International Journal of Psychosocial Rehabilitation, Vol. 24, Issue 08, 2020
ISSN: 1475-7192

APPLICATION OF ARCHITECTURAL KNOWLEDGE BASED GENETIC ALGORITHM AND FUZZY TOPSIS IN DECISION MAKING

[1]C. Dhaya, [2]R. Dhanalakshmi

ABSTRACT

Architectural knowledge is important for the architecting process, as it improves the quality of the software architecture evaluation process and the architecture itself [18]. All the stakeholders need to obtain relevant architectural knowledge in making design decisions. It has been stated that the major challenge in assessing software design is an exact description of the quality characteristics and specific knowledge about the design decisions. Though many qualitative architecture assessment methods are available, a quantitative evaluation method is needed to evaluate the candidate architectures over a set of architectural design characteristics. Software architecture evaluation framework addresses the competing objectives of cost minimization and quality maximization between different architectural options. Due to the uncertainty in the judgment for design quality characteristics, architectural knowledge based fuzzy genetic algorithm framework is developed for accessing quality attributes is developed to assist the selection of the underlying architectural designs. A new multi-criteria decision making (MCDM) method , Architectural knowledge based fuzzy Genetic Algorithm and Technique for Order of Preference by Similarity to Ideal Solution(FGA-TOPSIS) is proposed to describe the quality requirements of the envisioned system which forms the basis for the comparison and selection criteria. Experimental results obtained indicate that FGA-TOPSIS can be used as a feasible and effective a multi criteria decision making approach for architecture selection under partial or incomplete information (uncertainty).

Keywords*: Architectural Knowledge, Genetic Algorithm, Decision Making, Fuzzy TOPSIS.*

[1] Professor, Department of CSE, Adhiparasakthi Engineering College
[2] Professor, Department of CSE, KCG College of Engineering

I. Introduction

Software architecture [1] has been identified as an increasingly important part of software development. Software architecture design and evaluation are closely related activities. Software architectural evaluation becomes a well-known practice in software engineering community for developing the high quality software. Architectural evaluation reduces software development effort and costs and flaws in the early stages of the software development. Moreover, the evaluation process enhances the quality of the software by verifying the addressability of quality requirements [2]. Software architecture evaluation is a methodology which determines the properties, strengths and weaknesses of the software architecture or a software architectural style or frameworks. It assures the developers that their chosen architecture will meet both functional and non-functional requirements. A number of evaluation methods have been developed which are applicable in different phases of the software development cycle [3]. The difficult task in software development is to attain maximum quality with the estimated budget. In the proposed work, the best architectures are analyzed for the given goal which concentrates both on the risk and quality factor for the economic benefit. To ensure good management for any project, it's important to select the best alternative among a set of feasible alternatives, when considering several quality criteria's. The decision makers provide qualitative/quantitative assessments for determining the performance of each alternative with respect to each criterion, and the relative importance of evaluation criteria's with respect to the objective of the problem. Decision making process is often difficult and tricky when the subjective data's are present and the results are uncertain. When the information available is imprecise and uncertain in the architecture evaluation process, Fuzzy Genetic Algorithm is used for calculating the priority vector of quality criteria's which maximizes the triangular membership function. The ranking of the alternatives is defined by the technique for order preference by similarity to ideal solution that defines the positive ideal solution and negative ideal solution to maximize the benefit criteria and minimize the cost criteria.

II. Related Work

Software Architecture Evaluation process assess the system's quality attributes with respect to the software requirements of its developers, customers and architects. Detection in the initial stages of software development is less expensive to fix design errors [4]. Existing Scenario-based software architecture evaluations assess only a specific quality attribute for the given scenarios. It needs stakeholder's involvement in creating scenarios which improves documentation. Some of the mature scenarios based techniques are Architecture Tradeoff Analysis Method (ATAM)[5], Software Architecture Analysis Method (SAAM)[6], Architecture-Level Modifiability Analysis (ALMA)[7], Cost-Benefit Analysis Method (CBAM)[8], and Family-Architecture Assessment Method (FAAM)[9]. Clements et al[10] written a paper to integrate the ATAM and the CBAM. The CBAM takes the analysis done during the ATAM which helps in making the software design by relating priorities, costs, and benefits of architectural decisions.

M. Svanhberg et al.[11] propose a new framework using AHP for comparing different software architectures for a specific quality attribute and vice versa. Using a method like AHP in the quality driven software architecture evaluation process is an easier process. The drawback of using AHP is its inability to solve uncertainty in decision-making problems. In standard AHP, human judgments are represented as exact (or crisp) numbers. However, in many practical situations, the human's decision is uncertain and the decision-makers are unable to assign exact numerical values to the comparison judgments[12]. This uncertainty to capture the right judgments is insufficient and imprecise in the Quantitative assessment of Quality Attributes. Javanbarg et al.[13] proposed a method which derives crisp weightages from fuzzy comparison matrices. The application of fuzzy set theory has proven to be an effective approach[14]. The effectiveness of the alternatives with respect to all criteria is often measured by a fuzzy number where they are ranked by comparing with the corresponding fuzzy utilities. The technique for order preference by similarity to ideal solution called as TOPSIS[15] is one of the well-known methods for MCDM problems. Positive ideal and negative ideal solutions help in solving decision making problems with different decision makers.

Aleti[16] presented the systematic review on software architecture optimization, which aims in automating the finding the optimum selection with respect to a set of quality attributes. Both quality criteria and cost factor is important for selecting optimum architecture. In current practice, software architects try to find the solutions manually, which is time-consuming and error-prone that leads to suboptimal designs. Dhaya et. al [17] proposed a architectural development using Architectural Knowledge to support a framework for capturing and using architectural knowledge to improve the architecture evaluation. To automate the task, this paper proposes the evolutionary algorithm for prioritization determination and multi-objective optimization strategies for the identification of good architectures. The paper is organized as follows. Section 3 describes how the decision making process is applied for different design alternatives using Proposed Fuzzy Genetic algorithm based TOPSIS Decision Making Method followed by the demonstration through a case study in Section 4. Section 5 shows the experimental results in selecting the preferred conceptual design from a set of alternatives under various multiple criteria's with maximum benefits and minimum cost. Section 6 concludes the work of the paper.

