Using Multi-criteria Analysis to Handle Conflicts During Composition

Filipe Vieira
Dept. de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal
fil.vieira@gmail.com

Isabel Sofia Brito
Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Beja, Beja, Portugal
isabel.sofia@estig.ipbeja.pt

Ana Moreira
CITI/Dept. de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal
amm@di.fct.unl.pt

Abstract. *When two or more aspectual concerns with negative contributions are composed in a match point, conflicting situations might be triggered. This paper investigates the possibility of using Multiple Criteria Decision Making (MCDM) methods to support conflict resolution in the context of the AORE approach [2, 3]. We have chosen Analytic Hierarchy Process (AHP) created by Thomas Saaty [9] because: it is a well-known and accepted method; it is able to use information about our problem; it has the ability to quantify subjective judgments; it is capable of comparing alternatives in relation to established criteria; and it provides means to guarantee the consistency of the judgments. The final solution relies on the obtained concern rankings, which can be used to handle unresolved conflicts that might occur in a match point.*

1. Introduction

This paper builds on previous work on Aspect-Oriented Requirements Engineering (AORE) [2, 3] and addresses the well-known conflict management problem, also identified in other approaches (e.g., [6, 8]). Barbacci et al. [2] propose three main activities to support AORE: identify concerns, specify concerns and compose concerns. This paper concentrates on the composition activity, where conflicting situations may emerge in a given match point. A match point identifies specific locations in the base concerns where other concerns' behaviour (crosscutting or non-crosscutting) should be satisfied [3]. In this context, a conflict occurs any time two or more concerns that contribute negatively to each other, and have the same priority, need to be composed in the same match point. For example, consider the case where security and response time are applied in the same match point. It is well-known that these two concerns contribute negatively to each other, meaning that the developer may not be able to satisfy both with the same degree of satisfaction.

A solution to handling conflicts at the AORE level does not exist yet. The contribution of this paper is, therefore, to propose the use of AHP to support conflict management. There are several reasons why multi-criteria techniques are well suited to handle the kind of problems in hand. In particular, they help guarantee the logical consistency of many human-based judgments, as well as synthesizing the wide-range of data in a single solution. AHP offers the possibility to find, given a set of alternatives and a set of decision criteria, the best alternative.

This paper is organized as follows: Section 2 introduces an AORE model. Section 3 gives an overview of AHP approach. Section 4 shows the application of the AHP, in the context of the AORE approach, to a case study. Section 5 discusses related work. Finally, Section 6 summarizes our work and lists some open issues.

2. The AORE Model

The Aspect-Oriented Requirements Engineering (AORE) model [2, 3] defines three primary tasks, each one divided in several subtasks (see Fig. 1).

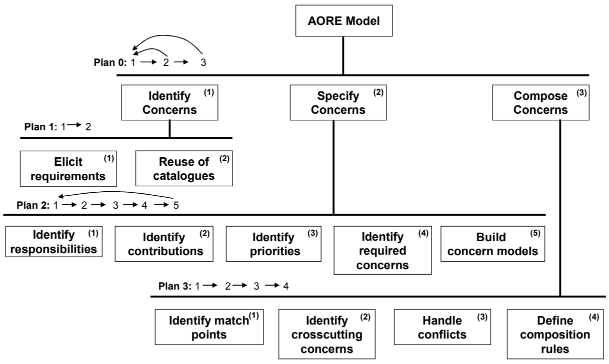

Fig. 1. The AORE model

The task, *identify concerns,* aims at identifying the concerns of a system, where a *concern* refers to a matter of interest which addresses a certain problem that is of interest to one or more stakeholders. Such a concern can be defined as a set of coherent requirements, defining a property that the future system must provide. This can be accomplished by analyzing the initial requirements, transcripts of stakeholders' interviews, etc. Good sources to guide concern identification are the existing catalogues, such as [1, 4].

The task, *specify concerns,* is composed of several subtasks whose main goal is to collect several types of information about a concern, store that information in a template and to build concern visual models (e.g., UML use case, interaction and class diagrams [11]). Stakeholders are a good source for (NFR) concern identification. Techniques such as the QAW method [7] can be used to elicit and specify such concern requirements.

