



## A Fuzzy Analytical Hierarchy Process (FAHP) Based Software Quality Assessment Model: Maintainability Analysis

Udaya Dayanandan<sup>1\*</sup>      Vivekanandan Kalimuthu<sup>1</sup>

<sup>1</sup>*Pondicherry Engineering College, India*

\* Corresponding author's Email: [udaya\\_anand@yahoo.com](mailto:udaya_anand@yahoo.com)

**Abstract:** Software quality assessment at the architectural level is efficient because it provides various advantages to find and fix design errors at the earliest stage of software development life cycle (SDLC). Numerous methods have been evolved to validate quality-related issues at the architectural level. The software quality comprises of a total of eight attributes where maintainability is one of the important attribute which affects the overall system quality. Though few methods are available for maintainability analysis, none of the state of art method employs fuzzy logic with Quality Model for Object Oriented Design (QMOOD) metric suite. In this paper, the fuzzy analytic hierarchy process (FAHP) method is proposed with Buckley method to evaluate the influence of maintainability and its sub-attributes in software quality at the architectural level. The proposed evaluation model is tested against several versions of the MFC and OWL windows application frameworks and the obtained results are compared with the state-of-the-art methods. This experimental analysis shows that the FAHP with Buckley method is superior to the other compared methods.

**Keywords:** Software architecture, Maintainability, Fuzzy logic, Buckley method, Multi-criteria decision making.

### 1. Introduction

The field of software quality assessment has been gaining importance among researchers and developers more recently. Several methods are available for object-oriented (OO) software analysis which can be applied when the product is complete or almost complete [1]. However, it is ineffective and too late to enhance the internal characteristics of the software closer to implementation. Therefore, there is a need to develop an evaluation model, which is employed at the architectural level to verify whether the design has the required internal properties, which can result in the development of better quality software. This will provide a chance for software developers to rectify issues, prevent non-conformance to standards and avoid unnecessary complexities, even prior to implementation. This will also help to reduce rework before and after implementation and eliminate the wastage of resources. ISO defines that the quality of software system comprises of eight

different attributes as shown in Fig. 1 [2]. Every individual attribute can be broken down to several sub-attributes. In this study, the attribute maintainability with its sub-attributes such as analyzability, modifiability, testability, modularity and reusability is focussed upon. Maintainability is referred to as the “degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers” [3].

The evaluation techniques at the architectural level are of two types namely questioning and measuring techniques. Questioning technique involves scenarios, questionnaires and checklists, and quantitative questions are used, which intend to bring out a discussion about the architecture. But, it does not provide quantifiable answers. So, measuring techniques are preferable which include metrics, simulations, prototype and experiences. Metrics is an important SA evaluation method to assess the non-functional attributes of architecture in a quantitative manner. Various metric set for OO designs are available such as Chidamber and

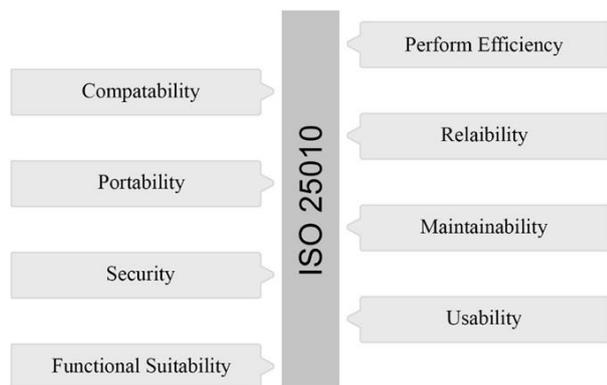


Figure. 1 Software quality attributes

Kemerer metrics (C & K) metrics [4], Metrics for Object Oriented Design (MOOD) metrics [5], Lorenz and Kidd’s Metrics [6], Quality Model for Object Oriented Design (QMOOD) metrics [7], and so on. From the available metric set, QMOOD is the effective metric suite as it employs direct mathematical formulas to determine the quality attributes. The list of design properties and metrics in QMOOD suite is given in Table 1.

