Utilizing TOPSIS: A Multi Criteria Decision Analysis Technique for Non-Functional Requirements Conflicts

Dewi Mairiza[1], Didar Zowghi[1], and Vincenzo Gervasi[1,2]

[1] Faculty of Engineering and Information Technology,
University of Technology, Sydney (UTS), Australia
[2] Dipartimento di Informatica
Università di Pisa, I-56125 Pisa, Italy
mairiza@it.uts.edu.au, Didar.Zowghi@uts.edu.au,
gervasi@di.unipi.it

Abstract. Experience shows that many software systems suffer from inherent conflict among Non-Functional Requirements (NFRs). It also confirms that resolution strategies for handling NFRs conflicts often result in changing overall design guidelines, not by simply changing one module. Therefore, in software system development, software developers need to analyse the NFRs and conflicts among them in order to make decisions about alternative design solutions. This paper presents the use of Multi Criteria Decision Analysis (MCDA) approach for NFRs conflict decision analysis. TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution), as one of the essential MCDA techniques has been adopted to resolve such conflict. We show how the systematic application of TOPSIS can assist software developers select the most preferable design solutions with respect to the conflicting NFRs. The quantitative result generated with this technique will be used as the basis for decision support. An example that shows the application of TOPSIS is also presented.

Keywords: Non-Functional Requirements, design solution, conflict resolution, MCDA, TOPSIS, decision analysis.

1 Introduction

Non-Functional Requirements play a critical role in the success of software projects. They address the essential issue of software quality [1-3] and they are also considered as the qualifications of operations [4, 5]. Prior study reveals that there are 252 types of NFRs listed in the literature [6]. Among them, 114 types correspond to the NFRs perspective in relation to the “quality”. This huge number reflects how NFRs can be more critical than individual Functional Requirements (FRs) in the determination of a system’s perceived success or failure. Neglecting NFRs may lead to software failure, as discussed in a series of systemic failures in the literature [6-9].

NFRs are interacting, which means that they tend to interfere, conflict, and contradict with one another [1]. Achieving a particular type of NFR can hurt the

D. Zowghi and Z. Jin (Eds.): APRES 2014, CCIS 432, pp. 31–44, 2014.

achievement of the other type(s) of NFRs. Unlike FRs, this inevitable conflict arises as a result of inherent contradiction among various types of NFRs [1, 2]. Certain combinations of NFRs in the software systems may affect the inescapable trade offs [2, 8, 10]. Dealing with and managing NFRs conflict is essential [11], not only because conflict among software requirements are inevitable [1, 12, 13], but also because conflicting requirements are one of the three main problems in the software development in term of the additional effort or mistakes attributed to them [13]. A study of two-year multiple-project analysis conducted by Egyed & Boehm [14, 15] reports that between 40% and 60% of requirements involved are in conflict, and among them, NFRs involved the greatest conflict, which was nearly half of the total requirements conflict [16]. Therefore, since conflict among NFRs have also been widely acknowledged as one of NFRs characteristics, managing this conflict as well as making this conflict explicit is important [17].

This paper presents the outcome of our longitudinal study of investigating conflicts among NFRs. Utilizing TOPSIS, an MCDA technique to resolve the NFRs conflicts is presented as the novel contribution of the paper. Integrating TOPSIS with our foregoing sureCM Framework can assist software developers performing NFRs conflict decision analysis quantitatively.

This article is organized in five sections. The first section is the introduction to NFRs and conflicts among them. The second section describes the research background and some earlier works. The use of TOPSIS for NFRs conflict decision analysis is presented in section three, continued by illustrating an example of how TOPSIS can be applied in NFRs conflict management in section four. Then, section five concludes this paper by highlighting some open issues that have emerged from the investigation.

