2015 12th International Conference on Information Technology - New Generations

A Fuzzy Decision Support Approach for Model-Based Tradeoff Analysis of Non-Functional Requirements

Mehrdad Saadatmand*†, Sahar Tahvili*
*Mälardalen University, Västerås, Sweden, mehrdad.saadatmand@mdh.se, sti10002@student.mdh.se
†Swedish Institute of Computer Science (SICS), Västerås, Sweden, mehrdad.saadatmand@sics.se

***Abstract*—One of the main challenges in addressing Non-Functional Requirements (NFRs) in designing systems is to take into account their interdependencies and mutual impacts. For this reason, they cannot be considered in isolation and a careful balance and tradeoff among them should be established. This makes it a difficult task to select design decisions and features that lead to the satisfaction of all different NFRs in the system, which becomes even more difficult when the complexity of a system grows. In this paper, we introduce an approach based on fuzzy logic and decision support systems that helps to identify different design alternatives that lead to higher overall satisfaction of NFRs in the system. This is achieved by constructing a model of the NFRs and then performing analysis on the model. To build the model, we use a modified version of the NFR UML profile which we have introduced in our previous works, and using model transformation techniques we automate the analysis of the model.**

Keywords*—*Non-Functional Requirements, Tradeoff Analysis, Multi-Criteria Decision Making, Fuzzy, TOPSIS, Optimization, Software Architecture.

I. Introduction

The ultimate goal in software engineering is to build software products which satisfy the requirements of different stakeholders. Requirements are generally categorized into functional and Non-Functional Requirements (NFRs) [1]. In the simplest form, functional requirements are those which define what the system should do, while the term non-functional requirement is used for requirements which specify how a system should perform. NFRs are usually described with terms that end with: 'ility' as in availability, 'ity' such as atomicity; while a few other ones such as performance and user-friendliness do not follow this pattern [1], [2].

While functional requirements usually receive more attention during the design of a system, there are domains in which NFRs play a critical role in determining the correctness of a system. An example of such cases are real-time systems whose correctness is not only dependent on the correctness of the logical results of computations, but also on the time at which the results are produced [3].

Addressing NFRs in the design of systems, however, is a challenging task. One of the challenges associated with NFRs is that NFRs have interdependencies and mutual impact on each other, and thus, cannot be considered separately and in isolation. For this reason, a careful tradeoff analysis needs to be done on NFRs in order establish balance among them [2], [4].

In our previous works [2], [4], we have introduced a model-based approach for modeling and tradeoff analysis of NFRs. As part of the approach we have also defined and introduced a UML profile called NFR Profile. In the approach and using the NFR profile, first NFRs are modeled along with different features in the system that are considered and adopted for satisfying those requirements. Moreover, the interdependecies and mutual impacts of features and NFRs are also modeled using appropriate types of relationships provided as part of the UML profile. To perform the analysis, an in-place model transformation solution is used to navigate the model, read in impact values and other important information, and then as the result, update the model with the calculated values indicating satisfaction degree of NFRs in the system. So in this approach, basically the user (e.g., system designer here), has already chosen a set of features, and also is expected to provide quantified values on the relationships between features and NFRs.

In this paper, based on the aforementioned approach, we introduce a new approach to provide the following important features and improvements:

- Apply fuzzy logic to ease the requirement on the user to provide detailed and accurate quantified values on relationships.
- By performing tradeoff analysis, enable to identify from a set of alternative design solutions, which one should be adopted to achieve an overall higher satisfaction degree for NFRs in the system.

On the other hand, in our previous work, it was considered that the user constructs the model including a set of specific features to satisfy NFRs. Then tradeoff analysis was performed on the model and it was left to the user to decide if this design is good enough with respect to the calculated satisfaction values of its NFRs or not. Here in this paper, our new approach, basically helps the user to identify which features to choose for satisfying an NFR considering all their impacts and side effects on other NFRs in the system. Therefore, as the starting point, the user first creates the NFR model of the system but consisting only of generic specification of features that he/she wants to be there. The output of the approach is that a specific

978-1-4799-8828-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ITNG.2015.24

set of the features are then identified for the user. In other words, the approach targets cases where different versions or implementations of a feature are available (e.g., in a repository of components), which have different characteristics in terms of impacts and side effects in the system. The importance of our proposed approach lies in the fact that such aforementioned interdependencies and impacts between different NFRs and features can easily get dramatically complex when the number of NFRs and also available features in a system grow. For this reason, there is a need to not only have an approach to find out which design alternative to choose, but also to do it in an automated way.

In short, in the new approach proposed in this paper, different alternatives that contribute to the satisfaction of each NFR along with their respective impacts on other NFRs in the system are considered. Then by applying TOPSIS [5], which is a method for multi-criteria decision analysis, a comparison is done on different sets of alternatives and the best one is picked out. The main criteria here is to collectively achieve a better and higher overall satisfaction for the NFRs in the system. As part of the approach, as we describe in detail in the rest of the paper, UML profiling technique is used for modeling the necessary concepts and elements, model transformation techniques are used to automate the analysis of the model, fuzzy logic is used to ease specification of impact degrees, and multi-criteria decision analysis using TOPSIS is applied to identify the best alternative.

The remainder of the paper is structured as follows. In Section II, background and preliminary information necessary to follow the applied techniques in the paper are explained. In this section, we also describe the NFR profile along with modifications done in it to incorporate fuzzy concepts.The proposed approach is described in Section III and its implementation and application example is provided in Section IV. In Section V some related works are discussed, and finally in Section VI, concluding remarks and future directions of this work are discussed.

II. Background & Preliminaries

A. UML Profile

One of the benefits of Model-Based Development is that it enables to perform analysis at earlier phases of development. This is provided for by capturing right amount of details and necessary information as part of models. For this purpose, the modeling language should be expressive enough to support modeling of desired concepts relevant in the domain. Generally, two methods are considered for defining a modeling language: 1) to define a Domain Specific Modeling Language (DSL/DSML) from scratch; 2) to extend and tailor a general purpose modeling language such as UML using its extension mechanism; i.e., UML Profiling. There are different benefits and drawbacks for each of the two approaches which are discussed in [6], [7]. One of the advantages of using UML profiles is that the learning curve to familiarize oneself with a UML Profile is usually less, as it is more probable that people in an organization may already be familiar with UML and have been using it. To customize and tailor the modeling concepts in UML, it offers some extension mechanisms such as tagged values, stereotypes and constraints. A profile, in essence, is a collection of such customizations and extensions which is intended for a particular domain. As examples of UML profiles, SysML (System Modeling Language) [8], and UML Profile for MARTE (Modeling and Analysis of Real-Time Embedded Systems [9] can be named.

