01 Jan 1994

A Fuzzy Logic-Based Foundation for Analyzing Imprecise Conflicting Requirements

J. Yen

Xiaoqing Frank Liu
Missouri University of Science and Technology, fliu@mst.edu

Recommended Citation

J. Yen and X. F. Liu, "A Fuzzy Logic-Based Foundation for Analyzing Imprecise Conflicting Requirements," *Proceedings of the 3rd IEEE Conference on Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence.*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1994.
The definitive version is available at https://doi.org/10.1109/FUZZY.1994.343889

A Fuzzy Logic-based Foundation for Analyzing Imprecise Conflicting Requirements

John Yen, *Member IEEE* and Xiaoqing Liu

***Abstract*— Imprecise requirements are represented by the canonical form in test-score semantics. The concepts of feasibility, satisfiability, and specificity are formalized based on the fuzzy sets. The relationships between requirements are classified to be conflicting and cooperative. A feasible overall requirement can thus be formulated based on the trade-off analysis of the conflicting requirements by using fuzzy multi-criteria optimization technique.**

I. Introduction

The requirements represent the criteria against which the acceptability of a realization of a target system is judged. Lack of precision leads to the difficulty to develop a procedure for determining a realization do or does not meet some particular requirement [1]. Thus a challenge with requirement engineering is that the requirements to be captured are usually described in qualitative terms which are imprecise in nature. Actually, as Balzer et al. have stated, informality is an inevitable and ultimately desirable feature of the specification process [2]. However, most existing specification methodologies either require that the requirements be stated precisely, such as in formal specification methodologies (e.g., Z [8], Larch [5], etc.), or convert informal requirements into formal ones (e.g., SAFE project [2] and Requirement Apprentice [6]). Therefore they do not capture the impreciseness of the requirements.

Another challenge with requirement engineering is that requirements often conflict with each other. However existing specification methods consider that a requirement specification, which contains conflicting requirements, to be inconsistent, and should be avoided since requirements are specified as crisp ones [1]. Moreover, it is very difficult, if not impossible, to specify a trade-off between conflicting requirements if these requirements are specified to be crisp.

In this paper, soft functional requirements are formulated based on the canonical form in test-score semantics [12]. One of purposes to conduct requirement analysis is to make the acquired requirements feasible. The concepts of feasibility, satisfiability, and specificity are formalized based on fuzzy sets. The relationships between requirements are classified to be conflicting and cooperative. Conflicting requirements can not be satisfied completely at the same time and a trade-off needs to be developed. The trade-offs are analyzed using fuzzy multi-criteria optimization techniques [13, 3] to formulate a feasible overall requirement and to facilitate to find a better design.

In the next section, we introduce the notation and formulation of soft requirements. The concepts of feasibility, satisfiability, and specificity are formalized in section 3. The classification of relationships between requirements is presented in section 4. Section 5 discusses how to use fuzzy multi-criteria optimization techniques for combining conflicting requirements. Finally as an example, a specification of conflicting requirements in expert system R1/SOAR is presented in section 6.

II. Fundamentals

A target system T can be specified as a set of state transitions [10]. Let ST be the set of plausible state transition $< s_1, s_2 >$ that can be performed by T, where s_1 is a before state, s_2 is a after state. If T is implemented, the working system is called its realization. A before state s_1 may have more than one plausible after state s_2 such that $< s_1, s_2 > \in ST$. It indicates that there may be many plausible realizations for T. For a given before state s_1, let $perform(T, s_1) = \{s_2 \mid < s_1, s_2 > \in ST\}$. The cardinality of the set $perform(T, s_1)$ may thus be greater than one. Assume that P is a realization of T, we use $exec(P, s_1)$ to denote the after state which can be obtained by P when given a before state s_1.

Definition 1 *(Rigid Functional Requirement [11]): A precise functional requirement R of the target system is a pair of formula $< \varphi_1, \varphi_2 >$ where φ_1 is a*

Authors are working at the center for fuzzy logic and intelligent systems research and department of computer science in Texas A&M university at college station in Texas. This research is supported by NSF Young Investigator Award IRI-9257293.