III. Proposed Architectural Knowledge based Fuzzy Genetic Algorithm and TOPSIS Decision Making Method

With evaluation alternatives evaluation criteria the decision making problem is outlined in hierarchical structure. a priority vector of criteria with respect to goal. is important weight of alternative respective to criterion . The steps involved in FGA-TOPSIS are illustrated in Fig. 1.

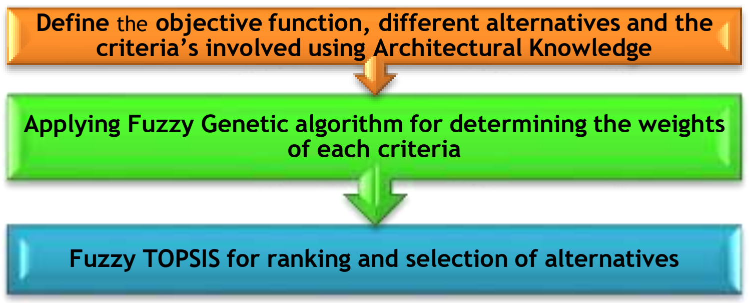
Define the objective function, different alternatives and the criteria's involved using Architectural Knowledge
Applying Fuzzy Genetic algorithm for determining the weights of each criteria
Fuzzy TOPSIS for ranking and selection of alternatives

Fig.1. AK based FGA – TOPSIS Decision making method

3.1 Proposed Fuzzy Genetic Algorithm

The pair-wise comparison judgments in linguistic form for criteria is . It is necessary to prioritize criteria's and determine the priority vector where for the further evaluation by the decision makers. Membership functions of fuzzy numbers may be taken as triangular or trapezoidal. The triangular fuzzy numbers are given in the form of triplets = (is evaluating the importance of factor i relative to factor j as judged by a decision maker. A scale quantifying linguistic judgments fuzzy numbers is given in the following Table 1.

Table 1 Linguistic variables for the importance weight of each criterion

Linguistic variables	**Importance weight**
Very low (VL)	(0, 0, 0.1)
Low (L)	(0, 0.1, 0.3)
Medium low (ML)	(0.1, 0.3, 0.5)
Medium (M)	(0.3, 0.5, 0.7)
Medium high (MH)	(0.5, 0.7, 0.9)
High (H)	(0.7, 0.9, 1.0)
Very high (VH)	(0.9, 1.0, 1.0)

Received: 18 Apr 2020 | Revised: 09 May 2020 | Accepted: 02 Jun 2020

Since $EW_{ij} = 1/EW_{ij}$ the matrix of pair-wise comparisons is sufficiently defined by

$n(n-1)/2$, which are the numbers of the elements in the upper triangle of the matrix for which $(i < j), (i = 1, 2, \dots, n-1), (j = 2, 3, \dots n)$.

Triangular Fuzzy membership function μ_{ij} is defined as follows:

$$\mu_{ij}\left({}^{w_i}/_{w_j}\right) = \begin{cases} \dfrac{{}^{w_i}/_{w_j} - x_{ij}^{l}}{x_{ij}^{m} - x_{ij}^{l}} & x_{ij}^{l} \le {}^{w_i}/_{w_j} \le x_{ij}^{um} \\ \dfrac{{}^{w_i}/_{w_j} - x_{ij}^{u}}{x_{ij}^{m} - x_{ij}^{u}} & x_{ij}^{m} \le {}^{w_i}/_{w_j} \le x_{ij}^{u} \\ 0 & Otherwise \end{cases} \qquad (1)$$

The fitness function is proposed by Moneim[22] as,

$$G(w_1, w_2, \dots . w_n) = \min_{i<j}(\mu_{12}, \mu_{13}, \dots . \mu_{ij}, \dots . \mu_{(n-1)n}) \qquad (2)$$

The problem of deriving a priority vector of n criteria can be given in the following optimization problem as

$$\begin{cases} Maximize\ G(w_1, w_2, \dots . w_n)\ subject\ to\ \sum_{i=1}^{n} w_i = 1 \\ where\ G(w_1, w_2, \dots . w_n) \end{cases} \qquad (3)$$

Genetic Algorithm [18] (GA) is an optimization algorithm based on the evolutionary ideas of natural selection and genetics to solve the problem formulated. The GA is a stochastic popular search heuristic that mimics the representation of natural biological evolution. GAs operates on a population of initial decision variables as chromosomes by applying the principle of survival of the fittest to produce better approximations. The quality of the solution is defined by the fitness function. The priority vector of criteria's is coded as real number chromosome. Each gene in the chromosome represents the weight of the criterion which lies in the range of 0 and 1. The representation of the chromosome is shown in Fig. 2.

W_1			W_i				W_n

Fig. 2. **Priority Vector with weights as genes**

Steps in the Fuzzy Genetic algorithm

- Generate a random chromosome, with genes uniformly distributed in the range of 0 and 1.

- The fitness of each chromosome in the population is evaluated using the Elitism function.
- Repeat the above steps until an initial population of n chromosomes are generated.
- Decide the crossover probability p_{cross} and mutation probability p_{mutat} to start reproduction. New population is created by repeating the following steps until the new population is complete.
  - –Two parent chromosomes from a population are selected with maximum fitness.
  - –A continuous random number x if generated. If $(x \; <= \; p_{cross})$ crossover the parents to form new offspring (children). If no crossover is performed, offspring is the exact copy of parents.
  - –A continuous random number y if generated. If $(y \; <= \; p_{mutat})$ mutate the cross-over offspring's at a locus (position in chromosome).
  - –New offspring's are placed in the new population.
- Use newly generated population for further iterations of the algorithm.
- If the end condition is satisfied, stop the iteration, and return the best solution that has the highest fitness value in the final population.

The algorithm for determining priority vector is represented in Algorithm 1. After the priority vector of each criterion is determined by the fuzzy genetic algorithm, Fuzzy TOPSIS decision making method is used to rank the alternatives.