As SA quality assessment involves the evaluation of diverse attributes and sub-attributes, it can be considered as a multiple criteria decision making (MCDM) problem. Numerous decision-making methods have also been developed and found in the literature [8]. AHP is the commonly used decision-making tool to address MCDM problem and is useful in various real-time applications. The five main steps involved in AHP are

- Decomposition of the system to a hierarchical structure
- Generating a pair-wise comparison matrix
- Computing eigen value and eigen vectors
- Conducting consistency tests and calculating weights
- Formula derivation for overall quality analysis

Though AHP is simple and effective, it deals only with crisp inputs and fails to map human judgements indecision-making scenarios, especially for qualitative criteria. So, AHP is enhanced by the inclusion of fuzzy concepts and is named as fuzzy AHP (FAHP). In FAHP, the pair-wise comparison of attributes and sub-attributes takes place using linguistic variable and membership functions. Fuzzy prioritization methods are available to identify fuzzy priorities from the fuzzy comparison matrices. A simple arithmetic mean algorithm is employed in [9] to identify fuzzy priorities. In [10], logarithmic least square method is used for prioritization and is

Table 1. QMOOD metric suite

Design property	Metrics
Design size	Design Size in Class (DSC)
Hierarchies	Number of hierarchies (NOH)
Abstraction	Average Number of Ancestors (ANA)
Encapsulation	Data Access Metrics (DAM)
Coupling	Direct Class Coupling (DCC)
Cohesion	Cohesion Among Methods in Class (CAM)
Composition	Measure of Aggregation (MOA)
Inheritance	Measure of Function Abstraction (MFA)
Polymorphism	Number of Polymorphic Methods (NOP)
Messaging	Class Interface Size (CIS)
Complexity	Number of Methods (NOM)

applied to microencapsulation process selection. [11] combined least square and geometric mean method to compute fuzzy priorities. Buckley [12] employed geometric mean to compute fuzzy priorities with triangular membership functions.

Although various metric sets are already applied to assess the quality attributes of SA, QMOOD metric suite has never been employed. At the same time, only few studies were carried out to analyze maintainability at the architecture level of the object class diagram. So, this paper introduces a new evaluation model using QMOOD metric suite to assess SA against maintainability using FAHP with Buckley method. To improve the performance of FAHP method, the missing values in the pair wise comparison matrix is filled with optimal values using genetic algorithm (GA). The mathematical formula is derived to calculate the priorities of maintainability and its attributes. The effectiveness of the proposed method is compared to FAHP with extent analysis method, AHP-GA and Chawla 25010 method. The experimental results verified that the proposed method is an appropriate way for maintainability analysis of the software system in architectural level. The evaluation of software systems at the earlier stages of SDLC prevents the wastage of cost, time and effort to a major extent.

The systematic organization of the paper is presented as follows. The existing literature on FAHP method in various domains is reviewed in section 2. The proposed methodology is explained in section 3 along with the steps involved in the entire process. Section 4 presents the experimental analysis with necessary diagrams and illustrations. The implementation of this study and concluding remarks are given in section 5.

## 2. Literature review

In this section, the usage of FAHP model in several applications in the real world is reviewed. FAHP with Buckley and extent analysis method was used to investigate the significance of continuing professional development programs (CPDP) on the changes in the conceptions of the teaching and learning processes of lecturers [13]. FAHP with Buckley method was used to address supplier selection problem in [14] and was applied to a gear motor company for the identification of the best supplier from three alternatives. To identify the consumer preferences for the telecom industry in India, a hybrid model using FAHP and data envelopment analysis (DEA) method was introduced [15]. FAHP method was incorporated to time dependency for efficient wafer fabrication process selection [16]. FAHP method was also used to solve the supplier address selection problem of a washing machine company in Turkey [17]; it used macros in MS Excel to determine the priority weights.

Synthetic FAHP extent analysis method was used to compute the final priority weight using TFN in [18]. Another evaluation model using FAHP was proposed to assess manufacturing plant sustainability [19]. Software reliability allocation problem was solved in [20] using FAHP method and a comparative analysis with AHP was made in terms of consistent ratio and sensitivity analysis. Next, FAHP was introduced in the field of heterogeneous wireless sensor network for network selection [21]. Fuzzy extended AHP (FEAHP) model was employed to select best suppliers for an automobile industry in India [22]. A balanced score card FAHP (BSC-FAHP) model was employed to choose the suppliers in the automobile industry in Iran [23]. Few studies on maintainability analysis have also been done and are found in the literature. [24] evaluated the software architecture styles (SAS) on the basis of coupling, complexity and cohesion metrics. The final rank of SAS maintainability area was also defined by AHP method. In [25], the author studied the relationship between internal (size, cohesion, coupling) and external quality attributes (class maintainability). Statistical methods were employed to build models using chosen internal attributes to identify the class maintainability. The obtained results implied that the developers could decrease the maintenance cost by reducing class size and coupling as well as increasing the cohesion. Another study was proposed in [26], to develop a quality assessment model for maintainability using AHP method with the help of QMOOD metric. But,

