ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 3 (2011) 1480–1484

Procedia Computer Science

www.elsevier.com/locate/procedia

WCIT 2010

A Quantitative Algorithm to Select Software Architecture by Trade-off between Quality Attributes

Farzaneh Hoseini jabali[a] ,Sayed Mehran Sharafi[b], Kamran Zamanifar[b] a*

[a] *Computer Department, Islamic Azad University, Najafabad branch, MSc student ,Young Research Club , Esfahan, Iran*
[b]*Computer Department, Islamic Azad University, Najafabad branch, Esfahan, 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 selects the suitable software architecture for certain software according to quality attributes.
In this method, for each design decision, different alternatives for a specific software quality attribute must be compared and also 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. 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.

Selection and/or peer-review under responsibility of the Guest Editor.

Key words: Software Architecture Evaluation, Quality Attribute, Stakeholder, Trade-off, Design Decision

1-Introduction:

Analysis, design and implementation of software systems are done in order to solve various problems and process information and data. 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 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 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

* Corresponding author. Tel.:+983312250729; fax: +983312250729.
E-mail address: hoseini.farzaneh@sco.iaun.ac.ir, mehran_sharafi@iaun.ac.ir, zamanifar@eng.ui.ac.ir

doi:10.1016/j.procs.2011.01.035

resulting software satisfies all of the stakeholders' requirements as far as possible. In this paper a numerical method will be 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) Uncertainly estimation in the resulting data
3) Using more adequate methods in order to consider the importance of participants' opinions in the 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 analyze kinds of quality or non-functional requirements. For example in [1], 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 [3]. 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 [7] and [6] 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 [4] 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 [8] 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 [5], 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 architecture.

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 [4] and the third is calculating the uncertainly in the resulting data and re-estimating data if the uncertainly is high.
All stages of this method are displayed in the flowchart fig. 1. The description of each stage follows.

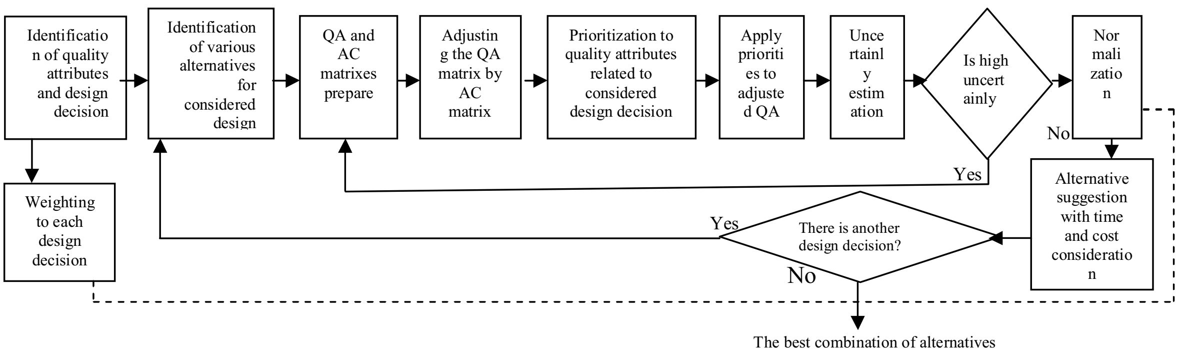

Fig 1: stage of mentioned method

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 certain software will be based will be introduced by 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 satisfaction of quality attributes for each design decision (QA matrix)

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 [2] functions can be used. In this paper we use "AHP [9] (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 [2]" 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 regarding satisfaction by each of the alternatives 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 estimating values. For each design decision, QA matrix will be adjusted by AC matrix if the values of QA are discordant with those of AC and finally the QAO matrix (optimal QA) will be derived.

Table3: hypothetical QA

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

Table 4: 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.

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

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 "AHP" method, 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.

7) *Applying the priority of quality requirements*

The output of 6^{th} 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:

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

8) *Calculation of Uncertainly*

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 3^{rd} stage and do the calculations again with more accuracy.

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

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:

$$WP_j = PQAO_j \times W_z \tag{8}$$

10) *Selection of a 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.

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

$\forall j \in [1,\ldots,m] : \sum^{nj}_{i=1} X_{i,j} = 1$

Cost ($x_{i1,1}$, $x_{i2,2}$, …., $x_{im,m}$) < = constraint cost
Time ($x_{i1,1}$, $x_{i2,2}$, …., $x_{im,m}$) < = 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 intimates that alternative *i* is selected for design decision *j*, and 0 denotes non selection of this alternative.
$WP_{i,j}$ intimates the normalized value score for alternative *i* of design decision *j*.

4-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 certain 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.

5-Refrences

1.J.A McCall, Quality factors ,in Encyclopedia of Software Engineering, J.I. Marciniak (ed),JohnWiley &Sons,New York, PP. 958-969,(1994).
2.K.P. Yoon and C. Hwang, Multiple Attribute Decision Making: An Introduction: Sage Publications,(1995).
3.L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice. 2 ed: Addison-Wesley,(2003).
4.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): pp. 547-573, (2003).
5.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).
6 ,PO. Bengtsso., Architecture-Level Modifiability Analysis, ISBN 91-7295-007-2, Blekinge Institute of Technology, Dissertation SeriesNo2002-2,(2002).
7.Sayed Mehran Sharafi, Extending Team Automata to Evaluate Software Architectural Design, COMPSAC: pp.393-400, (2008).
8.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).
9.T. L. Saaty, The Analytic Hierarchy Process, McGraw Hill, Inc., New York NY, (1980).