IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011
ISSN (Online): 1694-0814
www.IJCSI.org

A Quality Based Method to Analyze Software Architectures

Farzaneh Hoseini Jabali[1], Sayed Mehran Sharafi[2] and Kamran Zamanifar3

[1]Computer Department, Islamic Azad University, Najafabad Branch, Young Researcher Club
Vilashahr, Najafabad, Isfahan, 8581643771, Iran

[2] Computer Department, Islamic Azad University, Najafabad Branch
Isfahan, Iran

[3] Computer Department, Islamic Azad University, Najafabad Branch
Isfahan, Iran

Abstract

In order to produce and develop a software system, it is necessary to have a method of choosing a suitable software architecture which satisfies the required quality attributes and maintains a trade-off between sometimes conflicting ones. Each software architecture includes a set of design decisions for each of which there are various alternatives, satisfying the quality attributes differently. At the same time various stakeholders with various quality goals participate in decision-making. In this paper a numerical method is proposed that based on the quality attributes selects the suitable software architecture for a certain software.

In this method, for each design decision, different alternatives are compared in view of a certain quality attribute, and the other way around. Multi-criteria decision-making methods are used and, at the same time, time and cost constraints are considered in decision-making, too. The proposed method applies the stakeholders' opinions in decision-making according to the degree of their importance and helps the architect to select the best software architecture with more certainty.

***Key words**: Software Architecture Evaluation, Quality Attribute, Stakeholder, Design Decision*

1-Introduction

Software systems are analyzed, designed and implemented in order for various problems to be solved and information and data to be processed. Today, as problems are more complicated, the number of components of the software systems is increased and the structure of these components, systems organization, and change and development in these systems has become more complicated. Hence there is no choice but to have a clear and intelligible software architecture. Architecture, in which quality attributes can be pursued, is the first stage in software production. Quality attributes are to be considered in all the stages of design, implementation and transference; therefore, in the case that it is supported by the architecture, it can be pursued more easily. In designing a software, various stakeholders with different quality goals should be considered when sometimes different quality goals are in opposition with each other. Hence there should be chosen an architecture that, while maintaining a trade-off between quality attributes and considering constraints, seeks to realize stakeholders' goals as far as possible. Therefore, it is especially important to evaluate software architecture according to quality attributes in order to make sure that the resulting software satisfies all of the stakeholders' requirements as far as possible. In this paper a numerical method is proposed that chooses the suitable architecture for certain software based on quality attributes.

The method proposed in this paper has three positive points all at the same time:

1) Evaluation of fine-grained software architecture
2) Uncertainty estimation in the resulting data
3) Using more adequate methods in order to consider the importance of participants' opinions in decision-making.

In section “2” the related works will be reviewed. In section “3” the proposed method will be presented. In section “4” a case will be reviewed using the proposed method and finally section “5” will be allocated to conclusions.

2-Related works

Valuable steps are taken regarding software architecture. Some articles analyses kinds of quality or non-functional requirements. For example in [2], in addition to discussing and stating the way to find quality attributes, the paper has investigated the mechanisms to prioritize attributes. Also, according to researches, some quality attributes are in opposition with each other. For example there is conflict between performance and modifiability and also between each quality attribute and the cost [4]. The steps taken in the field of software architecture evaluation can be divided into two groups: The first group comprises the ways of evaluating software architecture according to only one attribute. For example the methods in [8] and [7] review and evaluate architecture with regard to performance and modifiability respectively. The second group comprises the ways to evaluate architecture with regard to the trade-off between different quality attributes. For example, in [5] a method is presented for selecting the most suitable software architecture from alternative software architectures. By prioritizing quality attributes through AHP method and applying it to the presented architectures, the numerical results are derived for decision-making. Also in [9] there is proposed a method named Archdesigner which, in addition to prioritizing quality attributes, allocates a weight to different design decisions and chooses the most suitable architecture through the numerical method. In [6], a probabilistic method is presented for selecting the most suitable software architecture form the presented alternative architectures. Here, after calculating the density of value vectors, the architecture having the highest density is chosen as the best one.