the use of AHP failed to perform well in the mapping of real-world expert ideas to real values. Though various models for maintainability analysis are proposed, none of the method adopted fuzzy logic concepts in SA assessment. This motivates us to develop a novel evaluation model for security analysis in SA.

## 3. Maintainability analysis using FAHP with Buckley method

The proposed method uses FAHP with Buckley method to perform efficient maintainability analysis in SA assessment. FAHP method is used to map uncertainty or expert’s judgments to equitable values. The entire process involves five steps which are listed below:

- Establish hierarchy
- Collect questionnaires
- Fill missing values using GA
- Computation of weights and
- Formula derivation

### Step 1

In the first step, the attributes involved in maintainability quality are identified as modularity, reusability, analyzability, modifiability and testability. The identified attributes are placed at level 2 with maintainability in level 1. The QMOOD metric suite is employed and the eleven design properties from the metric set are placed in level 3. Using this process, a hierarchical structure is derived as shown in Fig. 2.

### Step 2

In this step, a web-based questionnaire is designed and it allows the experts to express their view using linguistic variables. The experts rank the attributes and metrics, the higher rank denotes the selected metric is considered important at greater degree than other factors it is being compared with. The scale and degree of preference is given in Table 2 along with their TFN. Once the experts use linguistic variable for pair-wise comparison, they are converted to TFN. When the pair-wise comparison matrix is formed, the consistency of the

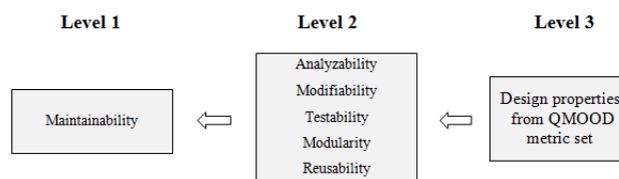


Figure. 2 Hierarchical structure of the entire process

Table2. Linguistic parameters and TFN

Scale	Degree of preference	TFN
1	Equal importance	(1,1,1)
3	Moderate importance of one factor over other	(2,3,4)
5	Strong or essential importance	(4,5,6)
7	Very strong importance	(6,7,8)
9	Extreme importance	(9,9,9)
2		(1,2,3)
4	Intermediate values	(3,4,5)
6		(5,6,7)
8		(7,8,9)

matrix is tested using consistency index and consistency ratio.

**Step 3**

Due to various reasons like lack of knowledge, many alternatives or more number of pair-wise comparisons are needed and the experts may not able to fill the entire questionnaire. This results in incomplete questionnaire responses and presence of missing values in the pair-wise comparison matrix. GA is one of the popular methods to fill optimum values from 1 to 9 in the missing places of the matrix. When the values are filled, the matrix is tested for consistency. When CR does not exceed 10%, the remaining steps will be carried out [27].

**Step 4**

This is the important step in the entire process where the metric weights are calculated and the formula would be derived using FAHP method. There are seven sub-processes involved in calculating weights and are explained below:

i. The pair-wise comparison matrix is updated after the missing values are filled as shown in Eq. (1).

$$\tilde{C}^k = \begin{bmatrix} \tilde{e}_{11}^k & \tilde{e}_{12}^k & \dots & \tilde{e}_{1j}^k \\ \tilde{e}_{21}^k & \dots & \dots & \tilde{e}_{2n}^k \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{e}_{n1}^k & \tilde{e}_{n2}^k & \dots & \tilde{e}_{nn}^k \end{bmatrix} \quad (1)$$

where  $\tilde{C}^k$  represents the pair-wise comparison matrix and  $\tilde{e}_{ij}^k$  indicate the expert's choice of  $i$ th attribute over  $j$ th attribute.