B. NFR Profile (Fuzzy Version)

In [2], we defined and introduced a profile for generic modeling and tradeoff analysis of NFRs, named as NFR Profile. In this section, we revisit NFR Profile with some modifications that are made here in order to enable and incorporate fuzzy concepts in the approach, and also to capture different feature alternatives. Figure 1 shows the structure of the profile depicting the stereotypes and concepts that it offers, as well as different properties that each stereotype has along with their type and definition:

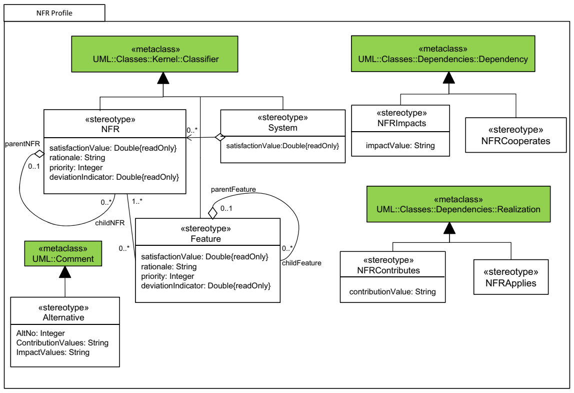

Fig. 1: NFR Profile

Description of the stereotypes and properties that are defined and considered in the profile are follows [2]:

`System`: In the hierarchy of NFRs (considering the refinement of NFRs into other NFRs), the root node will represent the system itself which can have several different NFRs represented in the model at the lower levels of the hierarchy as children model elements. The System stereotype is used to annotate this root model element as the system. The system is also considered as the context of the analysis.

`SatisfactionValue`: This property is used to represent the satisfaction degree of the model element it belongs to and to what extent it has been fulfilled. As can be seen in Figure 1, several stereotypes have this property. In case of the System stereotype, the value of this property shows the total calculated satisfaction value for the system (described later). This value is calculated and set by the analysis engine and the users cannot set it.

`NFR`: NFRs in the system are stereotyped and annotated with this stereotype. Since NFRs can have other NFRs as refinements and thus as children nodes, an association relationship to itself (reflexive aggregation) has been defined for it.

`Feature`: A feature in the system that is defined to satisfy an NFR is identified by using this stereotype. It is basically the equivalent of *Operationalization* concept in NFR Framework and Softgoal Interdependency Graph (SIG) [10] or *tactics* as used in [11].

NFRContributes: This stereotype is used to indicate that an NFR or Feature contributes directly to the satisfaction of another one. It has a property called contributionValue that specifies the degree of this contribution.

NFRImpacts: this is similar to NFRContributes stereotype but is used to include the impact of a model element on other NFRs. In other words, this stereotype is defined to capture the *side effects* of features and NFRs. ImpactValue property of this stereotype shows the degree of the impact.

NFRApplies: This stereotype is defined to enable the possibility to relate the NFR model to functional model elements (e.g. an NFR that applies to a component). For instance, if there is already a UML model of the system available (e.g., a class diagram), with this stereotype it can be specified to which part of that model an NFR or Feature applies and is related to.

Rationale: The rationale behind having an NFR or Feature and any other description about it can be captured and specified in this property. Both NFR and Feature stereotypes have this property.

Alternative: Different alternatives (i.e., different types, versions, implementations) for a feature in the system are described by using this stereotype. Each alternative can have different characteristics in terms of its contribution and also impact values on other nodes. This is captured by using the ContributionValues and ImpactValues properties of this stereotype. To distinguish between alternatives, this stereotype includes a property called AltNo.

Priority: This property which exists in both NFR and Feature stereotypes captures the preferences of customers (and also developers priorities when relevant and applicable) and their priorities in terms of the relative importance of NFRs and Features.

DeviationIndicator: By taking into account the priority and the satisfaction value of an NFR or Feature, a value for this property is calculated (as will be described soon) and provided which indicates to the designer the importance and magnitude of how much the satisfaction of an NFR or Feature has deviated or been violated. The deviation indicator value basically shows and helps to identify which parts of the system have deviated more from the specification (i.e., from being fully satisfied) and may need to be modified to achieve a better satisfaction level. This value is also calculated and set by the analysis engine and the users cannot set it. While the satisfaction value does not reflect user preferences and priorities, the deviation indicator value identifies to the designers which parts need to be considered first with respect to the preferences and priorities of the customers. This is especially helpful and beneficial for identifying such parts in complex systems.

To use the profile and perform calculations, there are several rules and assumptions that are defined on model elements and their relationships, and also on how to set and calculate values for different properties:

- The priority for an element can be an integer value between 1 and 5, with 5 being the highest priority and 1 the lowest.
- The satisfaction value for each leaf node is considered to be always 1 (since it is not meant to have and use a feature or component which is not fully implemented and is only working partially).
- The contributionValue property of the NFRContributes link connecting a child node to its parent shows how much and to what degree a node (e.g., a feature) contributes to the satisfaction of its parent NFR. The value set for this property will be a fuzzy variable (e.g., Very Low, Low, Medium, etc.). A fuzzy membership function can be considered to map fuzzy values to fuzzy variables.
- For impactValue property of NFRImpacts links, like contributionValue property of NFRContributes links, a fuzzy variable can be assigned and set.
- The contribution of a child node to its parent is calculated by multiplying the corresponding fuzzy value of the fuzzy variable (e.g., 0.9 for Very High) that is assigned to the satisfactionValue property of the child node by fuzzy value of the contributionValue property of the NFRContributes link that connects it to the parent.
- To calculate the satisfaction value of a node (i.e., an NFR), we start first by determining the total impacts that other nodes have on that node, as follows. The total impact for a node, denoted as I, is calculated as the sum of all impact values (their fuzzy values) of NFRImpacts links coming to that node. This is shown in the following formula considering that i_j is the fuzzy value for the impact of another node on the node for which we want to calculate the total impact value:

$$I = \sum i_j \tag{1}$$

- To calculate the satisfactionValue of a node, the total contributions from all of its children nodes are calculated, and then the total impact value is also taken into account. If s_k is the satisfaction value for each child node of a node, l_k is the fuzzy contribution value on the link that connects the child node k to its parent node (NFRContributes relationship), and I is the calculated total impact value, the satisfaction value of the node (parent for the considered children nodes) is calculated as:

$$\text{S} = \begin{cases} \min((\sum s_k * l_k) - I, 1) & \text{if}(\sum s_k * l_k) - I \geq 0 \\ 0 & \text{if}(\sum s_k * l_k) - I < 0 \end{cases} \tag{2}$$

Considering the above rules and formulas, the satisfaction value of a node will be in the range of 0 and 1. To perform these calculations, nodes are navigated and traversed starting from leaf nodes (considering that the satisfaction of leaf nodes is 1) and values are calculated using the above formulas upwards toward the top element which is the system.