Finally, the task *compose concerns*, offers the possibility to compose a set of concerns, incrementally, until the whole system is obtained. Each composition takes place in a match point in the form of a composition rule. A match point tells us which concerns (crosscutting or non-crosscutting) should be composed together. A composition rule shows how a set of concerns can be weaved together by means of some pre-defined operators. In order to accomplish this, we need to identify crosscutting concerns (those that are required by more than one other concern). At this point conflicting situations can be detected – whenever concerns that contribute negatively between them, have the same priority and need to be composed in the same match point. The AORE approach proposes a simple process to handle conflicts: intuitively allocate different priorities to conflicting concerns (for more information, see [2, 3]).

3. The AHP Method

AHP is a decision analysis method that ranks alternatives based on a number of criteria. This method provides a mathematically rigorous process for prioritization and decision-making. By reducing complex decisions to a series of pairwise comparisons and then synthesizing the results, decision-makers arrive at the best decision based on a clear rationale.

The AHP method can be described in 5 steps: problem definition, graphical representation of the problem, establishment of priorities, synthesis and consistency.

Step 1: Problem definition. In this step we define the problem goal (objective).

Step 2: Graphical representation of the problem. This step is divided into three sub-steps: state the objective (according to goal encountered in step 1), define the criteria and choose the alternatives. Fig. 2 illustrates an example of a hierarchical tree.

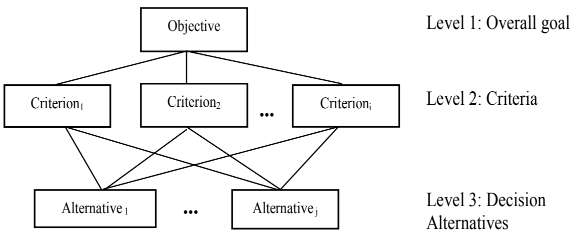

Fig. 2. AHP hierarchical tree

Step 3: Establish priorities. Pairwise comparisons are used to estimate the ratio of relative importance of two alternatives in terms of the criteria. To accomplish this, Saaty uses the scale [1..9] to rate the relative importance of one criterion over another, and constructs a matrix of the pairwise comparison ratings. If a criterion is less important than the other, then the inverse preference is rated in the scale 1, 1/2, 1/3, …, 1/9. The reason for this scale is based in psychological theories and experiments that points to a use of 9 unit scales as reasonable set that allows humans to perform discrimination between preferences for two items.

Step 4. Synthesis. In this step we calculate the priority of each alternative solution and criteria being compared. Several mathematical procedures can be used for synthetization, for example eigenvalues and eigenvectors [9]. However, a simple averaging solution respects better the symmetry of the problem and is much simpler to apply.

Step 5. Consistency. The quality of the decisions is guaranteed through a measure of consistency of pairwise comparison judgments by computing a consistency ratio (CR):

$$CR = \frac{CI}{RCI}$$

where CI represents the consistency index and RCI gives the random consistency index. RCI is an average random index derived from a sample of size 500 of randomly generated reciprocal matrices [10, pp. 59] and depends on the number of elements being compared (see Table 1).

CI is calculated using:

$$CI = (\lambda_{\max} - n)/(n-1)$$

where n is the number of items or alternatives being compared and λ_{max} is calculated using:

$$\frac{A \cdot W}{W} = CV$$

where A represents the pairwise comparison matrix, W the priority vector of alternatives with respect to a certain criteria and CV the consistency vector. The resulting vector of $A \cdot W$ is also known as the weighted sum vector. λ_{max} is obtained from the average of the sum of the values in vector CV.

CR is designed in such a way that values of the ratio exceeding 10% are indicative of inconsistent judgement; in such cases, the decision maker would probably want to revise the original values in the pairwise comparison matrix [12].

Table 1. RCI values per number of compared items

N	1	2	3	4	5	6	7	8	9
RCI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45

4. Applying AHP to the AORE Model

This section uses a running example to illustrate the application of the AHP method to the AORE model [2, 3]. The example we have chosen was inspired on the Washington subway system.

4.1 The Case Study

To use subway each client must posses a card in which was credited a certain amount of money. Each card can be bought and credited at selling machines that are available at subway stations. To start a journey, passenger should use his card on an entrance machine. This machine validates his card and allows passengers having a valid title to initiate a trip. In a similar way, when a passenger ends his journey, he inserts the card in an exit machine. At this moment the price to pay is calculated and debited from card. The price will depend on the distance travelled. If the card does not have enough credit to pay the trip, then s/he must credit a certain amount at any of the available machines on the subway. The client can request for a refund of the money in his/her card, by using a selling machine.