3-The proposed method

In this method it is tried to use exact data. In order to achieve this, 3 actions are followed: the first is the evaluation of software architectures at the fine-grained level (different alternatives of design decisions). Architecture evaluation with regard to the level of quality provision is complicated but the evaluation of its components at the fine-grained level is simpler. The second is adjusting the estimated amount through the method in [5] and the third is calculating the uncertainty in the resulting data and re-estimating data if the uncertainty is high.
All stages of this method are displayed in the flowchart fig. 1. The description of each stage follows.

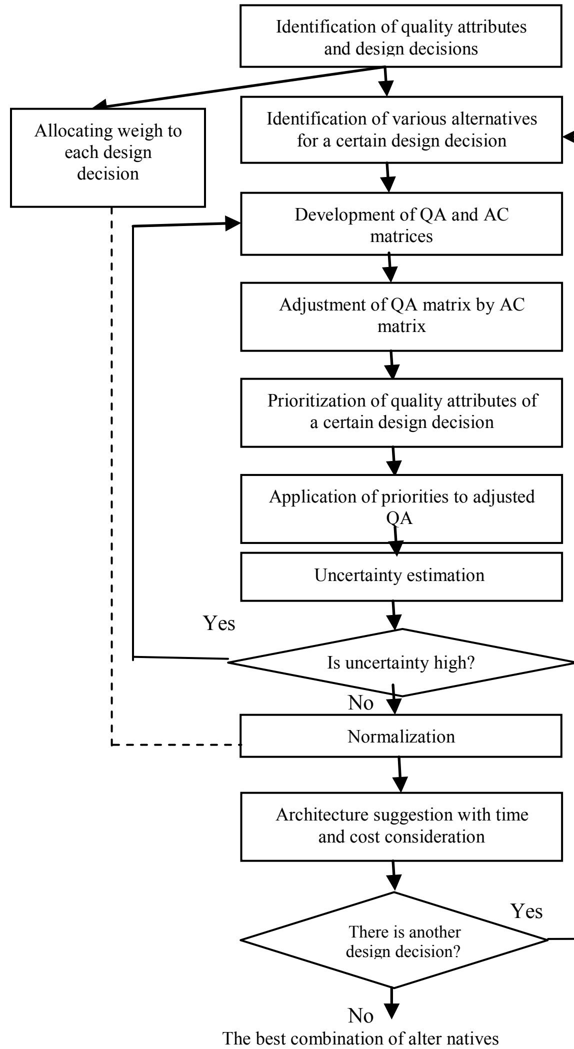

Fig. 1: Stages of the above method

1) *Identification of quality attributes and design decision*

In this stage the stakeholders' quality requirements that must be satisfied by a certain software, are identified and introduced. In [2], the method for finding the stakeholder`s quality requirements is investigated. At the same time, it is better that all the introduced quality attributes be in the same level of granularity. Also the design decisions on which a certain software will be are be introduced by the architect.

2) *Identification of various alternatives for each design decision*

In this stage, the suitable and available alternatives for each design decision must be identified and introduced as accepted alternatives for it. Also the characteristics of each of these alternatives must be identified clearly and explained for all those participating in decision-making.

3) *A relative comparison between various alternatives in view of quality attributes' being provided for each design decision*

In this stage for each design decision the ability of various alternatives for satisfaction of quality attributes is compared with each other and a numerical value is attributed to each of the alternatives in the provision of each quality attribute. For this purpose the MADM [3] functions can be used. In this paper we use "AHP [10] (Analytic Hierarchy Process)" method. This must be done by each member of the development team through different evaluation methods acceptable for them. As a result, an "Individual QA" matrix will be presented by each member of development team for each design decision, the rows and columns of which are the quality attributes and the alternatives introduced for that design decision respectively. Finally the average of the resulting "Individual QA"s will be derived through "Group decision making [3]" method and the QA matrix is obtained. Using "Group decision making" method enables one to allocate different importance degrees to different members of development team. It means that the one who has a more important opinion will get a higher degree and his opinion will be considered more in the decision-making. Also the sum in each row must equal 1. Otherwise, each row should be normalized. An example of a normal QA with 4 alternatives and 4 quality attributes is displayed in table "1".