- The DeviationIndicator is calculated after the calculation of satisfaction values using the following formula:

$$\begin{aligned} &DeviationIndicator = \\ &Priority - Priority * SatisfactionValue \end{aligned} \tag{3}$$

Based on this calculation and considering that the SatisfactionValue is always between 0 and 1 and priority is an integer value between 1 and 5, the value of DeviationIndicator will be in the range of $[0, 5]$. The perfect situation is when the DeviationIndicator value is 0, and the more this value increases the more is the deviation from the desired design, and thus, it indicates a bigger and more severe problem.

C. Decision Support Systems

A Decision Support System (DSS) [12] is a computer system to process information in order to support various decision making activities. Decision making is a process to make choices between different alternatives. There are various ways to perform decision making such as rational, intuitive, combinations and fuzzy. In this paper we apply a fuzzy decision support approach for solving our problem.

D. Fuzzy Set and Membership Function on IFS

Fuzzy set theory is an extension of set theory which is also used in solving multi criteria decision making problems over the recent decades [13], introduced by Lotfi A. Zadeh and Dieter Klaua in 1965. In the present paper we use TOPSIS[1], which is a multi-criteria decision analysis method, under fuzzy environment to solve a multi criteria decision making problem. This type of fuzzy optimization method for multicriteria decision making is based on the inclusion degrees of Intuitionistic Fuzzy Sets (IFS) [14].

A membership function indicates the degree of truth as an extension of evaluation. Fuzzy truth represents membership in vaguely defined sets. Some basic concepts of fuzzy logic which are relevant for this work based on the definitions provided by Yun Shi [15], Yang [16] and Kerre [17] are revisited here:

Let A be a space of points, with a generic element of A denoted by a, thus $A = \{a\}$.

Definition 1. A fuzzy set is a pair (A, m) where A is a set and $m : A \to [0, 1]$; for each $x \in A$, $m(x)$ is called the grade of membership of x in (A, m); for a finite set $A = \{x_1, \ldots, x_n\}$, the fuzzy set (A, m) is often denoted by $\{m(x_1)/x_1, \ldots, m(x_n)/x_n\}$.

Let $x \in A$; then x is called fully included in the fuzzy set (A, m) if $m(x) = 1$ and is called not included if $m(x) = 0$. The set $\{x \in A|m(x) > 0\}$ is called the support of (A, m) and the set is called a kernel, where x is a fuzzy member if $0 < m(x) < 1$, [16].

Definition 2. For any set X a membership function on X is any function from X to the real unit interval $[0, 1]$.

The membership function which represents a fuzzy set A is denoted by μ_A.

Definition 3. For an element x of X, the value $\mu_{A(x)}$ is called the membership degree of x in the fuzzy set A, [13].

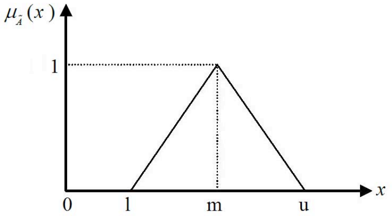

Fig. 2: Diagram of triangular membership functions

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x-l}{m-l}, & l < x < m \\ 1 & x = m \\ \frac{u-x}{u-m}, & m < x < u \\ 0 & otherwise \end{cases}$$

According to [5] using an additional degree and Intuitionistic fuzzy sets (IFS) can be used when no initial knowledge is available for the information to be modeled (i.e., limited history about prior trends).

Definition 4. An Intuitionistic Fuzzy Set (IFS) A on a universe U is defined as the following form:
$A = \{(u, \mu_A(u), \nu_A(u)) \mid u \in U\}$, where the functions $u_A : U \to [0, 1]$ and $v_A : U \to [0, 1]$ define the degree of membership and the degree of non-membership of the element $u \in U$ in A, respectively, and for every $u \in U$ we have $0 \leq \mu_A(u) + \nu_A(u) \leq 1$, [18].

According to [5] a fuzzy set can be written as:

$$\{(u, \mu_A(u), 1 - \mu_A(u)) \mid u \in U\} \tag{4}$$

IFS distributes fuzzy sets for every membership function μ and non-membership functions ν where $\nu = 1 - \mu$.

The following table summarizes the classical binary implication:

Definition 5. A mapping $I : [0, 1]^2 \rightharpoonup [0, 1]$ is a fuzzy implication if it satisfies the boundary conditions:
$I(0, 0) = I(0, 1) = I(1, 1) = 1$ and $I(1, 0) = 0$, [15].

A fuzzy implication can be generated by using three different approaches, R-implications, S-implications and QL-implications. In the present paper we use R-implications.

[1] Technique for Order of Preference by Similarity to Ideal Solution

a	b	a → b
0	0	1
0	1	1
1	0	0
1	1	1

TABLE I: Binary implication

Inclusion degree function of IFS: Assume U is a finite universe and R is an implication; I_{IFS} is a an inclusion degree function of IFS if R satisfies the following conditions [15]:

- $\forall a, b \in [0,1]$ and $a \leq b \Rightarrow R(a,b) = 1$
- $R(a,b)$ is non-decreasing with respect to b and non-increasing with respect to a.

By using this definition we can write

$$\begin{aligned} I_{IFS}(A,B) = \frac{1}{\mid U \mid} \sum_{u \in U} [\lambda R_\pi(\mu_A(u), \\ \mu_B(u)) + (1-\lambda) R_\pi(\nu_B(u), \nu_A(u))], \lambda \in [0,1] \end{aligned} \tag{5}$$

where $\mid U \mid$ is the cardinality of U which can be calculated by, [19],

$$\mid U \mid = \sum_{u \in U} \frac{1 + \mu_A(u) - \nu_A(u)}{2}. \tag{6}$$

There are different methods to calculate an R-implication which was introduced by several mathematicians. We use Goguen implication:

$$R_\pi(a,b) = \begin{cases} 1, & if\ a = 0, \\ \min\left(\frac{b}{a}, 1\right) & if\ a > 0 \end{cases} \tag{7}$$