As a result of the AORE application on this case study, the following concerns were identified:

- Non-functional concerns: Accuracy, Compatibility, Response Time, Availability, Integrity, Multi-Access and Fault tolerance.
- Functional concerns: Enter Subway, Validate Card, Buy Card, Exist Subway, Refund Card, Credit Card and Calculate Statistics.

To illustrate the use of the AHP process, we will focus on a single match point, the Enter subway. A template for Enter Subway specification is show in Table 2.

Table 2. Enter subway template

Name	**Enter Subway**
Sources	Stakeholders, set of initial requirements, knowledge of the system
Stakeholders	Passenger, System owner, Developer
Description	Is responsible for handling the initial part of a trip
Decomposition	<none>
Classification	Functional
List of Responsibilities	
Responsibilities	(1) Register an entrance (2) Return the card to the client
List of Contributions	
<none>	
List of Priorities	
Stakeholder Priorities	Passenger: Very Important System owner: Very Important Developer: Very Important
List of Required Concerns	
Required Concerns	(1) Response Time (2) Accuracy (3) Integrity (4) Availability (5) Validate Card (6) Multi-access (7) Fault Tolerance

The contributions (positive and negative) between the required concerns, with respect to Enter Subway, are depicted in Fig. 3.

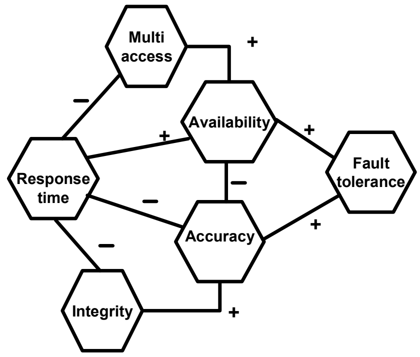

Fig. 3. Contributions between (non-functional) concerns

4.2 Applying the AHP Method

As described in Section 3, the AHP method is composed of five main steps. The result is a list of concerns, ordered by degree of importance in a match point.

Step 1: Problem definition
The goal is to rank the concerns required in the match point Enter Subway.

Step 2: Graphical representation of the problem
The alternatives represent different available options for the decision maker. Assuming *A* is a set of alternatives of a problem, our list is composed of all required concerns in the Enter Subway match point:

A = {Accuracy, Response time, Availability, Integrity, Multi-Access, Validate Card, Fault Tolerance}.

The criteria, also referred to as decision criteria, represent the different dimensions from which the alternatives can be analysed. For our problem, the criteria are a subset of the fields in the template (*Classification*, *Contribution* and *Priority*) and the *Type* criterion that states whether a concern is crosscutting. So, *C* is the set of criteria:

C = {Type, Classification, Contribution, Priority}.

By dividing the *Priority* criterion into sub-criteria it is possible to use stakeholder judgments as part of the decision problem. The main reason for the division is that the relation between concerns and priority is dependent of the judgement of each stakeholder regarding the variables that are influent in her/his perspective of the system. Therefore, consider $S_1 \ldots S_n$ to be the list of *n* stakeholders identified in the problem. The final set criteria, *C*, is:

$C = \{Type, Classification, Contribution, Priority_{S1}, Priority_{S2}, \ldots Priority_{Sn}\}$.

The chosen criteria are then used by the AHP method to rank the list of concerns in a given match point (see compose concerns task in AORE model). This ranking is therefore used as a means for solving conflicts and can also be used as a guide to define the composition rule.

With the identified objective, criteria and alternatives we can now build the hierarchical tree for our match point (see Fig. 4).

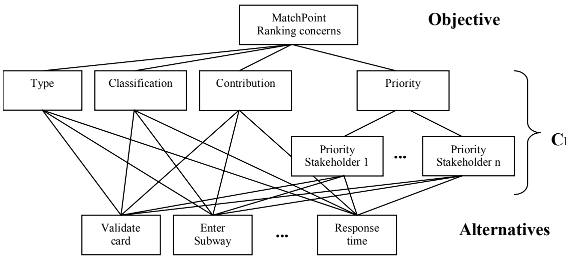

Fig. 4. Graphical representation of the enter subway match point.