Setting the Initial Population

```
population size : pop-size
current generation : gene ← 0
number of generations :  max-gen
number of offsprings : n – offspring
crossover probability : $p_{cross}$
mutation probability : $p_{mutat}$
crossover random number : $C_x$
mutation random number : $m_x$

for $i$ ← 1 to pop-size do
    insert the chromosomes from the created
    population for evolution
end
```

Fitness function

```
for $i$ ← 1 to pop-size do
$Fitness\ function\ (S) = \min_{i<j}(\mu_{12}, \mu_{13}, \mu_{ij}, \mu_{(n-1)n})$
end
new pop-size ←  0
for $i$ ← 1 to pop-size do
    if $\forall(X_i \in S)$
        insert $X_i$ to the next generation
        new pop-size = new pop-size +1
    end
end
```

Crossover Operation

```
n – offspring ← 0
for $i$ ← 1 to two selected chromosomes do
  if ($C_x < p_{cross}$)
     x ← random (chromosome)
     y ← random ( chromosome )
     crosspoint ← cutpoints at x and y
     create two offspring using new crossover
  end
end
if (deterministic)
     n – offspring ←n – offspring + 2
else
     selection is probabilistic
end
```

Sampling Space Determination

```
if (selection is deterministic) then
    for $i$ ← 1 to n – offspring do
      insert offspring $i$ to the population
    end
end
if (selection is probabilistic) then
    for $i$ ← 1 to n – offspring do
       replace offspring $i$ by its parents and
       insert them to the population
    end
```

```
**Sorting Operation**
for $i \leftarrow 1$ to new pop-size do
    sort the chromosomes based on their fitness
end

**Selection Operation**
for $i \leftarrow 1$ to new pop-size do
    select the top two parents from the sorted
    population for breeding process
end
end
**Mutation Operation**
for $i \leftarrow 1$ to two crossover chromosomes do
    if ($m_x < p_{mutat}$)
        x ← random ( chromosome )
        y ← random ( chromosome )
        determine $\bar{X}$ for the chromosomes
        replace $\bar{X}$ at the mutation point
    end
end
**Termination Test**
gen ← gen +1
if (gen < max_gen) then
    goto sorting operation
else
    goto Fitness function calculation
end
```

Algorithm 1. Proposed Fuzzy Genetic Algorithm to determine the Priority Vector

3.2 Proposed Fuzzy TOPSIS Decision Making Method

The TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) is a multi-criteria decision analysis method, which was developed by Hwang & Yoon[19]. The basic concept in this method is that the chosen alternative should have the shortest distance from the positive ideal solution (FPIS) and longest distance from the negative ideal solution (FNIS) [20]. The positive- ideal solution is a solution that maximizes the benefit criteria and minimizes the cost criteria, whereas the negative ideal solution maximizes the cost criteria and minimizes the benefit criteria [21]. In addition to PIS and NIS, the Euclidean distance is used to evaluate the relative closeness of alternatives to the ideal solution. Thus, the preference order of alternatives is ranked according to their relative closeness coefficients.

Algorithm 2: Proposed Fuzzy TOPSIS algorithm for ranking the alternatives

Step 1: *Determine Linguistic ratings and Construct Fuzzy Decision Matrix*

The fuzzy linguistic ratings $\tilde{x}_{ij}, (i = 1, \ldots, m, j = 1, \ldots, n)$, for alternatives $A_i$with respect to criteria $C_j$are identified, the appropriate linguistic variables for the weights of the criteria $\tilde{w}_j, (j = 1, \ldots, n)$are determined. Then, the Fuzzy Decision Matrix is constructed for different alternatives.

Step 2: *Normalize Fuzzy Decision Matrix*

In several MCDM problems, the raw data are normalized to eliminate anomalies with different measurement units and scales. Normalization of fuzzy decision matrix is accomplished using linear scale transformation. This is used to preserve that the ranges of normalized triangular fuzzy numbers to be included lies in the range $[0, 1]$. Suppose $\tilde{R}$ denotes normalized fuzzy decision matrix, then

$$\tilde{R} = [\tilde{r}_{ij}]_{m \times n} \tag{4}$$

where r_{ij} is the normalized value of $x_{ij} = (a_{ij}, b_{ij}, c_{ij})$.

If $(\tilde{x}_{ij}, i = 1, 2, \ldots, m, j = 1, 2 \ldots, n)$are triangular fuzzy numbers, then the normalization process can be performed by

$$\tilde{r}_{ij} = \left(\frac{a_{ij}}{c_j^+}, \frac{b_{ij}}{c_j^+}, \frac{c_{ij}}{c_j^+}\right), j \in B, c_j^+ = \max_i c_{ij}\, if j \in B \tag{5}$$

$$\tilde{r}_{ij} = \left(\frac{a_j^-}{c_{ij}}, \frac{a_j^-}{b_{ij}}, \frac{a_j^-}{a_{ij}}\right), j \in C, \qquad a_j^- = \min_i a_{ij}\, if j \in C \tag{6}$$

where B is the benefit criteria and C is the cost criteria respectively.

Step 3: *Calculate the weighted normalized fuzzy decision matrix*

The weighted normalized fuzzy decision matrix is computed by multiplying the weights of evaluation criteria from the fuzzy genetic algorithm with the normalized fuzzy decision matrix. By using Equation (14), the weighted normalized fuzzy decision matrix $\tilde{V} = [\tilde{v}_{ij}]_{m \times n}$is generated where

$$\tilde{v}_{ij} = \tilde{r}_{ij} (.) \tilde{w}_j \tag{7}$$

According to the weighted normalized fuzzy decision matrix, the elements $\tilde{v}_{ij}, \forall i, j$ are normalized positive triangular fuzzy numbers lies in the ranges of the closed interval $[0,1]$.

Step 4: *Determine FPIS and FNIS*

Compare two triangular fuzzy numbers $\tilde{A} = (a1, a2, a3)$ and $\tilde{B} = (b1, b2, b3)$ to find the maximum and minimum fuzzy numbers as follows

Suppose $\tilde{A}^i = (a_1^i, a_2^i, a_3^i)$, $i = 1, 2, \dots, n$ are n TFN.