E. TOPSIS method in multiple criteria fuzzy decision making

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is a well known decision making method, developed by Hwang and Yoon [5]. According to the definition of the TOPSIS method, we need to find a positive ideal solution A^+ and also a negative ideal solution A^-. The best alternative is an alternative which has the shortest distance from A^+ and also the farthest distance from A^-.
Assume that we have a set of alternatives M and a set of criteria C:

$$\begin{aligned} M = \{M_1, M_2, ..., M_n\} \\ C = \{C_1, C_2, ..., C_m\} \end{aligned}$$

we assume that the alternatives and criteria are represented (using IFS) as:

$$\begin{aligned} M_1 &= \{(C_1, \mu_{1,1}, \nu_{1,1}), (C_2, \mu_{1,2}, \nu_{1,2}), ..., (C_m, \mu_{1,m}, \nu_{1,m})\} \\ M_2 &= \{(C_1, \mu_{2,1}, \nu_{2,1}), (C_2, \mu_{2,2}, \nu_{2,2}), ..., (C_m, \mu_{2,m}, \nu_{2,m})\} \\ &\vdots \\ M_n &= \{(C_1, \mu_{n,1}, \nu_{n,1}), (C_2, \mu_{n,2}, \nu_{n,2}), ..., (C_m, \mu_{n,m}, \nu_{n,m})\} \end{aligned} \tag{8}$$

where $\mu_{i,j}$ indicates the degree by which the alternative M_i satisfies criterion C_j, $\nu_{i,j}$ indicates the degree by which the alternative M_i does not satisfy criterion C_j [5]. Considering that we use TOPSIS in Fuzzy environment, the sum of $\mu_{i,j}$ and $\nu_{i,j}$ cannot be greater than 1.

Definition 6. A fuzzy positive ideal solution is defined as

$$\begin{aligned} \tilde{A}^+ = \{&(C_1, \text{Max}\{\mu_{i,1}\}, \min\{\nu_{i,1}\}), \\ &(C_2, \text{Max}\{\mu_{i,2}\}, \min\{\nu_{2,m}\}), \\ &\vdots \\ &(C_m, \text{Max}\{\mu_{i,m}\}, \min\{\nu_{i,m}\})\}. \end{aligned} \tag{9}$$

Definition 7. A fuzzy negative ideal solution is defined as

$$\begin{aligned} \tilde{A}^- = \{&(C_1, \min\{\mu_{i,1}\}, \text{Max}\{\nu_{i,1}\}), \\ &(C_2, \min\{\mu_{i,2}\}, \text{Max}\{\nu_{i,2}\}), \\ &\vdots \\ &(C_m, \min\{\mu_{i,m}\}, \text{Max}\{\nu_{i,m}\})\}. \end{aligned} \tag{10}$$

To calculate the distance between alternatives $\tilde{A}^+$ and $\tilde{A}^-$ we define two inclusion degrees as follows:

Definition 8. The inclusion degree $D^+(M_i)$ of the positively ideal solution in alternative M_i and the inclusion degree $D^-(M_i)$ of the negatively ideal solution in alternative M_i are respectively defined as

$$D^+(M_i) = \text{Max}(I(\tilde{A}^+, M_i)), \tag{11}$$

$$D^-(M_i) = \min(I(M_i, \tilde{A}^-)) \tag{12}$$

where I denotes the inclusion degree function, see Equation 5.

Definition 9. The ranking index of alternative M_i is defined as

$$p_i = \frac{D^+(M_i)}{D^-(M_i) + D^+(M_i)}, \tag{13}$$

where $0 \leq p_i \leq 1$.

If there exists:
$i_0 \in \{1, 2, ..., n\}$ such that $p_{i_0} = Max\{p_1, p_2, ..., p_n\}$ then the alternative M_{i_0} is the best alternative, [5].

III. Proposed Approach

In this section, we describe our approach on how to use the fuzzy NFR profile to model NFRs and find out, among different design alternatives, the one which leads to a better overall satisfaction of NFRs in the system. The main idea in the approach is that different alternatives (version/type/implementation) of a feature can be selected to satisfy an NFR. However, for each feature, its impacts and side effects on other NFRs in the system should also be taken into account. For instance, in a system, there can be NFRs on security and low memory usage. To satisfy security, different encryption methods (as features) can be used leading to stronger or weaker levels of security. However, choosing a strong encryption method may also incur high memory usage, while a less strong one may have very low memory usage.

In our approach, first an NFR model of the system is constructed which contains a generic set of features; i.e., not specified which version or type of a feature is selected. In the context of the small example mentioned above, it would mean that it is just specified that an encryption method is needed in order to achieve security, but not specified which one of the available encryption methods. The output of the approach will then be that which specific version or implementation of features should be chosen considering all their side effects and impacts on other NFRs in the system.

The approach consists of the following steps:

1) First, a model of NFRs in the system along with different generic features (the concept of feature was introduced in Section II) that are intended to satisfy each NFR is constructed.
2) Different alternatives for each feature are also annotated in the model using the Alternative stereotype. The contribution and impact value for each feature alternative is specified using fuzzy variables (e.g., Low, High, etc.).
3) A model-to-text transformation is performed on the model which navigates and extracts different elements of type Feature that are used in the model along with different alternatives for each feature. The output is a list of all feature alternatives with the specification of how each one contributes to an NFR and has impacts on other NFRs, if any.
4) The output of the above step is fed as input to the fuzzy decision system which basically performs the steps and calculations described in the previous section. The output of this step is a set that identifies which alternative of each feature should be selected to have the best overall design with respect to the satisfaction of NFRs.
   To perform the decision making step, the criteria are defined in the fuzzy environment and then TOPSIS method is used. In this context, the weights of criteria here are basically fuzzy numbers (hence fuzzy decision making). Then, the collected data are analyzed in order to determine impact that features and NFRs have on each other. Finally, the decision making system is executed using Equations 5, 6, 7, 8, 9, 10, 11, 12 and 13.
   Figure 3 summarizes the steps taken in the decision making part of our approach which are based on the formulas and calculations described in the previous section.
5) Based on the identified set of feature alternatives, the NFR model is updated, using in-place model-to-model transformation technique, to reflect the exact version/type of each feature that should be considered in the design. This is done by updating the generically specified features with the exact version/type of them that should be used (making them specific now).
6) Finally, another in-place model-to-model transformation is performed on the updated model, which navigates nodes and model elements, and calculates satisfaction and deviation indicator values based on the characteristics of the fixed and concrete features that are selected.