2 Study Background

A number of techniques to manage NFRs conflict have been discussed in the literature [11]. Majority of them provide documentation, catalogue, or list of potential conflicts. These catalogues represent the interrelationships among various types of NFRs. Some examples are: the QARCC win-win approach [8, 18, 19], trace analyzer of the requirements traceability technique [20], and a technique that adopts a hierarchical constraint logic programming approach [21]. Apart from strength and weaknesses of each technique, NFRs can be viewed, interpreted, and evaluated differently by different people and different context within which the system is being developed. Consequently, the positive or negative relationships among NFRs are not always obvious. These relationships might change depending on the meaning of NFRs in the context of the system being developed. Due to this relative characteristic, cataloguing the NFRs relationships in order to represent the conflict among them would inevitably produce disagreement. Identifying the NFRs conflict without understanding the meaning of NFRs in the system may produce erroneous conflict identification and analysis.

This study is conducted as part of a long-term project of investigating the relative conflicts among NFRs. The project's ultimate goal is to develop a novel framework to effectively identify, characterize and resolve the NFRs conflict. Earlier versions of the framework have been published in [22-24]. The sureCM Framework utilizes an experimental approach as the basis to attain the evidence for managing the NFRs conflict. As shown in Figure 1, sureCM Framework has five-layer sequential process: P1 (Define Case); P2 (Identify Metrics and Measure); P3 (Setup and Run Experiments); P4 (Characterize Conflict); and P5 (Conflict Decision Analysis). Each process has different roles and outputs. Here, NFRs are characterized as the associated system functionality and systems operationalizations, and NFRs metrics and measures are used as parameters to gather the quantitative evidence in the experiments. Then, this empirical evidence will be used to perform conflict decision analysis. Conflict Decision Analysis (P5) process is currently limited to translating the experimental result into the conflict categorization. The decision about which alternative design solutions to be implemented within the system is not defined yet. Given the above context, we are motivated to perform further research into extending the framework for NFRs conflict decision analysis. The objective is to select the best design solution with respect to those conflicting NFRs. The main research question that we address is as follow:

> *"How can we use the Multi Criteria Decision Analysis (MCDA) approach to perform NFRs conflict decision analysis?"*

The utilization of MCDA approach for conflict decision analysis is presented as the novel contribution of this paper. This approach will be applied to analyze the alternative design solutions, with the ultimate goal to select the one that best satisfices[1] the conflicting NFRs.

3 NFRs Conflict Decision Using TOPSIS

Every decision requires the balancing of multiple factors, i.e. criteria. Therefore a formal analysis is needed to promote a good decision-making. Multi Criteria Decision Analysis (MCDA) assists decision makers to structure and solving decision problems involving multiple criteria. It provides guideline that help decision makers to organize and synthesize such information so that they will feel comfortable and confident about making the decision [26, 27]. It also helps to structure the problem. Based on these characteristics, we propose to apply MCDA to perform the NFRs conflict decision analysis in sureCM Framework. It can be used to evaluate and analyze the alternative design solutions. It can also be used to decide the best design solution that best satisfices the conflicting NFRs.

---

[1] Satisfice is the term first coined by Hebert Simon [25] H. A. Simon, "The science of the artificial," 1996.