Table 1: An example of a normal QA with 4 alternatives and 4 quality attributes

	C1	C2	C3	C4	Sum
Q1	QA1,1	QA1,2	QA1,3	QA1,4	1
Q2	QA2,1	QA2,2	QA2,3	QA2,4	1
Q3	QA3.1	QA3,2	QA3,3	QA3,4	1
Q4	QA4,1	QA4,2	QA4,3	QA4,4	1

4) *A relative comparison between quality attributes in view of alternatives' being provided for each design decision (AC matrix)*

This stage is similar to the previous one, but there is just a difference. For each design decision the quality attributes are compared with each other in view of satisfaction by every certain alternative. In this stage, too, the "Individual AC" matrices are calculated by development team and then the average is determined using the "Group decision making" method and the "AC" matrix is derived. Also the sum in each column must equal 1. Otherwise, each column should be normalized. An example of a normalized AC with 4 alternatives and 4 quality attributes is displayed in table "2".

Table 2: An example of a normal AC with 4 alternatives and 4 quality attributes

	C1	C2	C3	C4
Q1	AC1,1	AC 1,2	AC 1,3	AC 1,4
Q2	AC 2,1	AC 2,2	AC 2,3	AC 2,4
Q3	AC 3,1	AC 3,2	AC 3,3	AC 3,4
Q4	AC 4,1	AC 4,2	AC 4,3	AC 4,4
Sum	1	1	1	1

5) *Adjustment of QA matrix by AC matrix for each design decision*

In 3rd and 4th stages a comparison between columns (QA matrix) and rows (AC matrix) were made respectively. Both comparisons are in fact the same action with different perspectives which causes an increase in the quality of the act of estimating values. For each design decision, QA matrix will be adjusted by AC matrix if the values of QA are inconsistent with those of AC and finally the QAO matrix (optimal QA) will be derived.

Table3: A hypothetical QA

	C1	C2
Q1	0/6	0/4
Q2	0/3	0/7

Table 4: A hypothetical AC

	C1	C2
Q1	0/5	0/6
Q2	0/5	0/4

For example, tables 3 and 4 are hypothetical QA and AC matrices for a typical design decision respectively. For example, $p_{1,1}$ is the ability of alternative C_1 in satisfying the quality attribute Q_1. As can be seen, in the first column of AC, $P_{1,1} = P_{2,1}$ (1). Also in QA, $P_{1,1} = {}^{3p_{1,2}}\!/_{2}$ (2) and $P_{2,1} = {}^{3p_{2,2}}\!/_{7}$ (3). Combining equations 1, 2, and 3 leads to $7P_{1,2} = 2P_{2,2}$ (4) which, as can be seen, does not hold true in AC. Hence, because of the discrepancy between QA and AC in the above example, taken that both of the matrices have the same value, QA gets adjusted by AC. To calculate the optimal QA (QAO), initially k times QA' should be calculated by the following method. "K" is number of quality attributes.

$$QA'_{i,j} = \frac{QA_{1,j}AC_{j,j}}{AC_{1,j}} \quad \ldots\ldots\ldots \quad QA'_{i,j} = \frac{QA_{k,j}AC_{i,j}}{AC_{k,j}} \qquad (5)$$

Finally, QAO will be obtained by adding together k times different QA' and k times same QA and calculating the average.