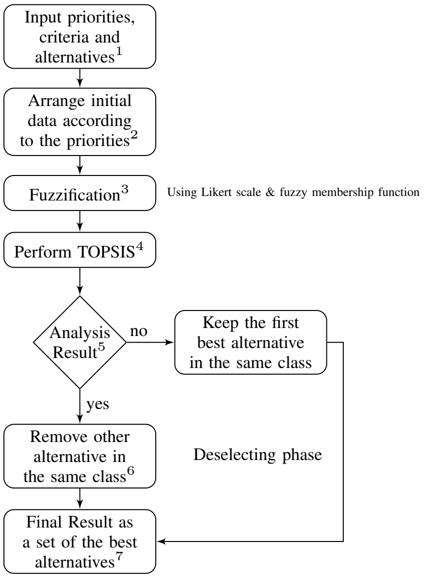
Input priorities, criteria and alternatives[1]
Arrange initial data according to the priorities[2]
Fuzzification[3]
Using Likert scale & fuzzy membership function
Perform TOPSIS[4]
Analysis Result[5]
no
Keep the first best alternative in the same class
yes
Remove other alternative in the same class[6]
Deselecting phase
Final Result as a set of the best alternatives[7]

Fig. 3: The process of DSS

As mentioned, the contribution and impact values are assigned by using fuzzy variables. The set of fuzzy variables, which are based on Likert scale[2] [20] and whose values are defined through the membership function, is illustrated in Figure 4. For example, a Very High contribution or impact relationship in the system can be defined as a fuzzy value in the $[0.7, 1]$ range.

The exact value which is selected from this range will depend on the priority of the NFRs (serving as criteria in the decision making approach). For example, an NFR with higher priority will have a value closer to 1 from $[0.7, 1]$ range than another NFR with a lower priority.

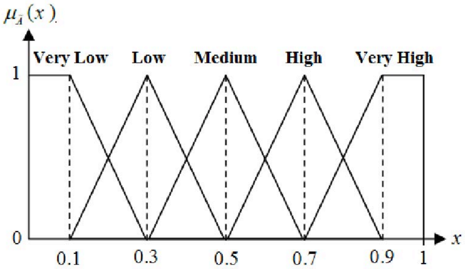

Fig. 4: Membership function of the fuzzy set

IV. Implementation and Application Example

In this section, we discuss the implementation of the introduced approach along with an example which demonstrates how it is applied.

The profile described earlier is implemented using Papyrus [21]. Papyrus is a modeling environment provided in Eclipse

[2] Likert scale is a 5-point scale used mainly in questionnaires to obtain preferences or degree of agreement on a subject.

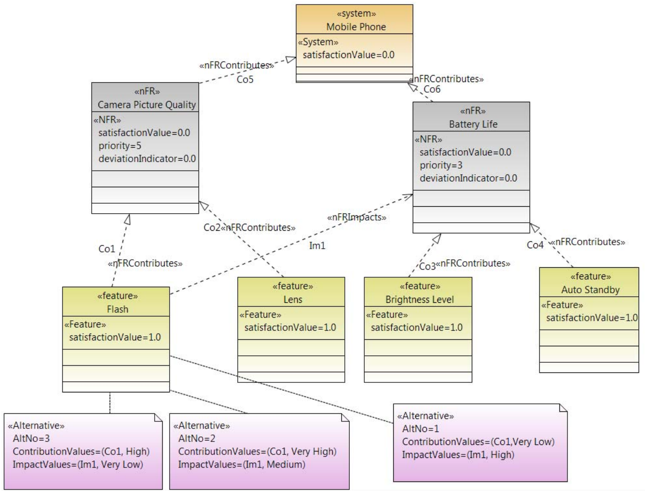

Fig. 5: Application example for a mobile phone (before analysis)

which particularly supports UML and UML-based languages and profiles (e.g., SysML and MARTE). Using the defined NFR profile in Papyrus, an NFR model of the system is then built. Figure 5 shows the application of the profile in designing the NFR model of a mobile phone. Two NFRs are defined in this partial example: one on the quality of the taken camera picture and the other one on the battery life. These two NFRs are captured and shown in the model as Camera Picture Quality (CPQ) and Battery Life (BL). There are two features that contribute to CPQ (the quality of taken photos): usage of a flash and type of the lens. Also two features are considered for Battery Life (BL): adjustment of screen's brightness level and auto standby feature. The contribution of each feature to its parent NFR is marked using NFRContributes stereotype (links named Co1, Co2, Co3, and Co4). In this system, the use of flash has impact on battery. The impact of the Flash feature on Battery Life is therefore specified using NFRImpacts stereotype on link Im1. As can be seen in this figure, for the Flash feature, there are three alternatives with different contribution and impact values as annotated in the model (for simplicity, only the alternatives for the Flash feature are shown here; and for the other features in the model, three alternatives are also considered and similarly annotated).

To extract the necessary information from the model to perform the fuzzy TOPSIS decision making process, a model-to-text model transformation is implemented using Acceleo [22]. The information that is extracted basically contains the number of alternatives for each feature and their characteristics (contribution and impact values) and NFRs in the system. NFRs are then considered as the main criteria in decision making process.

Table II,III, and IV show what sort of information is extracted and collected. In Table II, the contributions of different flash and lens alternatives on Camera Picture Quality (CPQ), as one of the criteria (C_1), are shown. Table III shows similar information for the contribution of Brightness Level (BRL) and Auto Standby (AS) features to the Battery Life (BL) criteria. Table IV captures the side effects of features on other NFRs; in the example, there is only the impact of `Flash` on Battery Life (BL, as the second criteria). This also implies that in the calculations, the impact of other features on BL will be 0.

Alternative	Feature	Contribution to CPQ(C_1)
M_1	Flash 1	Very Low
M_2	Flash 2	Very High
M_3	Flash 3	High
M_4	Lens 1	High
M_5	Lens 2	Very High
M_6	Lens 3	Low

TABLE II: Contribution of different feature alternatives to CPQ (Criteria 1)

Alternative	Feature	Contribution to BL(C_2)
M_7	BRL 1	Low
M_8	BRL 2	Medium
M_9	BRL 3	Very Low
M_{10}	AS 1	High
M_{11}	AS 2	Medium
M_{12}	AS 3	Very High

TABLE III: Contribution of different feature alternatives to BL (Criteria 2)

Feature	Impact on BL(C_2)
Flash 1	High
Flash 2	Medium
Flash 3	Very Low

TABLE IV: Impact of different features on BL

By using the Likert scale and fuzzy membership values shown in Figure 4 we are able to assign the weights of the criteria. The decision maker wants to choose the alternatives which satisfy both C_1 , C_2. Since in this case we have more than one alternative in each class (for example Flash 1,2, and 3 in the class of Flashes) we need to have an extra step in the decision making algorithm which deselects the other alternatives in the same class if decision making system proposes an alternative in that class (corresponding to step 6 in Figure 3). In other words, for example, we want to avoid situations where usage of two flashes is proposed.