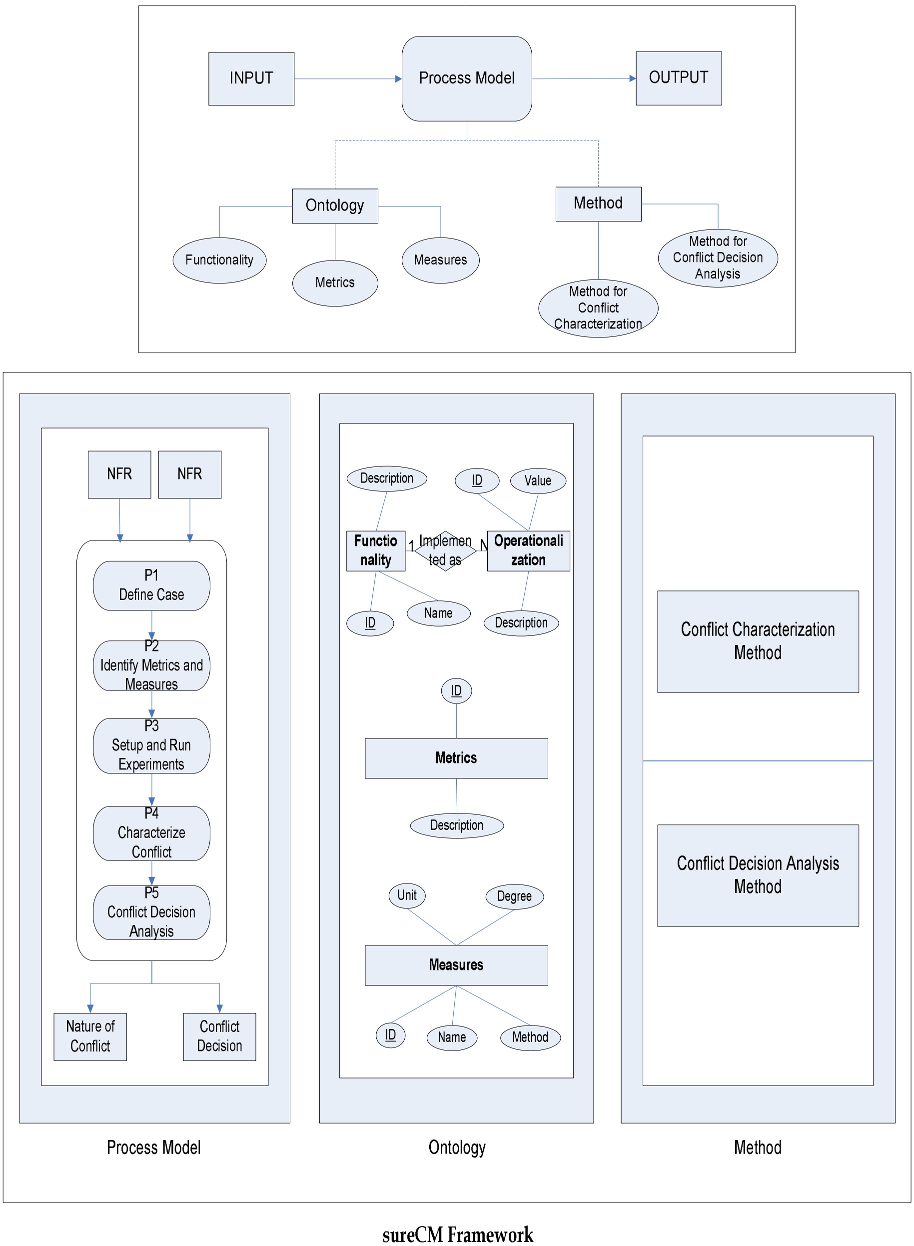

Fig. 1. sureCM Framework [23]

TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) is a goal-based technique in MCDA for finding the alternative that is closest to the ideal solution. The fundamental idea of TOPSIS is that the best solution is the one, which has the shortest distance to the ideal solution and the farthest distance from the negative-ideal solution. Therefore, the best solution is the one that can maximize all criteria.

TOPSIS consists of 6 steps as shown in Figure 2: (1) construct the normalized decision matrix; (2) construct the weighted normalized decision matrix; (3) determine the positive ideal solution and negative ideal solution; (4) determine separation from ideal solution; (5) calculate the relative closeness to the ideal solution; and (6) rank the preference order and select the closest option to ideal solution.

Some basic principles of TOPSIS are:

- ➔ The chosen alternative should be as close as possible to the ideal solution and as far as possible from the negative-ideal solution.
- ➔ The positive-ideal solution is formed as a composite of the best performance values exhibited (in the decision matrix) by any alternative for each attribute.
- ➔ The negative-ideal solution is the composite of the worst performance values; this means the one that has the worst attribute values.
- ➔ Proximity to each of these performance poles is measured in the Euclidean sense (e.g., square root of the sum of the squared distances along each axis in the "attribute space"), with optional weighting of each attribute.

Figure 2 shows the six steps of TOPSIS. It starts with creating an evaluation matrix $\mathbf{X_{ij}(_{mxn})}$ consisting of $\boldsymbol{m}$ alternative, $\boldsymbol{n}$ attributes/criteria and the score of each alternative with respect to each attribute. The matrix $\mathbf{X_{ij}(_{mxn})}$ is then normalized using such formula in step 1 to form the matrix $\mathbf{r_{ij}}$. A normalized decision matrix will be formed. Next step is calculating the weighted normalized decision matrix $\mathbf{v_{ij}}$ by multiplying the normalized scores $\mathbf{r_{ij}}$ by their corresponding weights $\mathbf{w_j}$. Weight is defined as a certain points that estimate the relative importance of each criteria. And weight is optional. Continue to step 3 to determine the positive ideal solution and negative ideal solution, that is, the worst alternative and the best alternative. $\mathbf{A^+}$ is the maximum value of each attribute, and $\mathbf{A^-}$ is the minimum value of each attribute.