Determine minimum fuzzy number by applying the following steps:

- List all a_j^i, $i = 1, 2, \dots, n$; $j = 1, 2, 3$.
- Sort increasingly a_j^i.
- Select the first three a_j^i as minimum TFN of $\tilde{A}^i$, $i = 1, 2, \dots, n$.
- Record this as $\tilde{A}_{min}$ where

$$\tilde{A}_{min} = \wedge_i \tilde{A}_i, i = 1, 2, \dots, n \tag{8}$$

Determine maximum fuzzy number by applying the following steps:

- List all a_j^i, $i = 1, 2, \dots, n$; $j = 1, 2, 3$.
- Sort increasingly a_j^i.
- Select the last three a_j^i as maximum TFN of $\tilde{A}^i$, $i = 1, 2, \dots, n$.
- Record this as $\tilde{A}_{max}$ where

$$\tilde{A}_{max} = \vee_i \tilde{A}_i, i = 1, 2, \dots, n \tag{9}$$

For the benefit criterion, the fuzzy positive ideal solution (FPIS) and fuzzy negative ideal solution (FNIS) is calculated by $\tilde{A}_{max}$ and $\tilde{A}_{min}$. For the cost criterion, the fuzzy positive ideal solution (FPIS) and fuzzy negative ideal solution (FNIS) can be calculated by $\tilde{A}_{min}$ and $\tilde{A}_{max}$ respectively.

Step 5: *Calculate the distances of each alternative to FPIS and FNIS*

The distance between the each alternative $\tilde{v}_{ij}$ with the positive ideal solution $\tilde{A}_{max}$ and the negative ideal solution $\tilde{A}_{min}$ can be calculated by using the distance function with $\propto = 1$.

$$L_i^+ = \sum_{j=1}^{n} D(f, \tilde{v}_{ij}, \tilde{A}_{max}) \qquad i = 1, 2, \dots, n \tag{10}$$

$$L_i^- = \sum_{j=1}^{n} D(f, \tilde{v}_{ij}, \tilde{A}_{min}) \qquad i = 1, 2, \dots, n \tag{11}$$

where L_i^+ denotes represents the distance of alternative A_i from FPIS, and L_i^- is the distance of alternative A_i from FNIS.

Step 6: *Obtain the closeness coefficient*

The closeness coefficient represents the distances to the fuzzy positive ideal solution (FPIS or $\tilde{A}_{max}$) and the fuzzy negative ideal solution (FNIS or $\tilde{A}_{min}$). The closeness coefficient (CC_i) of each alternative is calculated as:

$$CC_i = \frac{L_i^-}{L_i^- + L_i^+} i = 1, 2, \ldots, n$$
(12)

While $L_i^- \geq 0$and $L_i^+ \geq 0$, then, $CC_i \in [0,1]$.

Step 7: *Rank the order of alternatives*

The ranking order of all alternatives is obtained with closeness coefficient, allowing the decision-makers to select the most feasible alternative. An alternative with CC_i nearby 1 indicates that the alternative is close to the fuzzy positive ideal solution and far from the fuzzy negative ideal solution. A large value of closeness coefficient CC_i indicates a good performance of the alternative A_i .

IV. Case Study

Evaluation of architectural styles, design patterns and frameworks employs qualitative reasoning to motivate when and under what circumstances they should be used. This category of evaluation also requires experimental evidence to verify the usage of architectural styles or frameworks in general cases. To test the fitness of solutions developed the framework is applied in the Online Course Registration System. The designer identifies ten potential Web application frameworks [22] for the online course registration system. A quantitative method is needed for selecting the most suitable software architecture from alternative software architectures. The frameworks considered are: Wordpress (A1), Joomala(A2), Drupal(A3), Expression(A4), TextPattern(A5), Contao(A6), Silverstripe(A7), Umbraco(A8), Concrete5(A9), Django(A10).

By collaborating with the stakeholders, the features required for the online course registration are identified and examined as follows

Benefit criteria

- Efficiency(Q1): Ability of a software system to fulfill its purpose
- Interoperability(Q2): Ability to operate successfully by communicating and exchanging information with other external systems
- Reliability(Q3): Measured as its mean time to failure
- Security(Q4): System's ability to resist unauthorized usage
- Extensibility(Q5): Ability of the software to be extended beyond the functionality

- Availability(Q6): Proportion of time that the system is functional and working

Cost criteria

- Budget(Q7): Amount spend for the software development

V. Experimental Results

Step 1

The decision makers use the linguistic variables to assess the importance of the quality criteria by rating the alternatives with respect to each quality criterion and are tabulated in Table 2.

Table 2 Decision Makers ratings for different Design Alternatives

Quality Criteria's	Design Alternatives	Decision Makers				
		D1	D2	D3	D4	D5
Q1	A1	Poor	Medium Good	Fair	Good	Good
	A2	Fair	Fair	Medium Good	Fair	Very Good
	A3	Fair	Good	Medium Good	Fair	Medium Good
	A4	Good	Fair	Medium Poor	Poor	Very Poor
	A5	Very Poor	Very Poor	Fair	Very Poor	Poor
	A6	Medium Poor	Medium Good	Good	Very Good	Good
	A7	Very Good	Good	Very Good	Medium Good	Fair

Quality Criteria's	Design Alternatives	Decision Makers				
		D1	D2	D3	D4	D5
	A8	Medium Good	Very Good	Medium Good	Medium Good	Fair
	A9	Good	Medium Poor	Very Poor	Poor	Poor
	A10	Medium Good	Poor	Poor	Medium Poor	Very Poor
Q2	A1	Fair	Medium Poor	Good	Very Good	Fair
	A2	Medium Good	Fair	Fair	Medium Good	Poor
	A3	Medium Good	Good	Fair	Good	Very Poor
	A4	Medium Poor	Very Good	Poor	Very Good	Very Good
	A5	Fair	Medium Good	Very Poor	Medium Good	Medium Good
	A6	Good	Very Good	Medium Good	Poor	Good
	A7	Very Good	Medium Good	Fair	Very Poor	Very Good
	A8	Medium Good	Medium Good	Good	Medium Good	Medium Good
	A9	Very Poor	Poor	Fair	Fair	Very Poor

Quality Criteria's	Design Alternatives	Decision Makers				
		D1	D2	D3	D4	D5
	A10	Poor	Medium Poor	Very Poor	Fair	Poor
Q3	A1	Medium Good	Very Good	Good	Good	Fair
	A2	Fair	Medium Good	Medium Poor	Very Good	Poor
	A3	Good	Good	Medium Good	Fair	Very Poor
	A4	Fair	Very Good	Medium Poor	Fair	Very Good
	A5	Very Poor	Medium Poor	Fair	Poor	Medium Good
	A6	Medium Good	Poor	Poor	Very Poor	Fair
	A7	Good	Medium Good	Medium Poor	Very Good	Fair
	A8	Very Good	Very Poor	Medium Good	Medium Good	Poor
	A9	Medium Poor	Poor	Fair	Medium Good	Good
	A10	Poor	Good	Fair	Poor	Medium Poor
Q4	A1	Good	Fair	Good	Good	Very Poor