The specifications of the alternatives and criteria for IFS based on Equation 8 are calculated as follows:

$$
\begin{aligned}
M_1 &= \{(C_1, 0.3, 0.7), (C_2, 0.1, 0.9)\} \\
M_2 &= \{(C_1, 1, 0), (C_2, 0.3, 0.7)\} \\
M_3 &= \{(C_1, 0.9, 0.1), (C_2, 0.7, 0.3)\} \\
M_4 &= \{(C_1, 0.9, 0.1), (C_2, 0, 0)\} \\
M_5 &= \{(C_1, 1, 0), (C_2, 0, 0)\} \\
M_6 &= \{(C_1, 0.5, 0.5), (C_2, 0, 0)\} \\
M_7 &= \{(C_1, 0, 0), (C_2, 0.3, 0.7)\} \\
M_8 &= \{(C_1, 0, 0), (C_2, 0.5, 0.5)\} \\
M_9 &= \{(C_1, 0, 0), (C_2, 0.1, 0.9)\} \\
M_{10} &= \{(C_1, 0, 0), (C_2, 0.7, 0.3)\} \\
M_{11} &= \{(C_1, 0, 0), (C_2, 0.5, 0.5)\} \\
M_{12} &= \{(C_1, 0, 0), (C_2, 0.9, 0.1)\}
\end{aligned}
$$

In the above calculations, as an example, M_1 which represents Flash 1, affects C_1 (Criteria 1 which is CPQ) with a Very Low degree. Therefore, in the tuple $(C_1, 0.3, 0.7)$, the second parameter which indicates the degree by which the alternative M_1 satisfies criterion C_1 is assigned 0.3 (fuzzy value of Very Low); hence, its non-inclusion degree is 0.7 $(1-0.3)$. Moreover, Flash 1 impacts Battery Life (BL), which is our second criteria, with a Very High degree. This negative effect, is capture in M_1 with the $(C_2, 0.1, 0.9)$ tuple, whose non-inclusion degree is assigned the value of 0.9 (Very High considering the priority level of BL which is 3) which results in its inclusion degree of 0.1 $(1-0.9)$.

By using Equation 9 and 10 we get:

$$\tilde{A}^+ = \{(C_1, 0.9, 0), (C_2, 0.9, 0)\}$$
$$\tilde{A}^- = \{(C_1, 0, 0.7), (C_2, 0, 0.9)\}$$

According to Equation 11 and 12 the inclusion degrees of $\tilde{A}^+$ and $\tilde{A}^-$ are calculated as in Table V. The ranking index of the alternatives is then obtained by Equation 13 as shown in Table VI.

	M_1	M_2	M_3	M_4	M_5	M_6
$I(\tilde{A}^+, M_i)$	0.25	0.7	0.85	0.975	0.775	0.775
$I(M_i, \tilde{A}^-)$	0.9	0.45	0.3	0.4	0.35	0.825
	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}
$I(\tilde{A}^+, M_i)$	0.775	0.55	0.35	0.65	0.55	0.75
$I(M_i, \tilde{A}^-)$	0.7	0.6	0.8	0.5	0.6	0.4

TABLE V: The inclusion degrees of $\tilde{A}^+$ in M_i and M_i in $\tilde{A}^-$

p_1	p_2	p_3	p_4	p_5	p_6
0.217	0.608	0.739	0.709	0.688	0.484
p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
0.525	0.478	0.304	0.565	0.478	0.652

TABLE VI: The ranking index

We select the features that have the highest ranking index and propose the result as a set of features. As mentioned, other features in the same class must be removed from the proposed set (step 6 in Figure 3). In other words, we choose a feature with highest ranking index in each same class. For example in our case we removed Flash 2 with ranking index $p_2 = 0.608$ because the Flash 3 with higher ranking index ($p_3 = 0.739$) is already selected for the class of the Flash.

Based on the aforementioned calculations (step 7 in Figure 3), the following set is suggested as an optimal set of features for the NFR model:

$$\text{DSS=}\Big[\text{Flash 3, Lens 1, AutoStandby 3, Brightness Level 1}\Big]$$

To perform the TOPSIS fuzzy decision making calculations, we have used Matlab in combination with Stopsis project [23].

After identifying the set of features that leads to higher and better overall satisfaction of NFRs, the model is updated with the selected set of features. To do this, we have implemented and used an in-place model-to-model transformation using QVT Operational language (QVTo) [24], [25] in Eclipse. The updated model is shown in Figure 6.

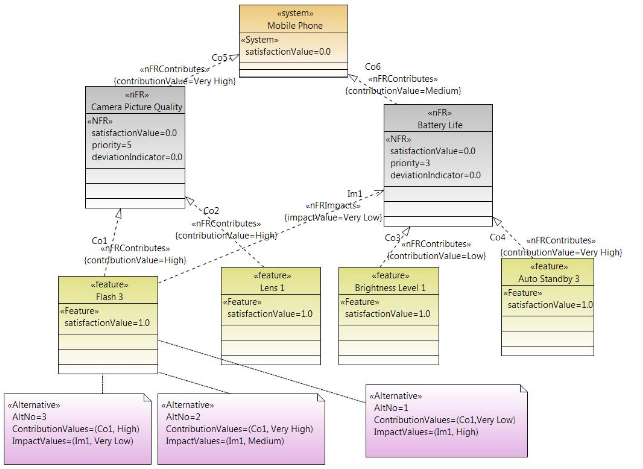

Fig. 6: NFR model updated with the selected features - before analysis

When the set of identified features are set in the model, another in-place model-to-model transformation is performed on the model which uses the fuzzy values of fuzzy variables and calculates the satisfaction and deviation indicator values. The model of the system after such analysis and calculations is shown in Figure 7.

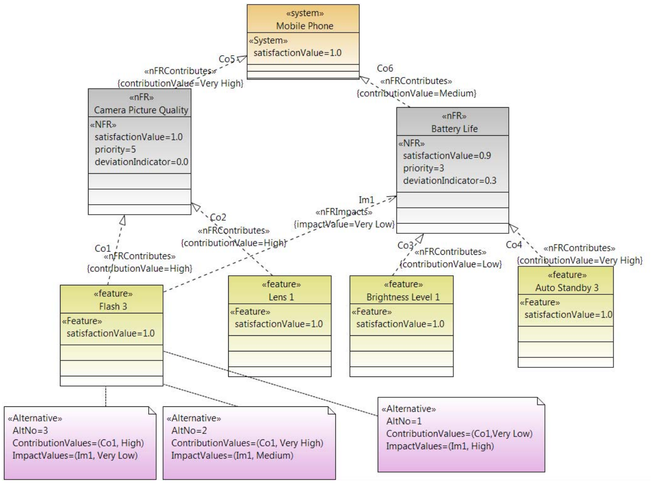

Fig. 7: NFR model updated with the selected features - after analysis

V. Related Work

One of the important works in analysis of NFRs is the NFR Framework which is introduced in [10]. It is a goal-oriented approach which constructs Softgoal Interdependency Graph (SIG) for representation of NFRs. The analysis in NFR Framework is basically a qualitative analysis of NFRs. In NFR