Step 4 is then calculating the distance for each alternative to the ideal solution. This step is taken to calculate the similarity to the worst condition, by calculating the separation measures for each alternative from the positive ($\mathbf{S^+}$) and negative ($\mathbf{S^-}$) ideal solution. This is then continued by calculating the relative closeness to the ideal solution ($\mathbf{C_i^*}$) and rank the preference order of alternative based on its relative closeness to the ideal solution, i.e. a set of alternatives would be preference-ranked according to descending order of $\mathbf{C_i^*}$.

Step 1: Construct the normalized decision matrix.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$$

Step 2: Construct the weighted normalized decision matrix.

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1j} & \cdots & v_{1n} \\ \vdots & \vdots & & \vdots & & \vdots \\ v_{i1} & v_{i2} & & v_{ij} & & v_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mj} & \cdots & v_{mn} \end{bmatrix} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & \cdots & w_j r_{ij} & \cdots & w_n r_{1n} \\ \vdots & \vdots & & \vdots & & \vdots \\ w_1 r_{i1} & w_2 r_{i2} & & w_j r_{ij} & & w_n r_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ w_1 r_{m1} & w_2 r_{m2} & \cdots & w_j r_{mj} & \cdots & w_n r_{mn} \end{bmatrix}$$

Step 3: Determine the positive ideal (S+) and negative ideal (S---) solutions.

$$A^+ = \{(\max_i v_{ij} \mid j \in J), (\min_i v_{ij} \mid j \in J') \mid i = 1,2,\ldots m\}$$
$$= \{v_1^+, v_2^+, \ldots, v_j^+, \ldots, v_n^+\}$$

$$A^- = \{(\min_i v_{ij} \mid j \in J), (\max_i v_{ij} \mid j \in J') \mid i = 1,2,\ldots m\}$$
$$= \{v_1^-, v_2^-, \ldots, v_j^-, \ldots, v_n^-\}$$

$$\text{where} \quad J = \{ j = 1,2,\ldots,n \mid j \ \text{associated with benefit criteria} \}$$
$$J' = \{ j = 1,2,\ldots,n \mid j \ \text{associated with cost criteria} \}$$

Fig. 2a. – NFRs Conflict Decision Analysis with TOPSIS (Step 1 – 3)

Step 4: Calculate the separation measures for each alternative.

– **Ideal Separation**

$$S_i^+ = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^+)^2} \qquad i = 1,2,\ldots,m$$

– **Negative-Ideal Separation**

$$S_i^- = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^-)^2} \qquad i = 1,2,\ldots,m$$

Step 5: Calculate the relative closeness to the ideal solution Ci*.

$$C_i^* = \frac{S_i^-}{(S_i^+ + S_i^-)}, \qquad 0 < C_i^+ < 1, \qquad i = 1,2,\ldots,m$$

$$C_i^* = 1 \quad if \quad A_i = A^+$$

$$C_i^* = 0 \quad if \quad A_i = A^-$$

Step 6: Rank the preference order.

Select the option with Ci* closest to 1 (the highest one).

Fig. 2b. NFRs Conflict Decision Analysis with TOPSIS (Step 4 – 6)

Figure 3 shows how TOPSIS can be implemented in sureCM Framework. In the framework, input for TOPSIS is the conflict relationship diagram, which is obtained from the previous sureCM process (P3 and P4). Conflict relationship diagram (as shown in Figure 4) is a two-dimensional conflict relationship graph that uses quantitative data obtained in process P3, i.e. running the experiments, as the evidence of existence of conflict [23]. Each operationalization taken in the experiments will be plotted based on its NFRs metrics calculation result. By plotting all of the defined operationalizations, a conflict relationship characterization will be created. In the context of this framework, the criteria refer to the conflicting NFRs, and the alternatives refer to the alternative design solutions/operationalizations. Output of this conflict decision process is a decision, which is the ranking of each alternative design solution based on its closeness to the ideal solution, i.e. maximum satisficing for each conflicting NFRs.