ii. When there are many experts, mean value  $\tilde{e}_{ij}$  of the preferences is computed in Eq. (2).

$$\tilde{e}_{ij} = \frac{\sum_{m=1}^M \tilde{e}_{ij}^k}{K} \quad (2)$$

iii. Using the mean value  $\tilde{e}_{ij}$ , Eq. (1) is updated as given in Eq. (3).

$$\tilde{C} = \begin{bmatrix} \tilde{e}_{11} & \dots & \tilde{e}_{1j} \\ \vdots & \ddots & \vdots \\ \tilde{e}_{n1} & \dots & \tilde{e}_{nn} \end{bmatrix} \quad (3)$$

iv. When Buckley method is employed, the geometric mean of fuzzy comparison value of every matrix is computed in Eq. (4).

$$\tilde{r}_i = (\prod_{j=1}^n \tilde{d}_{ij})^{1/n}, i = 1, 2, \dots, n \quad (4)$$

where  $\tilde{r}_i$  represents TFN.

v. Now the fuzzy weights of every metric are computed using Eqs. (5) and (6).

vi.

$$\begin{aligned} \tilde{w}_i &= \tilde{r}_i \otimes (\tilde{r}_1 \oplus \tilde{r}_2 \oplus \dots \oplus \tilde{r}_n)^{-1} \\ &= (l_i, m_i, u_i) \end{aligned} \quad (5)$$

where  $\tilde{w}_i$  represents the fuzzy weights and  $(l, m, u)$  represent the lower, middle and upper limit of TFN.

vii. The computed weight  $\tilde{w}_i$  is a fuzzy number which is converted to crisp number using centre of area (COA) method [11].

$$D_i = \frac{(l_i, m_i, u_i)}{3} \quad (6)$$

viii. Though  $D_i$  is the crisp number which represents the metric weight, the value should be normalized using Eq. (7)

$$N_i = \frac{D_i}{\sum_{i=1}^n D_i} \quad (7)$$

The normalized value  $N_i$  represents the weight of every metric and the above seven sub-processes should be repeated to compute the weights of the five attributes to formulize the software quality maintainability. Using the normalized value of metrics, the formulas for the five attributes are derived as shown in Eqs. (8) - (12) respectively.

**Modularity:** It is defined as the “degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components”.

$$\begin{aligned} \text{Modularity} &= (0.6 \times \text{coupling}) + (0.66 \times \text{composition}) + \\ & (0.57 \times \text{complexity}) - (0.52 \times \text{cohesion}) - \\ & (0.31 \times \text{messaging}) \end{aligned} \quad (8)$$

Table 3. Relationship between design properties and attributes

	Modularity	Reusability	Analyzability	Modifiability	Testability
Design size				↓	
Hierarchies		↓		↑	↓
Abstraction		↑	↓		
Encapsulation			↓	↓	
Coupling	↑		↑		
Cohesion	↓	↑	↓	↑	
Composition	↑				
Inheritance					↑
Polymorphism		↑			↑
Messaging	↓				↑
Complexity	↑	↓	↑	↑	↓

*Reusability:* Using ISO 25010, reusability can be defined as the “degree to which an asset can be used in more than one system or in building other assets”. The design properties which affect the integrity are abstraction, cohesion, hierarchies and coupling. FAHP method generates the formula to compute reusability as given in Eq. (9)

$$\text{Reusability} = 0.72 \times \text{abstraction} + 0.81 \times \text{cohesion} - 0.80 \times \text{complexity} - 0.79 \times \text{hierarchies} + 0.5 \times \text{polymorphism} \quad (9)$$

*Analyzability:* It can be stated as the “degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified”. The chosen metrics are coupling, abstraction, complexity and encapsulation.

$$\text{Analyzability} = 0.30 \times \text{coupling} + 0.45 \times \text{complexity} - 0.59 \times \text{abstraction} - 0.61 \times \text{encapsulation} - 0.55 \times \text{cohesion} \quad (10)$$

*Modifiability:* By ISO 25010, modifiability is the “degree to which the actions of an entity can be traced uniquely to the entity”. The design properties used to equate modifiability are polymorphism, coupling, cohesion and hierarchies.