0-7803-1896-X/94 $4.00 ©1994 IEEE

precondition and φ_2 is a postcondition, such that

$$\forall < s_1, s_2 > \in ST\ hold(\varphi_1, s_1) \implies hold(\varphi_2, s_2)$$

where "$hold(\varphi, s)$" is a boolean function which tests whether the condition φ is satisfied in state s, and $\implies$ is the logic implication.

A rigid functional requirement $< \varphi_1, \varphi_2 >$ is an essential requirement. It specifies the ***"state changes"*** that has to be achieved by a realization of the target system.

A fuzzy set $DST_R = \{< s_1, s_2 >, \mu_{DST_R} < s_1, s_2 > \}$ of desired state transitions can be defined for an imprecise requirement R, where $\mu_{DST_R} < s_1, s_2 >$ specifies the degree to which the state transition $< s_1, s_2 >$ is desired by R.

Definition 2 *(Soft Functional Requirement [11]):* ***A soft functional requirement R of the target system is specified as a pair of formula $< \varphi_1, \varphi_2 >$ where φ_1 is a soft precondition and φ_2 is a soft postcondition, such that $\forall < s_1, s_2 > \in S(DST_R)$,***

$$\mu_{fhold(\varphi_1, s_1) \overset{f}{\implies} fhold(\varphi_2, s_2)} = \mu_{DST_R} < s_1, s_2 >$$

where $S(DST_R)$ is the support of the fuzzy set DST_R, "fhold" is a function that returns the degree to which a formula φ is true in state s, and $\overset{f}{\implies}$ is the fuzzy logic implication.

Thus a soft requirement specifies the "state changes" that are desired to be achieved to some degrees by a realization of the target system. In the following discussion, B_R will denote a set of before state s_1 of R and A_R a set of after state s_2 of R such that $\mu_{DST_R} < s_1, s_2 > > 0$

A soft requirement can be represented using the *canonical form* in Zadeh's test score semantics [12]. It is established by the following theorem [11].

Theorem 1 *Let p be a proposition in its canonical form, X is A, X is a state variable in state s, and u_i is the value of X in s. Then*

$$fhold(p, s) = \mu_A(u_i)$$

In the following discussion, the requirement is referred to the soft functional requirement unless an explicit specification is made.

III. Satisfiability, Feasibility and Specificity

A soft requirement can be satisfied to different degrees by different realizations of a target system T. The satisfiability of a requirement by a realization can be viewed as the degree to which it is satisfied by the realization in the worst case.

Definition 3 *(satisfiability):*
Let $R =< \varphi_1, \varphi_1 >$ be a requirement of a target system T and P be a realization of T, the degree to which P satisfies R, denoted as $satisfiability(P, R)$, is defined as

$$\min_s (\mu_{fhold(\varphi_1, s) \overset{f}{\implies} fhold(\varphi_2, exec(P, s))}) = \alpha.$$

That a requirement is feasible implies that it can be fulfilled by some plausible realization of the target system. Thus the feasibility of a soft requirement can be viewed as the degree to which it can be satisfied by its most optimistic realization.

Definition 4 *(feasibility):*
Let R be a requirement, the feasibility of R is defined as

$$Feasibility(R) = \max_P(\min_s \mu_{DST_R} < s, exec(P, s) >).$$

We say that R_2 is more feasible than R_1 if

$$Feasibility(R_1) \leq Feasibility(R_2)$$

If a soft requirement is infeasible, i.e., its feasibility is zero, it cannot be satisfied by any possible realization of the target system. That is, for every possible realization of the target system, there exists a case such that the requirement cannot be satisfied at all.

In requirement engineering, that a requirement R_1 is more specific than another requirement R_2 implies that if R_1 can be satisfied, R_2 must be satisfied.

Definition 5 *(specificity):*
Let R_1 and R_2 are two requirements of T. We say that R_1 is more specific than R_2, that is,

$$R_1 \preceq R_2,$$

if and only if for every realization P of T,

$$satisfy(P, R_1) \leq satisfy(P, R_2).$$

Theorem 2, 3, and 4 can be directly derived from the above definitions.