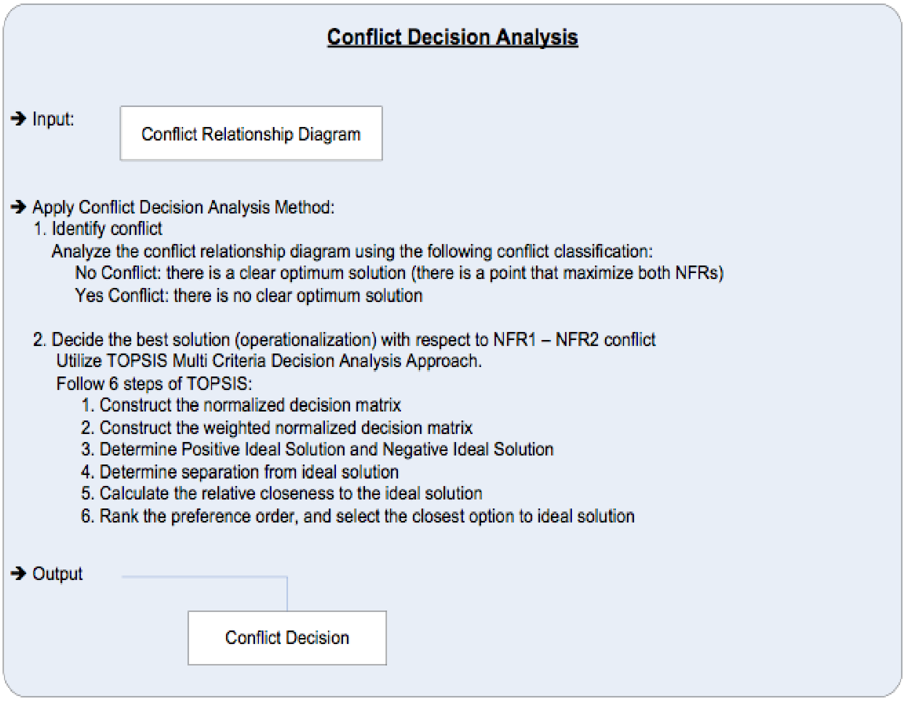

Fig. 3. sureCM Framework Conflict Decision Analysis

4 Applying TOPSIS: An Example

To show how TOPSIS can be applied for NFRs conflict decision analysis, in this paper we use two statement of NFR from the Chemical Tracking System [23]. We consider the two NFRs given in the specification document. NFR1 is considered a security requirement, while NFR2 is a usability requirement.

NFR 1: The Chemical Tracking System shall have identified/authenticated the user and protect user's personal information.

NFR 2: A chemist who has never used the system before shall be able to learn using the system easily and independently.

The sureCM Framework supports the characteriz and identification of conflict, as well as performing conflict decision analysis. The details of NFRs conflict characterization and identification (process P1; P2; P3; P4) have been presented previously in [23], so here we only focus on conflict decision analysis (process P5). As described in [23], the first four processes of sureCM Framework produce a set of experimental data and the conflict relationship diagram. By using 7 types of alternative design solutions for implementing security and usability, i.e. (1) Fixed Key; (2) Smart Card; (3) Scrambled Key; (4) Geometrical Pin Code; (5) Finger Print; (6) Palm Scanner; (7) Retina Scanner, the nature of conflict is illustrated as follow:

Exp. ID	Ops.	Security	Usability	Note
GA.T.1	Fixed Key	4	0.0641	---
GA.T.2	Smart Card	1	0.1053	---
GA.T.3	Scrambled Key	7	0.0503	---
GA.T.4	Geometrical Pin Code	9	0.0500	---
GA.T.5	Finger Print	1	0.1563	---
GA.T.6	Palm Scanner	1	0.1370	---
GA.T.7	Retina Scanner	1	0.1266	---