6) *Prioritization of quality requirements*

In this stage, the importance degree of the quality attributes must be determined quantitatively by each of the stakeholders. For this purpose, the "AHP" method is used in which each pair of quality requirements is compared and finally a value which is the importance degree of a quality attribute from the viewpoint of that stakeholder is allocated to it. Then all of the related weights for each quality requirement which are allocated by the stakeholders will be used to get the final weight of each quality attribute through the "Group Decision Making" method. By using this method which is proposed in this paper, the importance degree of each stakeholder' opinion is considered in calculating the final weight of each quality attribute.
An example of a priority matrix (PQA) with 4 quality attributes is displayed in table "5".

Table5. An example of a PQA matrix with 4 quality attributes

QUALITY ATTRIBUTE	PERIORITY
QA1	P1
QA2	P2
QA3	P3
QA4	P4

Analytical Hierarchy Process (AHP)

The analytic hierarchy process as developed by Thomas L. Saaty is designed to help in solving complex multi criteria decision problems [2]. Looking at software engineering AHP can be used when prioritizing multiple criteria/attributes, e.g. prioritizing features or quality attributes like usability and performance.

AHP uses scaled pair-wise comparisons between variables, as illustrated in Figure 2, where the variables are ***i*** and ***j*** and the scale between them denotes relative importance. The importance ratings can be seen in Table 6 below.

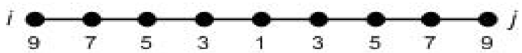

Figure 2: AHP Comparison Scale

As the variables have been compared the comparisons are transferred into an ***n x n*** matrix with their reciprocal values (***n*** is the number of variables). Subsequently the eigenvector of the matrix is computed. The method used for this is called *averaging over normalized column* and the product is the *priority vector,* which is the main output of using AHP for pair-wise comparisons.

AHP uses more comparisons than necessary, i.e. ***n (n – 1) / 2*** comparisons, and this is used for calculating the consistency of the comparisons. By looking at the *consistency ratio (CR)* an indication of the amount of inconsistent and contradictory comparisons can be obtained.

Table 6. AHP Comparison Scale

Relative intensity Definition		**Explanation**
1	Of equal importance	The two variables (i and j) are of equal importance.
3	Slightly more important	One variable is slightly more important than the other.
5	Highly more important	One variable is highly more important than the other.
7	Very highly more important	One variable is very highly more important than the other.
9	Extremely more important	One variable is extremely more important than the other.
2, 4, 6, 8	Intermediate values	Used when compromising between the other numbers.
Reciprocal	If variable *i* has one of the above numbers assigned to it when compared with variable *j*, then *j* has the value 1/number assigned to it when compared with *i*. More formally if *n* i j = *x* then *n* j i = *1/x*.	

7) Applying the priority of quality requirements

The output of 6th stage is the priority matrix (PQA), which contains the weight of quality attributes for the calculation of which the opinion of all stakeholders and the degree of importance of their opinion is considered. Finally, for each design decision, the priorities of quality requirements are applied to QAO by means of equation 6:

$$PQAO_j = \sum_{i=1}^{k} PQA_i QAO_{i,j} \qquad (6)$$

8) Calculation of Uncertainty

For each design decision, while deriving QAO through calculating the average of k times different QA' and k times same QA, the variance matrix (VC) is obtained. Then for each alternative of each design decision, variance is calculated through the equation 7. A high degree of variance shows that the calculated results of the previous stage aren't reliable and one shall return to the 3rd stage and do the calculations again with more accuracy.

$$\sum_{i=1}^{k} PQA_i^{\,2} VC_{i,j} \qquad (7)$$

9) Normalization

In this stage, the values which are allocated to various alternatives from stage 7 must be normalized. Because these values will be added together in the next stage, initially a weight is allocated to each design decision which shows the degree of its importance. It is natural for the more important design decisions to get higher weights. The allocated weight of z th design decision is shown by W_z. Hence for each design decision, the calculated results in stage 7 will be normalized by equation 8:

$$WP_j = PQAO_j \times W_z \qquad (8)$$

10) Selection of the suitable alternative

For each design decision, if passed through stage 8, the result obtained in stage 9 is used in order to choose and introduce the most suitable alternative for each design decision. In fact, the alternative that has the highest value in the equation 9 and doesn't violate time and cost limitations is introduced as the selected one. $WP_{i,j}$ denote WP for i th alternative of j th design decision.