Theorem 2 *Let R_1 and R_2 are two requirements of T. R_2 is more feasible than R_1 if*

$$R_1 \preceq R_2.$$

Theorem 3 *Let R_1 and R_2 be two requirements, if $R_1 \preceq R_2$, then*

$$Infeasible(R_2) \implies Infeasible(R_1)$$

Theorem 4 *Assume that $R_1 =< \varphi_1^1, \varphi_2^1 >$ and $R_2 =< \varphi_1^2, \varphi_2^2 >$ are two soft requirements of a target system. Let $R_C =< \varphi_1^1 \wedge \varphi_1^2, \varphi_2^1 \wedge \varphi_2^2 >$ and $R_D =< \varphi_1^1 \wedge \varphi_1^2, \varphi_2^1 \vee \varphi_2^2 >$, then*

$$R_C \preceq R_i \quad and \quad R_C \preceq R_D$$

where $1 \leq i \leq 2$.

Intuitively, it means that if R_C is satisfied, both R_1 and R_2 are satisfied, and so does R_D.

IV. The Classification of Relationships between Requirements

Two soft requirements are said to conflict with each other to a degree if an increase in the degree to which one requirement is satisfied often decreases the degree to which another requirement is satisfied.

Definition 6 *(conflicting degree with respect to(wrt) a given before state)*
Assume that $R_1 =< \varphi_1^1, \varphi_2^1 >$ and $R_2 =< \varphi_1^2, \varphi_2^2 >$ be two soft requirements of a target system T. For a before state $b_k \in B_{R_1} \cap B_{R_2}$, let the set of common after states of R_1 and R_2 wrt b_k be denoted as
$A_{R_1,R_2}(b_k) = perform(T, b_k) \cap A_{R_1} \cap A_{R_2}$,
and the set of after state pairs, in which an increase in the degree to which a requirement is satisfied decreases the degree to which another requirement is satisfied, be denoted as

$$\begin{aligned} \mathcal{F}(b_k) = \{(a_1,a_2) | a_1, a_2 \in A_{R_1,R_2}(b_k), \\ and\ (fhold(\varphi_2^1, a_1) - fhold(\varphi_2^1, a_2)) \times \\ (fhold(\varphi_2^2, a_1) - fhold(\varphi_2^2, a_2)) < 0\}. \end{aligned}$$

Then the degree R_1 and R_2 are conflicting wrt the before state b_k, denoted as $conf(b_k)$, is $\frac{\|\mathcal{F}(b_k)\|}{\|A_{R_1,R_2}(b_k)\|^2}$.

An example of conflicting requirements wrt a given before state is shown in Fig. 1.

Two soft requirements are said to *completely conflicting with each other wrt a given before state* if an increase in the degree to which one requirement is satisfied always decreases the degree to which another requirement is satisfied for the before state. It can be easily shown that two requirements are completely conflicting wrt a given before state whenever their conflicting degree wrt the before state is one. An example of completely conflicting requirements wrt a before state is shown in Fig. 2.

Having defined the conflicting degree wrt a given before state, we are ready to introduce several overall conflicting measures between two requirements.

Definition 7 *(optimistic, pessimistic and average conflicting degree)*
Assume that $R_1 =< \varphi_1^1, \varphi_2^1 >$ and $R_2 =< \varphi_1^2, \varphi_2^2 >$ are two soft requirements of a target system. The optimistic conflicting degree of R_1 and R_2 is

$$opt - conf(R_1, R_2) \ = \ \min_{b_k \in B_{R_1} \cap B_{R_2}} conf(b_k).$$

The pessimistic conflicting degree of R_1 and R_2 is

$$pess - conf(R_1, R_2) \ = \ \max_{b_k \in B_{R_1} \cap B_{R_2}} conf(b_k).$$

The average conflicting degree of R_1 and R_2 is

$$avg - conf(R_1, R_2) \ = \ \frac{\sum_{b_k \in B_{R_1} \cap B_{R_2}} conf(b_k)}{\|B_{R_1} \cap B_{R_2}\|}.$$

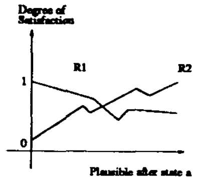

Figure 1: Conflicting Requirements wrt a Before State

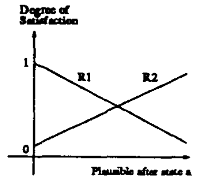

Figure 2: Completely Conflicting Requirements wrt a Before State

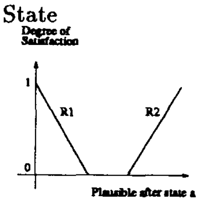

Figure 3: Disjointly Conflicting Requirements

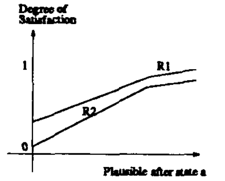

Figure 4: Completely Cooperative Requirements wrt a Before State

We now introduce the following two definitions to measure the overall degrees to which two requirements are completely conflicting.