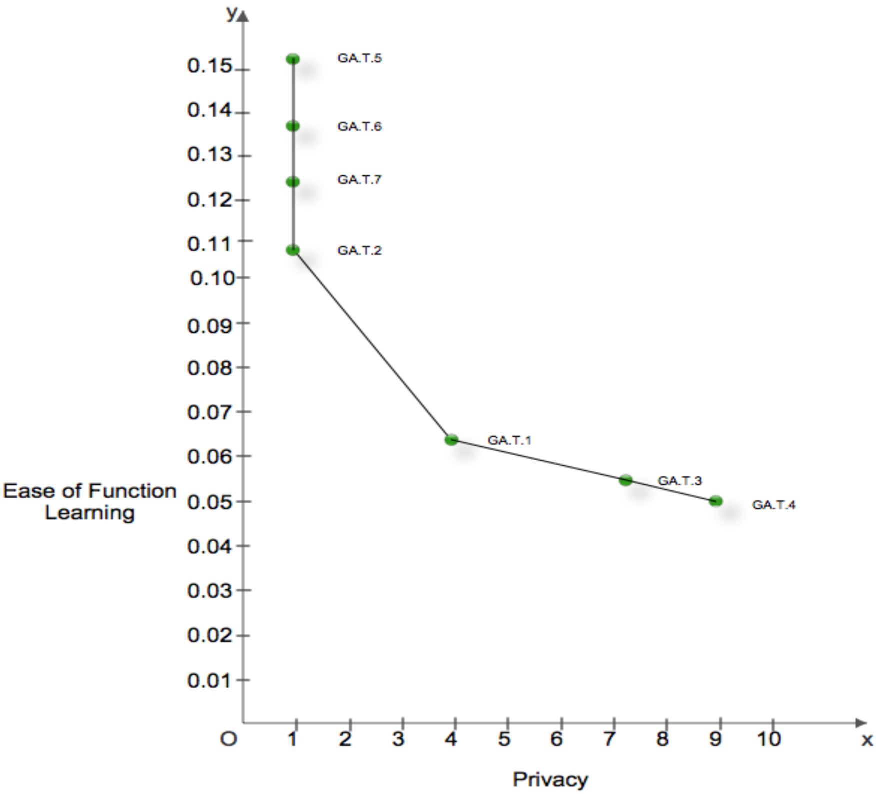

Fig. 4. Conflict Relationship Diagram of Security – Usability Conflict [23]

This will be used as the input for NFRs conflict decision analysis with TOPSIS, presented in Table 1. Then each TOPSIS step is described.

Table 1. TOPSIS Input

Weight	1	1
	Security	Usability
GA.T.1	4	0.0641
GA.T.2	1	0.1053
GA.T.3	7	0.0503
GA.T.4	9	0.0500
GA.T.5	1	0.1563
GA.T.6	1	0.1370
GA.T.7	1	0.1266
Goal	Maximize	Maximize

Step 1: Construct the normalized decision matrix.
Using formula in TOPSIS step 1, we normalize input data to form the matrix r_{ij}, shown in Table 2.

Table 2. TOPSIS Step 1

Alternative	Security	Usability
GA.T.1	0.326598632	0.2274
GA.T.2	*0.081649658*	0.3736
GA.T.3	0.571547607	0.1784
GA.T.4	**0.734846923**	*0.1774*
GA.T.5	*0.081649658*	**0.5545**
GA.T.6	*0.081649658*	0.4860
GA.T.7	*0.081649658*	0.4491

Step 2: Construct the weighted normalized decision matrix.
In this software project, there is no priority set for each criteria. Both security and usability have the same priority level. Therefore, the weighted normalized decision matrix gives the same result as generated in step 1.

Step 3: Determine the positive ideal (S+) and negative ideal (S-) solutions by determining A^+ and A^- for each criteria. This is done by selecting the highest element (for A^+) and the lowest element (for A^-) in each criteria of the matrix in Table 2. The result is presented in Table 3.

Table 3. Positive Ideal and Negative Ideal Solutions

A+

Security	Usability
0.734846923	0.55450207

A-

Security	Usability
0.081649658	0.1773839

Step 4: Calculate the separation measures for each alternative
Euclidean distance is used to measure the separation of each alternative design solution from the ideal alternative (positive ideal) and negative ideal alternative.