Framework, entities that function to satisfy NFRs are referred to as *Operationalizaiton*. Different types of dependencies and contributions of NFRs in NFR Framework are specified by *make*, *hurt*, *help*, *break*, and *undetermined* relationship types. Quantified SIG (QSIG) is introduced in [11] to enable a quantitative analysis of NFRs and their tradeoffs. Our approach is inspired by QSIG for providing a set of rules to enable quantitative calculations, but QSIG tries to maintain the same structure for NFRs as in NFR Framework, while we provide a different solution to better capture NFR relationships at the model level using UML. Moreover, we introduce the concept of deviation indicator value and also automate the analysis in our approach by using model transformation techniques. We also clearly distinguish between the priority and impact of an NFR while in QSIG these two concepts are somewhat merged as one. In [26], Fuzzy Qualitative and Quantitative SIG (FQQSIG) is introduced which is a model combining qualitative and quantitative methods. In FQQSIG, fuzzy variables are used to represent the assessments that are performed by NFR experts, and then a RAndom GEneration (RAGE) defuzzification process is done to quantify these assessments. The analysis in FQQSIG model is done following seven steps. Some of the key differences between our approach and FQQSIG is that FQQSIG does not deal explicitly with end-user priorities for NFRs (as QSIG), uses a pre-defined three-layer classification of NFRs for trustworthy software systems, and also does not focus on how to model NFRs and assumes a representation style and structure as NFR Framework. As for the application of UML for modeling NFRs in specific domains, in [27], we provide an approach for modeling NFRs in telecommunication systems as well as provide some background on challenges and issues with the management of NFRs in large organizations. For a more detailed introduction to NFRs and their challenges in software systems, our works in [2] and [27] can be consulted.

Application of fuzzy TOPSIS method: As we mentioned earlier the theory of fuzzy provides an appropriate tool solving the problem with uncertainties and complex data. The fuzzy TOPSIS method based on fuzzy sets is one of the most suitable approaches for solving different type of multi criteria multi objective decision making problem. Fuzzy TOPSIS can be applied to various branches of engineering, medical decision making, stock investment decision making, etc. In [28], TOPSIS is used in making decisions to select the best maintenance activity for cooling system on heavy trucks according to company's policies. In this case, 15 various criteria have been identified and 5 maintenance activities have been proposed. The result obtained by Fuzzy TOPSIS shows that preventive maintenance is the best maintenance activity that should be performed for sightly component. As another example in [29], supplier selection is demanded which is inadequate for dealing with the imprecise and vague nature of linguistic assessment. After determining the criteria that affect the supplier selection decisions, the results for both crisp and fuzzy TOPSIS methods are presented. In this case the decision maker identified 20 different decision-making criteria such as Joint Decision-Making (JDM), Information Sharing (IS), Use of Information Tools (UIT) and Resource Sharing (RS).The result obtained by Fuzzy TOPSIS shows that IS is the best decision [29].

Analytic Hierarchy Process (AHP): AHP [30], [31] is another well-known technique for solving multi criteria decision making problems. AHP is able to solve problems which contain conflicting factors, criteria containing imprecision or vagueness inherent in the information. The structure of AHP is based on decomposition of problems into hierarchies. The number of levels in the hierarchy depends on problem complexities which combine with the decision maker's model of the initial problem. AHP uses matrix algebra for classification the factors to arrive at a mathematically optimal solution [32]. The main difference between AHP and TOPSIS is in how the evaluation of criteria is done. TOPSIS is claimed to be better in regard to changes of alternatives and criteria [33]. It is also agile and more scalable with respect to the number of criteria and alternatives [33], [34].

VI. Conclusion and Future Work

In this paper, we introduced an approach for model-based tradeoff analysis of non-functional requirements. The approach uses a fuzzy extension of TOPSIS decision making method in order to evaluate different design alternatives and identify one which leads to better overall satisfaction of NFRs. By using a modeling solution based on UML profiles, NFRs and their relationships and interdependencies are captured and visualized. This also helps to treat NFRs as first-class citizens facilitating their maintenance throughout the lifecycle of a software product and with the vision of incorporating them with functional models. The analyses in our approach are automated using model-transformation techniques. Beside identifying an optimal set of alternatives, the other values that are calculated as the result of the analysis approach can be used in different contexts and for different purposes. For instance, in [35] we have shown how the calculated deviation indicator value can be used in software testing and to prioritize test cases.

The solution that we proposed in this paper tackles one of the very important challenges of NFRs which is the analysis of complex relationships and interdependencies that exist among them. It is quite difficult to take into account and analyze all these different dependencies and impacts in a manual fashion specially when dealing with larger systems. Automation of the analysis helps to alleviate this challenge. Our approach can be particularly interesting in the context of component-based development, when a system is built of already existing components. In which case, it helps to identify which components should be chosen to build a system with 'better' satisfaction of its NFRs; in other words, our approach can be used in optimizing system designs w.r.t. NFRs. Similarly, application of our proposed approach in software product lines, where each product is made of various features would be a future direction of this work to further investigate on. With respect to modeling, a more compact and efficient way for capturing different alternatives as part of the NFR model may also be considered as an improvement of the modeling part of the approach in future works.

Our approach also provides several interesting potentials for future extensions. As mentioned, the value of a fuzzy variable can also be defined as a set. In this work to perform analysis, we considered a specific value from the set for each fuzzy variable (e.g., 0.9 for 'Very High'). It would be interesting to extend the approach to enable calculations and analysis when a complete set and range of values is assigned to a fuzzy variable as its value; i.e., a fuzzy set-based analysis. Moreover, in this work, we were only interested in the best alternative as the output of the decision making process. However, the approach has the capability to identify, for example, the second or third best alternatives, if needed. This is also another advantage of using TOPSIS over AHP (since it is more complicated in AHP). Inclusion of different multi-criteria decision making methods such as AHP as a customizable decision making engine, for instance, could be another direction of this work.

Acknowledgements

This work has been supported by the Swedish Knowledge Foundation (KKS) through the ITS-EASY industrial research school [36] at Mälardalen University.

References

[1] M. Glinz, "On non-functional requirements," in *15th IEEE International Requirements Engineering Conference*, New Delhi, India, October 2007, pp. 21–26.