Quality Criteria's	Design Alternatives	Decision Makers				
		D1	D2	D3	D4	D5
	A2	Fair	Good	Very Good	Good	Poor
	A3	Fair	Fair	Medium Good	Very Good	Good
	A4	Poor	Very Poor	Very Poor	Fair	Medium Good
	A5	Very Poor	Good	Poor	Poor	Very Good
	A6	Very Good	Medium Poor	Fair	Medium Poor	Fair
	A7	Medium Good	Medium Good	Poor	Medium Good	Good
	A8	Medium Good	Medium Poor	Very Poor	Medium Good	Medium Poor
	A9	Poor	Fair	Very Good	Fair	Medium Good
	A10	Medium Poor	Fair	Medium Good	Very Poor	Medium Poor
Q5	A1	Medium Poor	Poor	Medium Good	Medium Poor	Medium Good
	A2	Good	Very Poor	Very Poor	Medium Good	Fair
	A3	Good	Very Good	Poor	Poor	Fair
	A4	Medium Good	Medium Good	Good	Very Good	Poor

Quality Criteria's	Design Alternatives	Decision Makers				
		D1	D2	D3	D4	D5
	A5	Very Poor	Medium Good	Fair	Medium Good	Good
	A6	Fair	Very Good	Fair	Medium Good	Medium Poor
	A7	Very Good	Medium Good	Poor	Poor	Very Poor
	A8	Medium Good	Medium Good	Very Poor	Good	Poor
	A9	Poor	Medium Poor	Fair	Good	Good
	A10	Fair	Poor	Medium Good	Very Good	Medium Good
Q6	A1	Good	Medium Good	Poor	Fair	Fair
	A2	Good	Fair	Good	Medium Good	Very Good
	A3	Very Good	Fair	Fair	Poor	Medium Good
	A4	Fair	Poor	Good	Medium Poor	Poor
	A5	Poor	Good	Fair	Very Good	Fair
	A6	Medium Poor	Medium Poor	Very Poor	Medium Good	Good

Quality Criteria's	Design Alternatives	Decision Makers				
		D1	D2	D3	D4	D5
	A7	Medium Good	Very Poor	Good	Good	Medium Poor
	A8	Medium Good	Poor	Fair	Very Good	Medium Good
	A9	Fair	Good	Fair	Medium Poor	Medium Poor
	A10	Very Poor	Medium Good	Poor	Poor	Very Good
Q7	A1	14.2 Dollars				
	A2	13.5 Dollars				
	A3	12.3 Dollars				
	A4	12.8 Dollars				
	A5	12.2 Dollars				
	A6	13.1 Dollars				
	A7	12.3 Dollars				
	A8	13.1 Dollars				
	A9	12.9 Dollars				
	A10	13.1 Dollars				

Step 2

Fuzzy decision matrix is constructed by converting the linguistic variables into triangular fuzzy numbers. By applying the fuzzy genetic algorithm, the priority vector of the each quality criteria is obtained. The priority criteria will have three genes representing the important quality criteria respective to the goal. Decision maker uses triangular fuzzy numbers to express pairwise-comparisons among quality criteria as shown in Table 3.

Table 3 Pairwise comparisons for different Quality criteria's

Criterion	Linguistic Preference	Fuzzy Number	Criterion	Criterion	Linguistic Preference	Fuzzy Number	Criterion
Q_1	**Very good**	**(9,10,10)**	Q_2	Q_3	**Medium Good**	**(5,7,9)**	Q_4
Q_1	**Medium Good**	**(5,7,9)**	Q_3	Q_3	**Medium Poor**	**(1,3,5)**	Q_5
Q_1	**Good**	**(7,9,10)**	Q_4	Q_3	**Good**	**(7,9,10)**	Q_6
Q_1	**Medium Poor**	**(1,3,5)**	Q_5	Q_3	**Good**	**(7,9,10)**	Q_7
Q_1	**Medium Good**	**(5,7,9)**	Q_6	Q_4	**Good**	**(7,9,10)**	Q_5
Q_1	**Medium Good**	**(5,7,9)**	Q_7	Q_4	**Medium Good**	**(5,7,9)**	Q_6
Q_2	**Medium Poor**	**(1,3,5)**	Q_3	Q_4	**Good**	**(7,9,10)**	Q_7
Q_2	**Medium Poor**	**(1,3,5)**	Q_4	Q_5	**Medium Poor**	**(1,3,5)**	Q_6
Q_2	**Poor**	**(0,1,3)**	Q_5	Q_5	**Poor**	**(0,1,3)**	Q_7
Q_2	**Poor**	**(0,1,3)**	Q_6	Q_6	**Medium Good**	**(5,7,9)**	Q_7
Q_2	**Medium Poor**	**(1,3,5)**	Q_7				

With reference to the Figure 5 and the Algorithm 1, the Fuzzy Genetic Algorithm for determining the priority vector is implemented using Matlab 7 with the following inputs: number of criteria (N = 7); size of population (M =30);

crossover probability (= 90%); mutation probability (= 10%); and number of reproduction (L =100). The solution obtained is displayed in Table 4.

Table 4 Priority Weight of each Quality Criteria

Quality Criteria	**Priority Weight of each Quality Criteria**
Q_1	**(0.756,0.891,0.889)**
Q_2	**(0.117,0.156,0.349)**
Q_3	**(0.357,0.501,0.652)**
Q_4	**(0.546,0.701,0.875)**
Q_5	**(0.091,0.178,0.299)**
Q_6	**(0.711,0.891,0.901)**
Q_7	**(0.457,0.695,0.819)**

The fuzzy decision matrix and the obtained priority weight of each quality criterion is displayed in Table5.

Table 5 Fuzzy decision matrix and Fuzzy weights of each quality criterion.