Step 3: Establish priorities

AHP uses a pairwise comparison matrix to determine the ratings of judgements. During this process comparisons are performed to establish priority measures. Each alternative priority measure is obtained by comparing two alternatives in relation to a certain criteria. The comparison of two alternatives is quantified by using a [1..9] scale. For example, in Table 3, the pairwise comparison matrix for *priority* is divided into the three criteria *System Owner*, *Passenger* and *Developer*, where, in this example, System Owner is three times more important than Passenger and Developer is two times more important than Passenger.

Table 3. Pairwise comparison matrix for stakeholders in terms of priority criteria

Priority	Passenger	System Owner	Developer
Passenger	1	1/3	1/2
System Owner	3	1	2
Developer	2	1/2	1

The final task of this step is the creation of pairwise comparison matrixes that will represent decision maker estimated priorities for each alternative. In this problem there are seven alternatives to be evaluated with respect to each criterion. Based on a criterion, the decision maker repeatedly compares one alternative decision to another until all possible pairwise comparisons matrix are completed. An example for all alternatives based on *Developer Priority* criterion is show in Table 4.

Table 4. Pairwise comparison of alternatives based on criterion Developer Priority for last level in hierarchal tree

Priority (Developer)	Accuracy	Response Time	Availability	Integrity	Multi-Access	Validate card	Fault tolerance
Accuracy	1	2	1	1	1	1	1
Response time	1/2	1	1/2	1/2	1/2	1/2	1
Availability	1	2	1	1	1	1	1
Integrity	1	2	1	1	1	1	2
Multi-access	1	2	1	1	1	1	2
Validate card	1	2	1	1	1	1	2
Fault tolerance	1	1	1	1/2	1/2	1/2	1

For each matrix the number of required comparisons is *n*(n-1)/2* (where *n* is the number of alternatives). For our example, 2nd level computation (see Fig. 4), the resulting number of comparisons is 21 per matrix, in a total of 84 comparisons required for all four criterions (2nd level in Fig. 4). Also note that 3rd level computation adds more 63 comparisons.

Given the high total number of required comparisons, one might think that this process is excessively time consuming and, therefore, almost useless for large problems. Nevertheless, there are ways to avoid deriving priorities for alternatives by means of pairwise comparisons, for example by deriving relative weights from difference comparisons. This would reduce the number of comparisons to *n-1*; however, if an error is introduced in this process, a strong negative impact will be felt in the final result [10, pp. 86]. Fortunately, there are automatic tools, such as Expert Choice, Rank Master and Descriptor, to support the AHP process, facilitating its use and speeding up the calculation process.

Step 4: Synthesis

After creating all pairwise comparisons it is now possible to calculate the priority of each alternative in terms of each criterion. As described before, there are several ways of obtaining the synthesized priorities. In our case, we will use simple averaging. In this step we determine priority vectors with relative priorities for each alternative with respect to a certain criterion. First we must calculate the normalized pairwise comparison matrix by dividing each element of the comparison matrix by the sum of each column. Then we find the estimate relative priorities by calculating the average of each row of the normalized matrix. At the end of this step we have the priority matrix, where each column is a priority vector. Table 5 shows all priority vectors for each alternative in terms of each criterion[1].

Table 5. Priority matrix (each column is a priority vector for a criterion)

Criteria / Alternatives	Type	Classification	Contribution	Priority
Accuracy	0.143	0.067	0.071	0.134
Response Time	0.143	0.067	0.058	0.157
Availability	0.143	0.067	0.097	0.134
Integrity	0.143	0.067	0.149	0.138
Multi-access	0.143	0.067	0.168	0.182
Validate Card	0.143	0.600	0.288	0.139
Fault Tolerance	0.143	0.067	0.168	0.116

To be able to calculate the final ranking, we need to determine the criteria priority vector that will be multiplied by the priority matrix (Table 5). The criteria priority vector (Table 7) is obtained by normalizing the pairwise comparison matrix in Table 6 and calculating the row average.

---

[1] Due to lack of space, we only give a simple explanation of the calculations needed to obtain these values.

Table 6. Pairwise comparison matrix for the criteria in respect to goal

GOAL	Type	Classification	Contribution	Priority
Type	1	1	1/5	1/7
Classification	1	1	1/5	1/7
Contribution	5	5	1	1
Priority	7	7	1	1

Table 7. Criteria priority vector (2nd level)

Type	0,072
Classification	0,072
Contribution	0,392
Priority	0,464

The results in Table 7 show that according to the preferences established on Table 5, the most important criteria are *Priority* and *Contribution*. The remaining criteria are relatively less important in terms of the overall goal (which is to handle conflicts in the match point Enter Subway).