[2] M. Saadatmand, A. Cicchetti, and M. Sjödin, "Model-based trade-off analysis of non-functional requirements: An automated uml-based approach," *International Journal of Advanced Computer Science*, vol. 3, no. 11, pp. 575–588, November 2013. [Online]. Available: http://www.es.mdh.se/publications/3375-

[3] J. Wegener and M. Grochtmann, "Verifying timing constraints of real-time systems by means of evolutionary testing," *Real-Time Syst.*, vol. 15, no. 3, pp. 275–298, Nov. 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1008096431840

[4] M. Saadatmand, A. Cicchetti, and M. Sjodin, "Toward model-based trade-off analysis of non-functional requirements," in *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*, Sept 2012, pp. 142–149.

[5] C. YU and Y. Luo, "A fuzzy optimization method for multi-criteria decision making problem based on the inclusion degrees of intuitionistic fuzzy sets," *Journal of Information and Computing Science*, pp. 146–152, 2008.

[6] B. Selic, "A systematic approach to domain-specific language design using uml," in *Object and Component-Oriented Real-Time Distributed Computing, 2007. ISORC '07. 10th IEEE International Symposium on*, may 2007, pp. 2 –9.

[7] I. Weisemoller and A. Schurr, "A comparison of standard compliant ways to define domain specific languages," in *Models in Software Engineering*, ser. Lecture Notes in Computer Science, H. Giese, Ed. Springer Berlin Heidelberg, 2008, vol. 5002, pp. 47–58. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-69073-3_6

[8] OMG SysML Specifcation, http://www.sysml.org/specs.htm, Accessed: December 2014.

[9] MARTE specification, http://www.omgmarte.org, Accessed: December 2014.

[10] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, ser. International Series in Software Engineering. Springer, October 1999, vol. 5.

[11] T. Marew, J.-S. Lee, and D.-H. Bae, "Tactics based approach for integrating non-functional requirements in object-oriented analysis and design," *The Journal of Systems and Software*, vol. 82, pp. 1642–1656, October 2009.

[12] P. G. W. Keen, "Decision support systems : a research perspective," *Report (Alfred P. Sloan School of Management. Center for Information Systems Research) ; no. 54*, 1980. [Online]. Available: http://hdl.handle.net/1721.1/47172

[13] L. Zadeh, "Fuzzy sets," *Information and Control*, pp. 338–353, 1965.

[14] K. T. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87 – 96, 1986. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165011486800343

[15] Baczynski.M and Jayaram.B, *Fuzzy Implications*, 2008.

[16] J. Yang and J. Watada, "Fuzzy clustering analysis of data mining: Application to an accident mining system," *International Journal of Innovative Computing, Information and Control*, pp. 5715–5724, 2012.

[17] B. D. Baets and E. E. Kerre, "Fuzzy relational compositions," pp. 109–120, 1993.

[18] K. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, pp. 87–96, 1986.

[19] J. Fan and W. Xie, "Subsethood measure: new definitions. fuzzy sets and systems," *Fuzzy Sets and Systems*, pp. 201–209, 1999.

[20] L. R, *A Technique for the Measurement of Attitudes*. Archives of Psychology, 1932.

[21] MDT Papyrus, http://www.eclipse.org/modeling/mdt/papyrus/, Accessed: November 2014.

[22] Acceleo (model-to-text transformation language), http://www.eclipse.org/acceleo/, Accessed: November 2014.

[23] Stopsis: Fuzzy similarity based fuzzy TOPSIS., http://www.mathworks.com/matlabcentral/fileexchange/36323-stopsis, Accessed: November 2014.

[24] QVT Operational Language, http://projects.eclipse.org/projects/modeling.mmt.qvt-oml, Accessed: November 2014.

[25] QVT Specification, http://www.omg.org/spec/QVT/, Accessed: November 2014.

[26] M.-X. Zhu, X.-X. Luo, X.-H. Chen, and D. D. Wu, "A non-functional requirements tradeoff model in trustworthy software," *Elsevier Journal of Information Sciences*, vol. 191, pp. 61–75, May 2012.

[27] M. Saadatmand, A. Cicchetti, and M. Sjödin, "Uml-based modeling of non-functional requirements in telecommunication systems," in *The Sixth International Conference on Software Engineering Advances (ICSEA 2011)*, Barcelona, Spain, October 2011.

[28] S. Tahvili, J. Österbrg, S. Silvestrov, and J. Biteus, "Solving complex maintenance planning optimization problems using stochastic simulation and multi-criteria fuzzy decision making," in *The 10th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences (ICNPAA 2014)*, July 2014, pp. 766–775.

[29] R. K. Shuklaa, D. Gargb, and A. Agarwal, "An integrated approach of fuzzy ahp and fuzzy topsis in modeling supply chain coordination," *Production & Manufacturing Research: An Open Access Journal*, vol. 2, no. 1, pp. 415–437, June 2014.

[30] T. L. Saaty, "How to make a decision: The analytic hierarchy process," *European Journal of Operational Research*, vol. 48, no. 1, pp. 9 – 26, 1990, desicion making by the analytic hierarchy process: Theory and applications. [Online]. Available: http://dx.doi.org/10.1016/0377-2217(90)90057-I

[31] ——, "Decision making with the analytic hierarchy process," *International Journal of Services Sciences*, vol. 1, no. 1/2008, pp. 83–98, 2008. [Online]. Available: http://dx.doi.org/10.1504/IJSSci.2008.01759

[32] B. B. Gursoy, O. Mason, and S. Sergeev, "The analytic hierarchy process, max algebra and multi-objective optimisation," *Linear Algebra and its Applications*, vol. 438, no. 11, pp. 2911–2928, April 2013. [Online]. Available: http://dx.doi.org/10.1016/j.laa.2012.11.020

[33] F. R. L. Junior, L. Osiro, and L. C. R. Carpinetti, "A comparison between fuzzy {AHP} and fuzzy {TOPSIS} methods to supplier selection," *Applied Soft Computing*, vol. 21, no. 0, pp. 194 – 209, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494614001203

[34] I. Ertuğrul and N. Karakaşoğlu, "Comparison of fuzzy ahp and fuzzy topsis methods for facility location selection," *The International Journal of Advanced Manufacturing Technology*, vol. 39, no. 7-8, pp. 783–795, 2008. [Online]. Available: http://dx.doi.org/10.1007/s00170-007-1249-8

[35] M. Saadatmand and M. Sjodin, "On combining model-based analysis and testing," in *Information Technology: New Generations (ITNG), 2013 Tenth International Conference on*, April 2013, pp. 260–266.

[36] ITS-EASY post graduate industrial research school for embedded software and systems, http://www.mrtc.mdh.se/projects/itseasy/, Accessed: August 2014.