$$\text{Modifiability} = 0.6 \times \text{hierarchies} + 0.65 \times \text{cohesion} + 0.62 \times \text{complexity} - 0.42 \times \text{encapsulation} - 0.45 \times \text{design size} \quad (11)$$

*Testability:* It is the “degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.” The design properties which influence testability are inheritance, complexity, polymorphism and messaging.

$$\text{Testability} = 0.59 \times \text{inheritance} + 0.53 \times \text{polymorphism} + 0.56 \times \text{messaging} - 0.32 \times \text{complexity} - 0.36 \times \text{hierarchies} \quad (12)$$

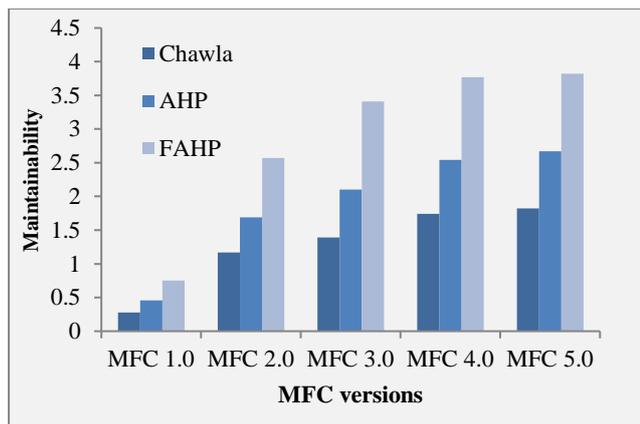


Figure. 3 Comparative analysis of various methods for different versions of MFC

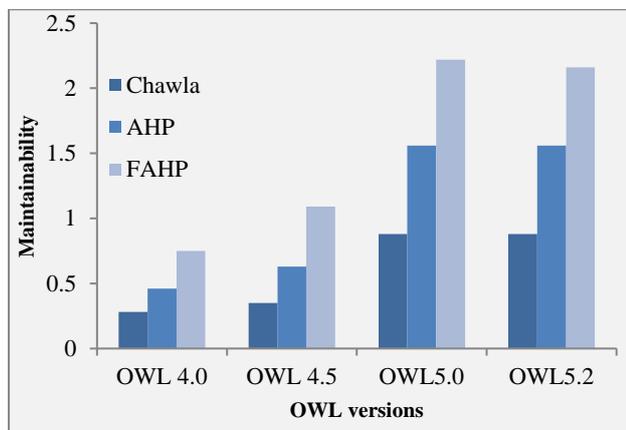


Figure. 4 Comparative analysis of various methods for different versions of OWL

The FAHP process is repeated to compute the weight of the five attributes to identify their importance in maintainability of quality and the derived formula is given in Eq. (13)

$$\text{Maintainability} = 0.49 \times \text{modularity} + 0.1 \times \text{reusability} + 0.28 \times \text{modifiability} + 0.11 \times \text{analyzability} + 0.19 \times \text{testability} \quad (13)$$

Once the fuzzy weights of all attributes and metrics were found, the formula to compute maintainability was determined. The relationship between the design properties and the attributes are tabulated in Table 3.

## 4. Experimental analysis

### 4.1 Implementation

The performance of the proposed method is evaluated by implementing it in five versions of MFC and four versions of OWL [28]. The modified versions of same software are chosen to identify the consistency and enhancement of the proposed method. The obtained normalized results metrics and attributes of proposed FAHP with Buckley

method is compared with AHP and Chawla method. The proposed method is validated using the maintainability index (MI) measure and the effectiveness is verified by comparing its MI value with AHP and Chawla (ISO 25010) method [28].

### 4.2 Results and discussion

The proposed method was applied to five versions of MFC and four versions of OWL to determine the normalized weight of metrics and quality attributes of maintainability. Using the normalized weights of quality attributes, the formula was derived and the corresponding results of metrics and quality attributes are tabulated in Tables 4 and 5, respectively.

To highlight the efficiency of the proposed method for maintainability analysis, it is compared to Chawla (25010) and AHP method in terms of maintainability. The obtained results of FAHP and existing methods are tabulated in Table 6 and also illustrated in Figs. 3 and 4. From this table, it is clear that FAHP method attains better maintainability than the other two methods for all versions MFC and OWL.