Design Alternative	**Quality Attributes**																				
	Q1			**Q2**			**Q3**			**Q4**			**Q5**			**Q6**			**Q7**		
A1	**4.4**	**6.2**	**7.8**	**4.6**	**6.4**	**7.8**	**6.2**	**8**	**9.2**	**4.8**	**6.4**	**7.6**	**2.4**	**4.2**	**6.2**	**3.6**	**5.4**	**7.2**	**14.2**	**14.2**	**14.2**
A2	**4.6**	**6.4**	**8**	**3.2**	**5**	**7**	**3.6**	**5.2**	**6.8**	**5.2**	**6.8**	**8**	**3**	**4.2**	**5.6**	**6.2**	**8**	**9.2**	**13.5**	**13.5**	**13.5**
A3	**4.6**	**6.6**	**8.4**	**4.4**	**6**	**7.4**	**4.4**	**6**	**7.4**	**5.4**	**7.2**	**8.6**	**3.8**	**5.2**	**6.6**	**4**	**5.6**	**7.2**	**12.3**	**12.3**	**12.3**
A4	**2.2**	**3.6**	**5.2**	**5.6**	**6.8**	**7.6**	**5**	**6.6**	**7.8**	**1.6**	**2.6**	**4.2**	**5.2**	**6.8**	**8.2**	**2.2**	**3.8**	**5.6**	**12.8**	**12.8**	**12.8**
A5	**0.6**	**1.2**	**2.6**	**3.6**	**5.2**	**7**	**1.8**	**3.2**	**5**	**3.2**	**4.2**	**5.4**	**4**	**5.6**	**7.2**	**4.4**	**6**	**7.4**	**12.2**	**12.2**	**12.2**
A6	**5.8**	**7.6**	**8.8**	**5.6**	**7.2**	**8.4**	**1.6**	**2.8**	**4.6**	**3.4**	**5.2**	**6.8**	**4.2**	**6**	**7.6**	**2.8**	**4.4**	**6**	**13.1**	**13.1**	**13.1**
A7	**6.6**	**8.2**	**9.2**	**5.2**	**6.4**	**7.4**	**5**	**6.8**	**8.2**	**4.4**	**6.2**	**8**	**2.8**	**3.8**	**5.2**	**4**	**5.6**	**7**	**12.3**	**12.3**	**12.3**

A8	**5.4**	**7.2**	**8.8**	**5.4**	**7.4**	**9.2**	**3.8**	**5**	**6.4**	**2.4**	**4**	**5.8**	**3.4**	**4.8**	**6.4**	**4.4**	**6**	**7.6**	**13.1**	**13.1**	**13.1**
A9	**1.6**	**2.8**	**4.4**	**1.2**	**2.2**	**3.8**	**3.2**	**5**	**6.8**	**4**	**5.6**	**7.2**	**3.6**	**5.4**	**7**	**3**	**5**	**6.8**	**12.9**	**12.9**	**12.9**
A10	**1.2**	**2.4**	**4.2**	**0.8**	**2**	**3.8**	**2.2**	**3.8**	**5.6**	**2**	**3.6**	**5.4**	**4.4**	**6**	**7.6**	**2.8**	**3.8**	**5.2**	**13.1**	**13.1**	**13.1**
Weight	**(0.756,0.891,0.889)**			**(0.117,0.156,0.349)**			**(0.357,0.501,0.652)**			**(0.546,0.701,0.875)**			**(0.091,0.178,0.299)**			**(0.711,0.891,0.901)**			**(0.457,0.695,0.819)**		

Step 3

Construct the weighted normalized fuzzy decision matrix as displayed in Table 6.

Table 6 Fuzzy normalized weighted decision matrix of alternative designs

Design Alternatives	Quality Attributes																				
	Q1			Q2			Q3			Q4			Q5			Q6			Q7		
A1	**0.3616**	**0.6005**	**0.7537**	**0.0585**	**0.1085**	**0.2959**	**0.2406**	**0.4357**	**0.6520**	**0.3047**	**0.5217**	**0.7733**	**0.0266**	**0.0912**	**0.2261**	**0.2782**	**0.5230**	**0.7051**	**0.3926**	**0.5971**	**0.7036**
A2	**0.3780**	**0.6198**	**0.7730**	**0.0407**	**0.0848**	**0.2655**	**0.1397**	**0.2832**	**0.4819**	**0.3301**	**0.5543**	**0.8140**	**0.0333**	**0.0912**	**0.2042**	**0.4792**	**0.7748**	**0.9010**	**0.4130**	**0.6281**	**0.7401**
A3	**0.3780**	**0.6392**	**0.8117**	**0.0560**	**0.1017**	**0.2807**	**0.1707**	**0.3267**	**0.5244**	**0.3428**	**0.5869**	**0.8750**	**0.0422**	**0.1129**	**0.2407**	**0.3091**	**0.5423**	**0.7051**	**0.4533**	**0.6893**	**0.8123**
A4	**0.1808**	**0.3487**	**0.5025**	**0.0712**	**0.1153**	**0.2883**	**0.1940**	**0.3594**	**0.5528**	**0.1016**	**0.2119**	**0.4273**	**0.0577**	**0.1476**	**0.2990**	**0.1700**	**0.3680**	**0.5484**	**0.4356**	**0.6624**	**0.7806**
A5	**0.0493**	**0.1162**	**0.2512**	**0.0458**	**0.0882**	**0.2655**	**0.0698**	**0.1743**	**0.3543**	**0.2032**	**0.3423**	**0.5494**	**0.0444**	**0.1216**	**0.2625**	**0.3400**	**0.5811**	**0.7247**	**0.4570**	**0.6950**	**0.8190**
A6	**0.4766**	**0.7360**	**0.8503**	**0.0712**	**0.1221**	**0.3187**	**0.0621**	**0.1525**	**0.3260**	**0.2159**	**0.4239**	**0.6919**	**0.0466**	**0.1302**	**0.2771**	**0.2164**	**0.4261**	**0.5876**	**0.4256**	**0.6473**	**0.7627**
A7	**0.5423**	**0.7942**	**0.8890**	**0.0661**	**0.1085**	**0.2807**	**0.1940**	**0.3703**	**0.5811**	**0.2793**	**0.5054**	**0.8140**	**0.0311**	**0.0825**	**0.1896**	**0.3091**	**0.5423**	**0.6855**	**0.4533**	**0.6893**	**0.8123**
A8	**0.4437**	**0.6973**	**0.8503**	**0.0687**	**0.1255**	**0.3490**	**0.1475**	**0.2723**	**0.4536**	**0.1524**	**0.3260**	**0.5901**	**0.0377**	**0.1042**	**0.2334**	**0.3400**	**0.5811**	**0.7443**	**0.4256**	**0.6473**	**0.7627**
A9	**0.1315**	**0.2712**	**0.4252**	**0.0153**	**0.0373**	**0.1442**	**0.1242**	**0.2723**	**0.4819**	**0.2540**	**0.4565**	**0.7326**	**0.0400**	**0.1172**	**0.2552**	**0.2318**	**0.4842**	**0.6660**	**0.4322**	**0.6573**	**0.7746**
A10	**0.0986**	**0.2324**	**0.4058**	**0.0102**	**0.0339**	**0.1442**	**0.0854**	**0.2069**	**0.3969**	**0.1270**	**0.2934**	**0.5494**	**0.0488**	**0.1302**	**0.2771**	**0.2164**	**0.3680**	**0.5093**	**0.4256**	**0.6473**	**0.7627**
Weight	**(0.756,0.891,0.889)**			**(0.117,0.156,0.349)**			**(0.357,0.501,0.652)**			**(0.546,0.701,0.875)**			**(0.091,0.178,0.299)**			**(0.711,0.891,0.901)**			**(0.457,0.695,0.819)**		