$$\text{Maximize} \sum_{j=1}^{m} \sum_{i=1}^{nj} X_{i,j} WP_{i,j} \qquad (9)$$

$\forall j \in [1 ,...,m] : \sum_{i=1}^{nj} X_{i,j} = 1$

Cost $(x_{i1,1}, x_{i2,2}, \ldots, x_{im,m}) <=$ constraint cost

Time $(x_{i1,1}, x_{i2,2}, \ldots, x_{im,m}) <=$ constraint time

m≥1: the number of introduced design decisions for the software being studied

$N_j \geq 2$: the number of alternatives for j th design decision.

$X_{i,j} \in [0,1]$, where 1 shows alternative *I*'s being selected for design decision *j*, and 0 its not being selected

$V_{i,j}$ shows the normalized value score for alternative *i* of design decision *j*

4 –A case study

The Glass Box [1] project is a part of a research program that was begun in 2002 and in early 2003 its first version was put into operation successfully. GB is a software system which is used in the analysts' working environment. GB application is used by the user to get information from one's workstation during information gathering and analysis tasks. It is also a software for testing the platforms for the participants' research projects. Figure 3 depicts the relationship between the GB application and the various stakeholders involved.

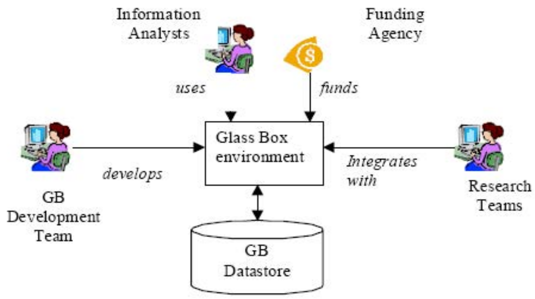

Figure 3: GLASS BOX Stakeholders

4-1. S*tages1 and 2: Identification of design decisions and quality attributes and v arious alternatives for each of them*

9 design decisions in GB project design are considered, that, because of limitations, only 3 design decisions shown in table 7 are studied. The information resulted from stages1 and 2 which include design decisions being studied and the related quality attributes and stakeholders are shown in table 7.

4.2. *Stage 3: A relative comparison between various alternatives in view of quality attributes' being provided for each design decision (QA matrix)*

In this stage, for each design decision, individual QAs will be calculated through AHP method by the development team. Then the final QA will be obtained by calculating the average of individual QAs by Group Decision Making method. Because of limitations, only QA of EVNT design decision is shown in table 8.

Table 8: QA matrix of EVNT design decision

Quality attributes	alternatives			
	JMS	*MSMQ*	*TRGR*	*COAB*
Reliability	0.475	0.225	0.38	0.52
Performance	0.21	0.363	0.235	0.315
Complexity of implementation	0.315	0.412	0.385	0.165

4.3. *Stage 4: A relative comparison between quality attributes in view of alternatives' being provided for each design decision (AC matrix)*

Table 7: Selected decisions being studied, quality attributes, and stakeholders

Design decision	alternatives	Quality attributes	stakeholders
Architecture (ARCH)	• 3-tier using J2EE (THTJ) • 3-tier using .Net (THTD) • 2-tier (TWOT) • COABS (COAB)	• Modifiability • Scalability • Performance • Cost •Development effort • Portability • Ease of installation	• Development team • Research Teams • Funding agency
Event Notification (EVNT)	• Publish-Subscribe using JMS (JMS) •Publish-Subscribe using MSMQ (MSMQ) •Database triggers (TRGR) •COABS (COAB)	• Reliability • Performance •Complexity of implementation	• Development team • Research Teams
Authentication (AUTH)	• Database-based security (DB) • J2EE-based security (J2EE) •.Net-based security (.NET) •COABS (COAB)	• Complexity of implementation • Ease of deployment and setup	• Development team • Research Teams

In this stage, similar to the earlier one, individual ACs will be calculated through AHP method by development team for each design decision. Then the final AC will be obtained by calculating the average of individual ACs by Group Decision Making method. Because of limitations, only AC for EVNT design decision is shown in table 9.