Here, for Chawla method, the maintainability of MFC 1.0 is 0.28, MFC 2.0 is 1.17, MFC 3.0 is 1.39, MFC 4.0 is 1.74 and MFC 5.0 is 1.82. For OWL versions, Chawla method attains the maintainability for OWL 4.0 is 0.28, OWL 4.5 is 0.35, OWL 5.0 is 0.88 and OWL 5.2 is 0.88 respectively. Likewise, for AHP method, the maintainability of MFC 1.0 is 0.46, MFC 2.0 is 1.69, MFC 3.0 is 2.1, MFC 4.0 is 2.54 and MFC 5.0 is 2.67. Similarly, for OWL versions, AHP method attains the maintainability for OWL 4.0 is 0.46, OWL 4.5 is 0.63, OWL 5.0 is 1.56 and OWL 5.2 is 1.56 respectively. Similarly, for FAHP method, the maintainability of MFC 1.0 is 0.75, MFC 2.0 is 2.57, MFC 3.0 is 3.41, MFC 4.0 is 3.77 and MFC 5.0 is 3.82. Likewise, for OWL versions, FAHP method attains the maintainability for OWL 4.0 is 0.75, OWL 4.5 is 1.09, OWL 5.0 is 2.22 and OWL 5.2 is 2.16 respectively. FAHP method achieves higher MI of 3.82 for MFC 5.0 whereas AHP and Chawla attain a value of 2.67 and 1.82, respectively. Particularly, for OWL 5.2, FAHP attains a maximum value of 2.16 which is much higher than the compared methods. From the table, it is also observed that the proposed method attains higher performance for latest versions of software compared to older versions. A lower value of 0.75 is attained for older versions of MFC 1.0 and the higher value of 3.82 is obtained for the recent version MFC 5.0. Likewise, the lower value of 0.75

Table 4. Normalized values of design metrics for several versions

METRIC	MFC 1.0	MFC 2.0	MFC 3.0	MFC 4.0	MFC 5.0	OWL 4.0	OWL 4.5	OWL5.0	OWL5.2
Design size	1.00	1.28	1.83	2.86	3.24	1.00	1.73	4.35	4.34
Hierarchies	1.00	1.00	1.00	5.00	6.00	1.00	1.33	2.42	2.25
Abstraction	1.00	1.26	1.46	1.39	1.37	1.00	0.97	1.56	1.55
Encapsulation	1.00	0.79	0.88	0.92	0.94	1.00	0.94	0.88	0.89
Coupling	1.00	1.26	1.50	1.55	1.48	1.00	1.21	2.53	2.52
Cohesion	1.00	0.74	0.81	0.79	0.83	1.00	1.00	1.00	1.00
Composition	1.00	3.46	5.27	5.23	5.49	1.00	1.90	3.85	3.72
Inheritance	1.00	5.39	6.55	7.16	7.57	1.00	0.92	1.06	1.06
Polymorphism	1.00	2.66	2.83	4.14	4.19	1.00	1.66	3.71	3.59
Messaging	1.00	1.81	2.23	2.65	2.55	1.00	0.95	1.83	1.80
Complexity	1.00	1.70	2.13	2.56	2.44	1.00	0.87	1.34	1.33

Table 5. Normalized values of quality attributes for several versions

	MFC 1.0	MFC 2.0	MFC 3.0	MFC 4.0	MFC 5.0	OWL 4.0	OWL 4.5	OWL 5.0	OWL 5.2
Modularity	1.00	3.06	4.48	4.61	4.68	1.00	1.66	3.74	3.65
Reusability	-1.00	-1.13	-1.47	-4.29	-4.91	-1.00	-0.81	-1.44	-1.35
Modifiability	1.00	1.23	1.25	3.43	3.80	1.00	0.81	0.61	0.50
Analyzability	-1.00	-0.49	-0.44	-0.20	-0.30	-1.00	-0.94	-0.65	-0.65
Testability	1.00	4.70	5.57	5.28	5.17	1.00	1.20	2.32	2.30
<b>Maintainability</b>	<b>0.75</b>	<b>2.57</b>	<b>3.41</b>	<b>3.77</b>	<b>3.82</b>	<b>0.75</b>	<b>1.09</b>	<b>2.22</b>	<b>2.16</b>