Step 4

The maximum and minimum of each column are determined as FPIS and FNIS respectively as shown in Table7.

Table 7 Max and min of each column of alternative design

Q1	A_{max}	**0.8503**	**0.8503**	**0.8890**
	A_{min}	**0.0493**	**0.0986**	**0.1162**
Q2	A_{max}	**0.2959**	**0.3187**	**0.3490**
	A_{min}	**0.0585**	**0.0102**	**0.0153**
Q3	A_{max}	**0.5528**	**0.5811**	**0.6520**
	A_{min}	**0.0621**	**0.0698**	**0.0854**
Q4	A_{max}	**0.8140**	**0.8140**	**0.8750**
	A_{min}	**0.3047**	**0.1016**	**0.1270**
Q5	A_{max}	**0.2771**	**0.2771**	**0.2990**
	A_{min}	**0.0266**	**0.0311**	**0.0333**
Q6	A_{max}	**0.7443**	**0.7748**	**0.9010**
	A_{min}	**0.1700**	**0.2164**	**0.2164**
Q7	A_{max}	**0.3926**	**0.4130**	**0.4256**
	A_{min}	**0.8123**	**0.8123**	**0.8190**

Step 5

The distance between the each alternative with the positive ideal solution and the negative ideal solution are calculated and are displayed in Table 8 and Table 9 respectively.

Table 8 The distance of each A_i(i = 1,2,3) from A_{max}

Quality	**A 1**	**A 2**	**A 3**	**A 4**	**A 5**	**A 6**	**A 7**	**A 8**	**A 9**	**A 10**
Q_1	**0.1449**	**0.1255**	**0.1081**	**0.5196**	**1.0692**	**0.0415**	**0.0164**	**0.0624**	**0.6806**	**0.7665**
Q_2	**0.0781**	**0.0975**	**0.0836**	**0.0733**	**0.0947**	**0.0674**	**0.0784**	**0.0641**	**0.1487**	**0.1522**
Q_3	**0.0492**	**0.1800**	**0.1339**	**0.1042**	**0.3280**	**0.3623**	**0.0947**	**0.1923**	**0.1926**	**0.2792**
Q_4	**0.1817**	**0.1456**	**0.1139**	**0.7170**	**0.4480**	**0.3122**	**0.1969**	**0.4763**	**0.2613**	**0.5414**
Q_5	**0.0665**	**0.0674**	**0.0527**	**0.0332**	**0.0471**	**0.0422**	**0.0737**	**0.0579**	**0.0499**	**0.0420**
Q_6	**0.1526**	**0.0061**	**0.1325**	**0.3591**	**0.0985**	**0.2729**	**0.1343**	**0.0971**	**0.1990**	**0.3562**
Q_7	**0.0631**	**0.0863**	**0.1430**	**0.1163**	**0.1490**	**0.1025**	**0.1430**	**0.1025**	**0.1115**	**0.1025**

Table 9 The distance of each (i = 1,2,3) from A_{min}

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Q_1	**0.4957**	**0.5342**	**0.5756**	**0.1304**	**0.0029**	**0.7831**	**0.9304**	**0.7024**	**0.0661**	**0.0433**
Q_2	**0.0281**	**0.0187**	**0.0248**	**0.0302**	**0.0197**	**0.0353**	**0.0272**	**0.0394**	**0.0040**	**0.0037**
Q_3	**0.2749**	**0.0997**	**0.1400**	**0.1743**	**0.0282**	**0.0194**	**0.1889**	**0.0899**	**0.0914**	**0.0449**
Q_4	**0.3250**	**0.3794**	**0.4412**	**0.0306**	**0.1081**	**0.1947**	**0.3135**	**0.0999**	**0.2392**	**0.0745**
Q_5	**0.0109**	**0.0100**	**0.0172**	**0.0328**	**0.0210**	**0.0247**	**0.0077**	**0.0144**	**0.0191**	**0.0248**
Q_6	**0.1916**	**0.5981**	**0.2141**	**0.0525**	**0.2626**	**0.0924**	**0.2109**	**0.2662**	**0.1467**	**0.0514**
Q_7	**0.1073**	**0.0818**	**0.0421**	**0.0578**	**0.0392**	**0.0678**	**0.0421**	**0.0678**	**0.0611**	**0.0678**

Step 6

The closeness coefficients, of the candidate architectures are displayed in Table 10. According to the closeness coefficient, the preference order of the alternatives is ranked. The best selection that satisfies the maximum benefit with minimum cost is Silverstripe (A7).