Table 9: AC matrix of EVNT design decision

Quality attributes	alternatives			
	JMS	*MSMQ*	*TRGR*	*COAB*
Reliability	0.358	0.277	0.123	0.242
Performance	0.179	0.204	0.335	0.282
Complexity of implementation	0.371	0.163	0.144	0.322

4.4. *Stage 5: Adjusting the QA matrix by AC matrix for each design decision*

Because of inconsistency between QA and AC of EVNT design decision that is shown in tables 6 and 7, the values of QA will be improved by AC. Optimal QA (QAO) for EVNT design decision is shown in table 10.

Table 10: QAO matrix of EVNT design decision

Quality attributes	alternatives			
	JMS	*MSMQ*	*TRGR*	*COAB*
Reliability	0.402	0.4	0.056	0.142
Performance	0.16	0.126	0.478	0.227
Complexity of implementation	0.45	0.05	0.05	0.45

4.5. *Stages 6 and 7: Prioritizing the quality attributes and applying priorities to QA*

In this stage, for each design decision the architect will ask various stakeholders to prioritize quality attributes. In this paper, because of limitations, only priorities of EVNT design decision is shown in table 11. In this example the development team's opinion is twice as importance as the research team's; this being considered in calculating the average.

Table 11: The weight allocated to quality attributes by the stakeholders of EVNT design decision

Quality attributes	Development Team	Research Team	Aggregated
Reliability	0.26	0.335	0.285
Performance	0.318	0.418	0.351
Complexity of implementation	0.422	0.247	0.364

Finally the calculated priority is applied to QAO obtained in stage5. The result obtained from applying priority of EVNT design decision is shown in table 12.

Table 12: Alternatives' values of EVNT design decision

alternatives	JMS	MSMQ	TRGR	COAB
values	0.3	0.22	0.2	0.28

4.6. *Stage 8: Uncertainty estimation*

In this stage, a variance matrix (VC) will be calculated for each design decision, then the variance of alternatives of each design decision will be calculated. In this paper only variance matrix of EVNT design decision is shown in table 13 and the variance of alternatives of the design decision being studied is shown in table 12. As can be seen, the alternatives' variance of EVNT design decision is reasonable and there is no need to do the calculation again.

Table 13: Variance matrix of EVNT design decision

Quality attributes	Alternatives			
	JMS	*MSMQ*	*TRGR*	*COAB*
Reliability	0.005	0.036	0.033	0.042
Performance	0.001	0.057	0.053	0.044
Complexity of implementation	0.015	0.061	0.053	0.04

4.7. *Stage 9: normalization*

The weight allocated to design decisions of GB project is shown in figure 4. The weight allocated to each design decisions is applied to PQAO matrix of the same design decision and normal PQAO (WP) will be obtained. The WP matrix of EVNT design decision is shown in table 12.

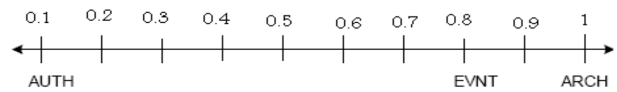

Figure4: Weights allocated to design decisions of GB project

4.8. *Stage 10: Selection of the suitable alternative*

In this stage, it is necessary to know time and cost constraints of GB project. According to the given information this project must be finished in less than 12 months. Calculating the exact necessary time for each alternative is difficult. But determining the alternatives that violate the constraints is simple. In this case the J2EE alternative of AHUT design decision violates the time constraint. Hence according to table 14 the colored alternatives are selected for the intended design decision.