Table 6. Comparison of various methods interms of maintainability

	MFC 1.0	MFC 2.0	MFC 3.0	MFC 4.0	MFC 5.0	OWL 4.0	OWL 4.5	OWL5.0	OWL5.2
Chawla	0.28	1.17	1.39	1.74	1.82	0.28	0.35	0.88	0.88
AHP	0.46	1.69	2.1	2.54	2.67	0.46	0.63	1.56	1.56
FAHP	0.75	2.57	3.41	3.77	3.82	0.75	1.09	2.22	2.16

is attained for older versions of OWL 4.0 and the higher value of 2.16 is obtained for the recent version OWL 5.2. The existing methods show poor performance due to the presence of missing values in the comparison matrix. The inclusion of GA in FAHP method enhances the performance of the proposed method significantly. From the above analysis, it is concluded that the proposed method is much efficient for maintainability analysis of SA, not only for the applied software but also for any software systems.

## 5. Conclusion

To validate the maintainability of a software system at the initial stage of the SDLC, a new evaluation model using FAHP with Buckley method was proposed in this study. In addition, GA is used to fill the missing places in the pair-wise comparison matrix. The proposed method was used to compute

the normalized weights of the metrics and attributes. Using the obtained weights, the formulas for quality attributes and maintainability were derived. The performance of the proposed method was tested against multiple versions of MFC and OWL windows application frameworks. The FAHP method was compared with two state-of-the-art methods namely Chawla ISO 9126 and Chawla ISO 25010. The average MI of FAHP method was 1.73 whereas existing methods achieved a value of 1.01 and 1.04, respectively. The proposed method is thus found to be a better choice to validate SA for maintainability criteria. In future, this work can be extended by the application of FAHP model to assess various quality attributes other than maintainability. In addition, the performance can be enhanced by the optimization of SA using evolutionary algorithms.