The computation to determine the final ranking can be performed by multiplying the priority matrix by the criteria priority vector. The final ranking is illustrated in Table 8.

Table 8. Final ranking for the Enter Subway match point

Alternative	Overall Priority	Rank
Validate Card	0.231	1
Multi-access	0.166	2
Integrity	0.137	3
Fault tolerance	0.135	4
Availability	0.115	5
Response Time	0.111	6
Accuracy	0.105	7

Therefore, for our Enter Subway match point (where negative contributions were found between several concerns, as shown in Fig. 3) we were able to rank all the potential conflicting concerns with different priorities. However, if this was not the case, a solution could be achieved either by changing the criteria or the stakeholder weights. In any of these cases, the process should be reapplied from Step 3.

This ranking can be used to support the architects and designers' solution choices and decisions. For example, this means that in the Enter SubWay match point the architecture design choice will be determined by the concerns ranked higher (at the top of Table 8) given less importance to those ranked lower down. This means that the importance of each concern in determining the type of architecture is proportionally greater the higher its rank in the table. This may mean, at the limit, that it may not be possible to satisfy some of the lower-ranked concerns at all.

Step 5: Consistency

AHP provides means to evaluate the consistency of the judgements that the decision maker demonstrated during the pairwise comparisons. This reduces any possible error that might have been introduced during the judgement process. This value is calculated automatically for all the pairwise comparison matrices[2]. In our case, the consistency rate obtained for the example in Table 4 was 0.01. This is a good indication that consistent judgments were made on pairwise comparisons.

4.3 Discussion of the Results

Based on the results obtained by applying the AHP method, it is possible to use rankings to solve conflicting situations that might occur in a match point. Based on the obtained rankings, decisions can be taken to handle unresolved conflicts (see discussion in Step 4).

The situations that can influence negatively the results of the application of the AHP method fall into two domains: problem and multi-criteria. From the problem domain, we should be able to guarantee that the questions below are answered during the initial activities of the AORE model:

- Have all the concerns been identified?
- Were all the stakeholders considered?
- How correct is the stakeholder knowledge about the problem?
- At what level of decomposition should AHP be applied?

In what concerns the MCDM domain, we should remember that:

- there may never be a single MCDM method that can ensure that the derived ranking of the alternatives is the correct one, and
- human judgements are not error free, even if the consistency level is below 10%, judgements are always subjective. Consistency only tells that the judgements are valid in a logical perspective. However, the reasoning used to achieve those values is subjective and may not reflect the best alternatives.

As the reader can appreciate, the application of the method can be time consuming and certainly error prone, mainly if all the calculations are done by hand. This technique has been proved useful in all the case studies we have solved so far. However, automatic tools should be used to reduce the overhead introduced in the AORE model. To facilitate the application of the AHP method, we have programmed it in Excel.

At the time this paper is being written, we are experimenting the Expert Choice tool as a means to computerize our final concern ranking. The comparison has not been done in detail, since Expert Choice uses a combination of techniques to obtain the final raking. However, we could already confirm that the final rakings do not

[2] All the matrices calculated during the application of the AHP method have been automatically handled in Excel.

present a significant variation from those obtained using AHP.

5. Related Work

The need to deal with conflicting situations has become a critical area in aspect-oriented requirements engineering. Several authors have detected the need to handle conflict resolution in requirements engineering (for example [6, 8]) and the topic appears regularly in the Early Aspects workshop series. However, the existing work does not offer means for systematic conflict detection or trade-off analysis, yet. For example, [4] proposes a reasoning schema to detect and solve conflicting situations between non-functional concerns. However, this process does not support means for a rigorous consistency checking neither does it offer a systematic trade-off analysis technique.

Yen presents a formal framework that facilitates the identification and the tradeoff analysis of conflicting requirements [13]. Yen's approach describes a systematic approach for analyzing the tradeoffs between conflicting requirements using the techniques in decision science. This approach is based on a fuzzy set theoretic foundation for representing imprecise requirements. Our approach differs from this by offering a mathematical process to compute the consistency of priorities in order to minimize the errors during decision making. Moreover, AHP requires less computational resources to achieve its goal and solves conflicting situations without using complex operations.