Table 14: Obtained alternatives' values and variances

Design decisions	alternatives	Normalized WP	variance
EVNT	JMS	0.24	0.002
	MSMQ	0.176	0.018
	TRGR	0.16	0.016
	COAB	0.224	0.014
AUTH	DB	0.035	0.03
	J2EE	0.017	0.0002
	.NET	0.025	0.012
	COAB	0.024	0.002
ARCH	THTJ	0.29	0.0004
	THTD	0.2	0.0007
	TWOT	0.28	0.0002
	COAB	0.23	0.0003

5-Conclusion and future works

The method proposed in this paper chooses the most suitable fine-grained alternative for each design decision, considering time and cost constrains. After combining these alternatives, the best architecture for a cer tain software is chosen. The proposed method attempts to increase accuracy in choosing the suitable architecture through adjusting the estimated values and prevents mistakes by estimating the degree of uncertainty in the results. The limitation is the use of AHP method for comparison which is proposed as an area of further research. Also a software can be developed in which this method is implemented.

The value scores of alternatives that related to studied design decisions are shown in figure 4.

According to figure 5, DB and THTJ and JMS alternatives have the most value scores for AUTH and ARCH and EVNT design decisions respectively.

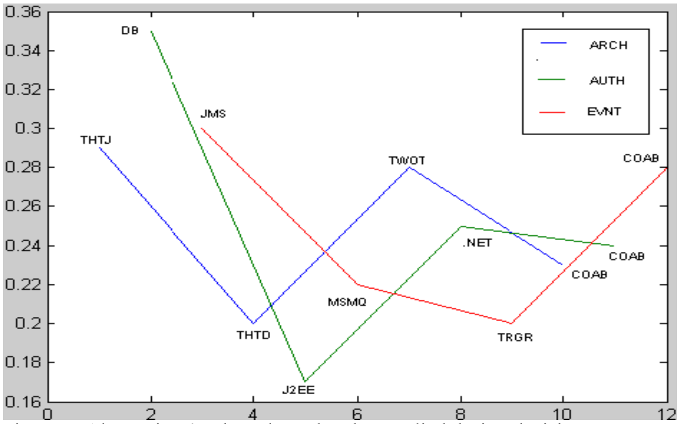

Figure 5. Alternatives' values that related to studied design decisions

6-Refrences

[1] I.Gorton and J.Haack *Architecturing in the Face of Uncertainty: An Experience Report.Proce.International Conference on Software Engineering..Edinburgh, Scotland, 2004.*

[2] J.A McCall, "Quality factors",in Encyclopedia of Software Engineering, J.I. Marciniak (ed),John Wiley &Sons,New York, PP. 958-969,1994.

[3] K.P. Yoon and C. Hwang, *Multiple Attribute Decision Making: An Introduction*: Sage Publications,1995

[4] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*. 2 ed: Addison-Wesley,2003

[5] M. Svahnberg, C. Wholin, and L. Lundberg, A Quality-Driven Decision-Support Method for Identifying Software *Architecture Candidates.* Int. Journal of Software Engineering and Knowledge Engineering, 13(5): p. 547-573, 2003.

[6] M.vafaei, F.shams, S.setayesh" A Probabilistic Method of Design Decision Evaluation of Value for Software Architecture Trade-off Analysis"12th International Conference of Computer Company of Iran, 2006.

[7] Bengtsson, PO., Architecture-Level Modifiability Analysis, ISBN 91-7295-007-2, Blekinge Institute of Technology, Dissertation SeriesNo2002-2, 2002.

[8] Sayed Mehran Sharafi," Extending Team Automata to Evaluate Software Architectural Design". COMPSAC: 393-400, 2008.

[9] T.Al-Neem, I.Gorton, M.A.Babar, F.Rabhi and B.Benatalah, "A quality-driven Systematic Approach for Architecting Distributed SoftwareApplications", In Proceedings of the 27th International Conference on Software Engineering (ICSE), st.louis, USA, 2005.

[10] T. L. Saaty, *"The Analytic Hierarchy Process"*, McGraw Hill, Inc., New York NY, 1980.