtered-out 11 components out of 15 components. Out of 15 components 2 components show ambiguity. Matrix based computations of this framework save a lot of

Table 6: Category based classification of calculators

Category	Components
Good	Calculator 7 and microsoft Cal 2
Average	Calc ME 4.1, Calc with paper roll, complex calculator, Calc tape 1, Add' emup Calc and ESB Calc Pro
Bad	Calculator Pro, 1st Calc, smart maths, desktop Calc, multipurpose Calc, See & Calc and B&G Calc

Table 7: Distribution of components

Total components	Selected components	Filtered-out components	Ambiguous components
15	2	11	2

computational cost and efforts over manual arithmetic computations. It clearly shows that this framework reduces the software development efforts.

CONCLUSION

Software component selection is a challenging task for researchers and academicians from a long time. The adaptability and flexibility of our framework help in the classification of software components and help decision makers to select the appropriate component as per their requirements. Fuzzy computing based Synthetic evaluation approach is well suited for software component classification and evaluation problem. It improves practicability and flexibility in software component fitness evaluation and classification. The proposed framework reduces the efforts and computational cost for component fitness evaluation. This framework provides fitness evaluation of software components on the basis of different quality factors and sub-factors. While using this framework, software industry can reduce their development cost, time and also increases the quality of the final software product.

In future, K-mean clustering and other classification and selection schemes can be used to classify different components. In the current study, components with a membership degree of 0.48 to 0.55 are introduced as highly ambiguous components and fitness of these components can be improved by using different evolutionary approaches. The Conversion process will be put forward in the future.

REFERENCES

- Anil, S., R. Jadhav and M. Sonar, 2009. Evaluating and selecting software packages: A review. *Inform. Softw. Technol.*, 51(3): 555-563.
- Arun, S., K. Rajesh and P.S. Grover, 2008. Estimation of quality for software components: An empirical approach. *SIGSOFT Softw. Eng. Notes*, 33(6): 1-10.
- Bachman, F., L. Bass, C. Buhman, S.C. Dorda, F. Long, J. Robert, R. Seacord and K. Wallnau, 2000. Volume II: Technical Concept on Component based Software Engineering. CMU/SEI-2000-TR-008 ESC-TR-2000-007.
- Cai, W., 1998. Introduction of extenics. *Syst. Eng. Theory Pract.*, 18(1): 76-84.
- Jianli, D. and S. Ningguo, 2007. Research and improvement of the fuzzy synthesis evaluation algorithm based on software quality. *Comp. Eng. Sci.*, 29(1): 66-69.
- MacCormack, A., R. John and B. Carliss, 2007. The Impact of Component Modularity on Design Evolution: Evidence from the Software Industry. Harvard Business School Technology and Operations Mgt. Unit Research Paper 08-038.
- Manoj, K., A. Sharma and R. Kumar, 2012. Multi-faceted measurement framework for test case classification and fitness evaluation using fuzzy logic based approach. *Chiang Mai J. Sci.*, 39(3): 486-497.
- Nasib, S.G. and P.S. Grover, 2004. Few important considerations for deriving interface complexity metric for component-based systems. *SIGSOFT Softw. Eng. Notes*, 29(2): 4.
- Thomas, L.S., 2008. Decision making with the analytic hierarchy process. *Int. J. Serv. Sci.*, 1(1): 83-98.
- Xia, C., M.R. Lyu, W. Kam-Fai and K. Roy, 2000. Component-based software engineering: Technologies, development frameworks and quality assurance schemes. *Proceeding of the 7th Asia-Pacific Software Engineering Conference (APSEC 2000)*, pp: 372-379.