Goal-oriented approaches have detected priority as the base attribute to handle conflicting goals. Formal techniques and heuristics are proposed for detecting conflicts and divergences from specifications of goals/requirements and of domain properties [5]. The major difference with respects to our work is that consistency of judgements is not guaranteed.

6. Conclusions and Future Work

This paper presented a rigorous technique to support conflict management at the aspect-oriented requirements engineering level. It represents an advance over existing approaches, offering, for the first time, a formal instrument to handle conflicts that can be integrated to the existing AORE body of knowledge.

The AHP method is used to find, given a set of alternatives and a set of decision criteria, the best alternative for a given problem. Such a technique is not only useful in the context we have described here, but it looks very promising as a tool to support architectural choices during the software architecture design. Based on specific conflict handling characteristics, the AHP is a valid option because:

- it offers an interpretation/analysis of the problem with low complexity;
- it uses deductible and systematic procedures as another way of reducing complexity of the analysis of the problem in the perspective of AHP;
- it permits tradeoffs in conflicting situations by changing parameters (for example, by changing the weights of the criteria) of the analytical model;
- it allows estimating the level of satisfaction of each alternative and to obtain the order of importance for each concern that is the main goal of using AHP model;
- it calculates the judgements logical consistency;
- its final result is based in the synthesis of the different judgements.

Currently we are working on how to simplify the overhead caused by the necessary pairwise comparisons. In the near future we will investigate how derivation of relative weights from different comparisons or duality approaches for eliciting comparisons can be used. We are also interested in optimizing the use of different linguistic scales (currently we are using Saaty's scales).

In the near future we plan to decrease the level of granularity, by handling conflicts at the responsibility level. Currently, some of the concerns are too coarse grained (e.g. availability, response time) and, apart from the possibility of having different interpretations for different contexts, it may well be that certain conflicting situations will disappear when handled at a lower abstraction. The decomposition of non-functional concerns will play a fundamental role in this process.

There are other fields from the AORE template (in Table 2), such as responsibilities, that have not been used in the decision process. We plan to explore if they can be useful, given that most of the existing AORE approaches use a similar concept.

Acknowledgements

We would like to thank the reviewers for their very thorough comments and useful links that will help us improve our work.

This work has been partially supported by the Portuguese FCT POSC//EIA/60189/2004 Grant.

References

1. M. Barbacci, M. Klein, T. Longstaff, C. Weinstock, "Quality Attributes", Technical Report CMU/SEI-95-TR-021, http://www.sei.cmu.edu/publications/documents/95.-reports/95.tr.021.html.
2. I. Brito, A. Moreira, "Integrating the NFR Approach in a RE Model", Early Aspects Workshop at AOSD'04. Lancaster, UK. 2004.

3. I. Brito, A. Moreira, “Towards a composition process for aspect-oriented requirements”, Early Aspects Workshop at AOSD Conference. Boston, USA, 2003.
4. L. Chung, B. Nixon, E. Yu, J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, 2000.
5. A. Lamsweerde, R. Darimont, E. Letier, “Managing Conflicts in Goal-Driven Requirements Engineering”, IEEE Transactions on SE, 24(11), 1998, pp. 908 – 926.
6. A. Moreira, A. Rashid, J. Araújo, “Multi-Dimensional Separation of Concerns in Requirements Engineering”, 13th IEEE International Conference on RE, Paris, France, August 2005.
7. R. Nord, W. Wood, P. Clements, “Integrating the Quality Attribute Workshop (QAW) and the Attribute-Driven Design (ADD) Method”, Technical Note CMU/SEI-2004-TN-017.
8. A. Rashid, A. Moreira, J. Araújo, “Modularization and Composition of Aspectual Requirements”, AOSD 2003 Conference, Boston, USA, ACM Press, March 2003.
9. T.L. Saaty, *The Analytic Hierarchy Process,* Network: McGraw-Hill, 1980.
10. E. Triantaphyllou, *Multi-Criteria Decision Making Methods: A Comparative Study*, Kluwer Academic Publishers, 2000.
11. Unified Modeling Language – Specification, version 2.0, http://www.omg.org, Jan 2004.
12. A.S., Williams *Quantitative methods for business*, South-Western publishing Co, 2000.
13. J. Yen, W. A. Tiao, “A Systematic Tradeoff Analysis for Conflicting Imprecise Requirements”, 3rd IEEE International Symposium on RE, 1997.