## References

- [1] G. Abowd, L. Bass, P. Clements, R. Kazman, L. Northop, and A. Zaremski, *Recommended Best Industrial Practice for Software Architecture Evaluation*, CMU/SEI-96-TR-025, Carnegie-Mellon Univ, Pittsburgh, P.A., 1997.
- [2] C.-H. Lung and K. Kalaichelvan, "An Approach to Quantitative Software Architecture Sensitivity Analysis", *International Journal of Software Engineering and Knowledge Engineering*, Vol. 10, No. 1, pp. 97-114, 2000.
- [3] P. Bengtsson, "Towards Maintainability Metrics on Software Architecture: An Adaptation of Object-Oriented Metrics", In: *Proc. of First Nordic Workshop on Software Architecture*, 1998.
- [4] S. R. Chidamber and C. F. Kemerer, "Towards a Metrics Suite for Object Oriented Design", In: *Proc. of Sixth Annual ACM SIGPLAN Conf. on OOPSLA*, 1991.
- [5] F. B. e Abreu and W. Melo, "Evaluating the Impact of Object-Oriented design on Software Quality", In: *Proc. of the Third International Software Metrics symposium*, pp. 90-99, 1996.
- [6] M. Lorenz, *Object-Oriented Software Metrics: A Practical Guide*, Prentice-Hall, Upper Saddle River, N.J., 1994.
- [7] J. Bansiya and C. Davis, "A Hierarchical Model for Object-Oriented Design Quality Assessment", *IEEE Transactions on Software Engineering*, Vol. 28, No. 1, pp. 14-17, 2002.
- [8] H. S. Kilic, "An Integrated Approach for Supplier Selection in Multi-Item/Multi-Supplier Environment", *Applied Mathematical Modelling*, Vol. 37, pp. 7752-7763, 2013.
- [9] D. A. Chang, "Application of the Extent Analysis Method on Fuzzy AHP", *European Journal of Operational Research*, Vol. 95, pp. 649-655, 1996.
- [10] Y. C. Erensal, T. Oncan, and M. L. Demircan, "Determining Key Capabilities in Technology Management Using Fuzzy Analytic Hierarchy Process: A Case Study of Turkey", *Information Sciences*, Vol. 176, pp. 2755-2770, 2006.
- [11] P. J. M. van Laarhoven and W. Pedrycz, "Fuzzy Extension for Saaty's Priority Theory", *Fuzzy Sets and Systems*, Vol. 11, pp. 229-241, 1983.
- [12] J. J. Buckley, "Fuzzy Hierarchical Analysis", *Fuzzy Sets and Systems*, Vol. 17, pp. 233-247, 1985.
- [13] Y. Owusu-Agyeman, O. Larbi-Siaw, B. Brenyaa, and A. Anyidoho, "An Embedded Fuzzy Analytic Hierarchy Process for Evaluating Lecturers' Conceptions of Teaching and Learning", *Studies in Educational Evaluation*, Vol. 55, pp. 46-57, 2017.
- [14] M. B. Ayhan, "A fuzzy AHP Approach for Supplier Selection Problem: A Case Study in a Gear Motor Company", *arXiv Prepr. arXiv*, p. 1311.2886, 2013.
- [15] A. Kumar, R. Shankar, and R. M. Debnath, "Analyzing Customer Preference and Measuring Relative Efficiency in Telecom Sector: A hybrid Fuzzy AHP/DEA Study", *Telematics and Informatics*, Vol. 32, No. 3, pp. 447-462, 2015.
- [16] H. C. Rajput, A. S. Milani, and A. Labun, "Including Time Dependency and ANOVA in Decision-Making Using the Revised Fuzzy AHP: A Case Study on Wafer Fabrication Process Selection", *Applied Soft Computing*, Vol. 11, pp. 5099-5109, 2011.
- [17] O. Kilincci and S. A. Onal, "Fuzzy AHP Approach for Supplier Selection in a Washing Machine Company", *Expert System with Applications*, Vol. 38, pp. 9656-9664, 2011.
- [18] C. Kahraman, *Fuzzy Multi-criteria Decision Making: Theory and Applications with Recent Developments*, Springer Science & Business Media, New York, N.Y., 2008.
- [19] H. M. M. Jayawickrama, A. K. Kulatungaa, and S. Mathavanb, "Fuzzy AHP Based Plant Sustainability Evaluation Method", *Procedia Manufacturing*, Vol. 8, pp. 571-578, 2017.
- [20] S. Chatterjee, J. B. Singh, and A. Roy, "A Structure-Based Software Reliability Allocation Using Fuzzy Analytic Hierarchy process", *International Journal of Systems Science*, Vol. 46, No. 3, pp. 513-525, 2015.
- [21] R. K. Goyal, S. Kaushal, and A. Sangaiah, "The Utility Based Non-linear Fuzzy AHP Optimization Model for Network Selection in Heterogeneous Wireless Networks", *Applied Soft Computing*, 2017.
- [22] V. Yadav and M. K. Sharma, "Multi-criteria Decision Making for Supplier Selection Using Fuzzy AHP Approach", *Benchmarking An Int. Journal*, Vol. 22, No. 6, 2015.
- [23] M. R. Galankashi, S. A. Helmi, and P. Hashemzahi, "Supplier Selection in Automobile Industry: A Mixed Balanced Scorecard-Fuzzy AHP Approach", *Alexandria Engineering Journal*, Vol. 55, pp. 93-100, 2016.
- [24] G. ShahMohammadi, "Evaluation of the Software Architecture Styles from Maintainability Viewpoint", *International*

- Journal of Computer Science and Information Technology*, Vol. 6, No. 1, pp. 183-197, 2014.
- [25] J. Al Dallah, "Object-Oriented Class Maintainability Prediction Using Internal Quality Attributes", *Information and Softw. Technology*, Vol. 55, No. 11, pp. 2028-2048, 2013.
- [26] D. Udaya and K. Vivekanandan, "An Empirical Evaluation Model for Software Architecture Maintainability for Object oriented Design", In: *Proc. of the International Conf.on Informatics and Analytics*, 2016.
- [27] D. Udaya and K. Vivekanandan, "A Quantitative Approach to Evaluate Software Architecture Security and Filling Missing Values Using Genetic Algorithm", *International Journal of Emerging Technological Advanced Engineering*, Vol. 7, No. 9, pp. 403-413, 2017.
- [28] M. K. Chawla and I. Chhabra, "SQMMA: Software Quality Model for Maintainability Analysis", In: *Proc. of the eighth Annual ACM India Conference*, pp. 9-17, 2015.