Table 4. Separation Measures

Alternative	(S+)	(S-)
GA.T.1	0.523123696	0.250004453
GA.T.2	0.677792669	0.196186593
GA.T.3	0.409979485	0.489899105
GA.T.4	0.377118171	0.653197265
GA.T.5	0.653197265	0.377118171
GA.T.6	0.656776090	0.308647985
GA.T.7	0.661640891	0.271752134

Step 5: Calculate the relative closeness to the ideal solution C_i^*
The higher C_i^* is the better, which means the closer the alternative to the ideal solution. Using TOPSIS step 5 formula, C_i^* is calculated and presented in Table 5.

Table 5. Relative Closeness

Alternatives	C*
GA.T.1	0.323367417
GA.T.2	0.224475112
GA.T.3	0.544405779
GA.T.4	0.633977947
GA.T.5	0.366022053
GA.T.6	0.319701977
GA.T.7	0.291144381

Step 6: Rank the preference order
A set of alternatives can now be preference ranked according to the descending order of *Ci**. The best solution is the alternative with C_i^* closest to 1, which is the highest one. The result of step 6 is presented in Table 6.

Table 6. Final Result

Alternatives	C*	Ranked
GA.T.1	0.323367417	4
GA.T.2	0.224475112	7
GA.T.3	0.544405779	2
GA.T.4	0.633977947	1
GA.T.5	0.366022053	3
GA.T.6	0.319701977	5
GA.T.7	0.291144381	6

Ranked	Design Solutions
1	Geometrical Pin Code
2	Scrambled Key
3	Finger Print
4	Fixed Key
5	Palm Scanner
6	Retina Scanner
7	Smart Card

As shown in Table 6, the highest C_i^* is the alternative 4 (C_i^* = 0.633977947), that means GA.T.4 is the design solution that has maximum security and maximum usability, among other alternatives. Therefore, according to TOPSIS, software developer should consider taking alternative 4 (Geometrical Pin Code) as the design solution that can maximize the satisfaction of those conflicting NFRs, security and usability.

5 Conclusion

This paper describes a novel idea of utilizing TOPSIS, an MCDA technique, to resolve the conflict among NFRs, particularly to perform conflict decision analysis that can assist software developer deciding the best alternative design solution that can maximize the satisficing of NFRs in conflict. Conflict decision analysis using TOPSIS will be integrated as part of our foregoing sureCM Framework [22-24], i.e. an integrated experimental-based framework for NFRs conflict management and analysis. Requirements statement from a Chemical Tracking System has been used as an example to show how TOPSIS can be applied.

As part of a long-term project of investigating conflict among NFRs, a number of important task remain to complete:

1) Conducting empirical evaluation

The effectiveness of the framework will be empirically evaluated through controlled experiments. The reason for conducting controlled experiments is because: (a) "controlled experiments make it possible for the careful observation and precise manipulation of independent variables (e.g. proposed framework); (b) allowing for greater certainty; and (c) encourage the researcher to try out novel frameworks in a safe and exploratory environment before implementing them in the real world settings" [28]. Effectiveness and efficiency will be used as the evaluation criteria. Effectiveness means that this framework can be used to manage the NFRs conflict by considering NFRs relative characteristic, while efficiency represents how fast people can identify the conflict using the framework.

2) Developing a semi-automatic tool

To support the framework utilization, we also plan to develop a semi-automatic tool that can assist software developers, particularly requirements engineers to perform conflict management among NFRs.

Acknowledgments. We would like to thank The International Schlumberger Foundation Paris for funding this research through Faculty for the Future Award Program. We would also like to thank The Centre for Human Centered Technology Design (HCTD) at UTS for partially funding Dr Gervasi's trip for research collaboration.