Table 10 Computations of L_i^+, L_i^-, CC_i

Design Alternatives	L_i^+	L_i^-	CC_i	RANK
Wordpress	**0.78019**	**1.38935**	**0.64039**	**4**
Joomala	**0.70381**	**1.72635**	**0.71039**	**2**
Drupal	**0.66680**	**1.55595**	**0.70001**	**3**
Expression	**1.86414**	**0.56723**	**0.23330**	**8**
TextPattern	**2.12477**	**0.59146**	**0.21775**	**9**
Contao	**1.16634**	**1.25207**	**0.51773**	**6**
Silverstripe	**0.63640**	**1.82172**	**0.74110**	**1**
Umbraco	**1.01795**	**1.31472**	**0.56361**	**5**
Concrete5	**1.59315**	**0.67801**	**0.29853**	**7**
Django	**2.20537**	**0.34525**	**0.13536**	**10**

VI. Conclusion

The quality of the software architecture mainly depends on the architects' experiences and the decision making abilities. Architectures that exhibit good trade-off among multiple quality requirements without exceeding available capital investments are recognized as a critical issue for which the decision maker needs to consider several aspects. Quality cannot be added to the system as an afterthought, it must be built into the system from the beginning. The point of maximum quality attainment for the minimum amount of investment is exactly the point of interest to the software manager. When crisp data is inadequate to model the real life situations, the decision makers use linguistic variables. The proposed method simulates the uncertain judgments with the meta-heuristic approach like Genetic Algorithmic is proposed for deriving priorities from fuzzy pairwise comparison judgments. The proposed method of deriving priorities considers judgments represented by both triangular fuzzy numbers. Following the FGA, an improved Fuzzy TOPSIS technique is used to cumulate the ratings and produce an overall performance score in selecting each alternative. As regards, a fuzzy number is greater than or equal to another fuzzy number, a new method was proposed in calculating the Fuzzy Positive Ideal Solution (FPIS) and Fuzzy negative Ideal Solution (FNIS). The case study validates the suitability and usefulness of the proposed framework.

References

1. Bass, L., Clements, P., Kazman, R. “Software Architecture in Practice.” SEI Series in Software Engineering. Addison-Wesley, Reading (1998).
2. Roy, B., Graham, T.C.N. “Methods for Evaluating Software Architecture: A Survey.” p. 82, School of Computing TR 2008-545, Queen’s University (2008).
3. Clements, Paul, Rick Kazman, Mark Klein. “Evaluating Software Architectures: Methods and Case Studies.” Addison-Wesley Professional; ISBN 0-201-70482X (2002).
4. G. Abowd, L. Bass, P. Clements, R. Kazman, L. Northrop, and A. Zaremski, "Recommanded Best Industrial Practice for Software Architecture Evaluation," SEI, Carnegie Mellon University CMU/SEI-96-TR-025, (1997).
5. Rick Kazman, Len Bass, Gregory Abowd and Mike Webb, “SAAM: A Method for Analyzing the Properties Software Architectures,” Proceedings of the 16th International Conference on Software Engineering, pp. 81-90, http://www.sei.cmu.edu/ata/publications. html #reports, (1994).
6. R. Kazman, M. Klein and P. Clements, “ATAM: Method for Architecture Evaluation,” Technical report, http://www.sei.cmu.edu/ata/ ata_method.html.
7. “CBAM: Cost Benefit Analysis Method,” Technical report, http://www.sei. cmu. edu/ata/ products_services /cbam.html.
8. PO Bengtsson, “Architecture-Level Modifiability Analysis,” Ph.D. Thesis, Blekinge Institute of Technology, Dissertation series No 2002-2, (2002).
9. Thomas J. Dolan, “Architecture Assessment of Information-System Families,” Ph.D. Thesis, Eindhoven University of Technology, Feb. (2002).
10. Nord, Robert L., Mario R. Barbacci, Paul Clements, Rick Kazman, Mark Klein. “Integrating the Architecture Tradeoff Analysis Method (ATAM) with the cost benefit analysis method (CBAM).”

No. CMU/SEI-2003-TN-038. Carnegie-Mellon University Pittsburgh PA Software Engineering INST, (2003).

11. M. Svahnberg, C. Wohlin, L. Lundberg and M. Mattsson, “A Method for Understanding Quality Attributes in Software Architecture Structures,” Proceedings of the 14th International Conference in Software Engineering and Knowledge Engineering, pp. 819-826, July (2002).
12. Li, Min, and Yu Jun Ma. "Determination of Weight Values for Assessment Indexes of Engineering Materials Based on Fuzzy Pairwise Comparison Method." Advanced Materials Research 680 , pp. 109-112, (2013).
13. M.B. Javanbarg, C. Scawthorn, J. Kiyono, B. Shahbodaghkhan, Fuzzy AHP-based multicriteria decision making systems using particle swarm optimization, Expert System with Applications 39, pp. 960–966, (2012).
14. Calabrese, Armando, Roberta Costa, and Tamara Menichini. "Using Fuzzy AHP to manage Intellectual Capital assets: An application to the ICT service industry." Expert Systems with Applications 40, no. 9 pp. 3747-3755, (2013).
15. Yue, Zhongliang. "TOPSIS-based group decision-making methodology in intuitionistic fuzzy setting." Information Sciences 277, pp. 141-153, (2014).
16. Aleti, B. Buhnova, L. Grunske, A. Koziolek, and I. Meedeniya, “Software Architecture Optimization Methods: A Systematic Literature Review,” IEEE Transactions on Software Engineering, vol. 39, no. 5, pp. 658-683, May (2013).
17. Dhaya, C., and G. Zayaraz. "Fuzzy based Quantitative Evaluation of Architectures using Architectural Knowledge." International Journal of Advanced Science and Technology (2012).
18. D. L. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, (1989).
19. Hwang C. L., Yoon, K. “Multiple Attribute Decision Making: Methods and Applications.” Springer-Verlag, New York, (1981).
20. Ertugrul, _I, Karakasoglu, N. “Performance evaluation of Turkish cement firms with fuzzy analytic hierarchy process and TOPSIS methods.” Expert Systems with Applications 36, no. 1, 702-715, (2009).
21. Wang, Ying-Ming, Taha Elhag. “Fuzzy TOPSIS method based on alpha level sets with an application to bridge risk assessment.” Expert systems with applications. 31, no. 2,pp. 309-319, (2006).
22. David Plans Casal. “Advanced Software Development for Web Applications.” Technical Report TSW0505, JISC Technology and Standards Watch, (2005).