References

[1] Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-functional requirements in software engineering. Kluwer Academic Publishers, Massachusetts (2000)

[2] Ebert, C.: Putting requirement management into praxis: Dealing with nonfunctional requirements. Information and Software Technology 40, 175–185 (1998)

[3] Firesmith, D.: Using quality models to engineer quality requirements. Journal of Object Technology 2, 67–75 (2003)

[4] Kotonya, G., Sommerville, I.: Non-functional requirements (1998)

[5] Mittermeir, R.T., Roussopoulos, N., Yeh, R.T., Ng, P.A.: Modern software engineering, foundations and current perspectives. Van Nostrand Reinhold Co., New York (1989)

[6] Mairiza, D., Zowghi, D., Nurmuliani, N.: An investigation into the notion of non-functional requirements. In: 25th ACM Symposium on Applied Computing, Switzerland (2010)

[7] Breitman, K.K., Prado Leite, J.C.S., Finkelstein, A.: The world's a stage: A survey on requirements engineering using a real-life case study. Journal of the Brazilian Computer Society 6, 1–57 (1999)

[8] Boehm, B., In, H.: Identifying quality-requirements conflict. IEEE Software 13, 25–35 (1996)

[9] Leveson, N.G., Turner, C.S.: An investigation of the Therac-25 accidents. IEEE Computer 26, 18–41 (1993)

[10] Wiegers, K.E.: Software requirements, 2nd edn. Microsoft Press, Washington (2003)

[11] Mairiza, D., Zowghi, D., Nurmuliani, N.: Managing conflicts among non-functional requirements. In: 12th Australian Workshop on Requirements Engineering (AWRE 2009), Sydney, Australia (2009)

[12] Chung, L., Nixon, B.A., Yu, E.: Dealing with change: an approach using non-functional requirements. Requirements Engineering 1, 238–260 (1996)

[13] Curtis, B., Krasner, H., Iscoe, N.: A field study of the software design process for large systems. Communication of the ACM 31, 1268–1287 (1988)

[14] Boehm, B., Egyed, A.: WinWin requirements negotiation processes: A multi-project analysis. In: 5th International Conference on Software Processes (1998)

[15] Egyed, A., Boehm, B.: A comparison study in software requirements negotiation. In: 8th Annual International Symposium on Systems Engineering, INCOSE 1998 (1998)

[16] Robinson, W.N., Pawlowski, S.D., Volkov, V.: Requirements interaction management. ACM Computing Surveys 35, 132–190 (2003)

[17] Paech, B., Kerkow, D.: Non-functional requirements engineering - quality is essential. In: 10th International Workshop on Requirements Engineering: Foundation for Software Quality, pp. 27–40 (2004)

[18] Boehm, B., In, H.: Aids for identifying conflicts among quality requirements. IEEE Software (March 1996)

[19] In, H., Boehm, B., Rodgers, T., Deutsch, M.: Aplying WinWin to quality requirements: A case study. In: 23rd International Conference on Software Engineering, Toronto, Ontario, Canada, pp. 555–564 (2001)

[20] Egyed, A., Grünbacher, P.: Identifying requirements conflicts and cooperation: how quality attributes and automated traceability can help. IEEE Software 21, 50–58 (2004)

[21] Guan, Y., Ghose, A.K.: Use constraint hierarchy for non-functional requirements analysis. In: Lowe, D., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 104–109. Springer, Heidelberg (2005)

[22] Mairiza, D., Zowghi, D.: An ontological framework to manage the relative conflicts between security and usability requirements. In: The Third International Workshop on Managing Requirements Knowledge (MaRK 2010), in conjunction with the 18th IEEE International Requirements Engineering Conference (RE 2010), Sydney, Australia (2010)

[23] Mairiza, D., Zowghi, D., Gervasi, V.: Conflict characterization and analysis of non functional requirements: An experimental approach. In: 12th International Conference on New Trends in Software Methodologies, Tools and Techniques (SOMET 2013), Budapest, Hungary, pp. 83–91 (2013)

[24] Mairiza, D.: Non-functional requirements in software development projects: A systematic review. Presented at the ACS – BRASIG 29, Sydney, Australia (September 2011)

[25] Simon, H.A.: The science of the artificial (1996)

[26] Belton, V., S.T.J.: Multiple criteria decision analysis: An integrated approach. Kluwer Academic Publishers (2002)

[27] Zopounidis, C., Pardalos, P.M.: Handbook of multicriteria analysis. Springer, Heidelberg (2010)

[28] Damian, D.: Empirical studies of computer support for distributed requirements negotiation. University of Calgary (2001)