Definition 8 *(existentially completely conflicting requirements)*
Let $R_1 =< \varphi_1^1, \varphi_2^1 >$ and $R_2 =< \varphi_1^2, \varphi_2^2 >$ are two soft requirements. We say R_1 and R_2 are existentially completely conflicting with each other if and only if

$$pess - conf(R_1, R_2) \ = \ 1.$$

That is, they completely conflict with each other wrt at least one before state in $B_{R_1} \cap B_{R_2}$.

Definition 9 *(universally completely conflicting requirements)*
Let $R_1 =< \varphi_1^1, \varphi_2^1 >$ and $R_2 =< \varphi_1^2, \varphi_2^2 >$ are two soft requirements. We say R_1 and R_2 are universally completely conflicting with each other if and only if

$$opt - conf(R_1, R_2) \ = \ 1.$$

That is, they completely conflict with each other wrt all before states in $B_{R_1} \cap B_{R_2}$.

Sometimes two soft requirements cannot be satisfied at all at the same time. That is, if one requirement is satisfied to some degree, another cannot be satisfied at all, and vice versa.

Definition 10 *(disjointly conflicting requirements)*
Two soft requirement R_1 and R_2 are said to be disjointly conflicting if and only if

$$S(DST_{R_1}) \bigcap S(DST_{R_2}) = \emptyset,$$

where $S(DST_R)$ denotes the support of the fuzzy set DST_R which specifies the degrees to which state transitions are desired for a soft requirement R.

Thus if R_1 and R_2 are disjointly conflicting requirements, then

$$\forall s_1, s_2\ \mu_{DST_{R_1}}(s_1, s_2) > 0 \implies \mu_{DST_{R_2}}(s_1, s_2) = 0$$

and vice versa. Therefore there cannot be any trade-off between two disjointly conflicting requirements. Only one of them can be satisfied as shown in Fig. 3.

Two soft requirements often cooperate with each other. It means an increase in the degree to which one requirement is satisfied often increases the degree to which another requirement is satisfied.

Definition 11 *(cooperative degree wrt a given before state)*
Assume that $R_1 =< \varphi_1^1, \varphi_2^1 >$ and $R_2 =< \varphi_1^2, \varphi_2^2 >$ be two soft requirements of a target system T. For a before state $b_k \in B_{R_1} \cap B_{R_2}$, let the set of after state pairs, in which an increase in the degree to which a requirement is satisfied also increases the degree to which another requirement is satisfied, be denoted as

$$\begin{aligned}\mathcal{G}(b_k) = \{&(a_1, a_2) | a_1, a_2 \in A_{R_1,R_2}(b_k),\\ &and(fhold(\varphi_2^1, a_1) - fhold(\varphi_2^1, a_2))\times\\ &(fhold(\varphi_2^2, a_1) - fhold(\varphi_2^2, a_2)) \geq 0\}.\end{aligned}$$

Then the degree R_1 and R_2 are cooperative wrt before state b_k, denoted as $coop(b_k)$, is $\frac{\|\mathcal{G}(b_k)\|}{\|A_{R_1,R_2}(b_k)\|^2}$.

Two soft requirements are said to *completely cooperative with each other wrt a given before state* if an increase in the degree to which one requirement is satisfied always increases the degree to which another requirement is satisfied for the before state. It can be easily shown that two requirements are completely cooperative wrt a before state whenever their cooperative degree wrt the before state is one. An example of completely cooperative requirement wrt a before state is shown in Fig. 4. Two requirements are called *completely cooperative* if they completely cooperative with each other wrt all before states in $B_{R_1} \cap B_{R_2}$.

We now introduce several overall cooperative measures between two requirements.

Definition 12 *(optimistic, pessimistic, and average cooperative degree)*
Assume that $R_1 =< \varphi_1^1, \varphi_2^1 >$ and $R_2 =< \varphi_1^2, \varphi_2^2 >$ are two soft requirements of a target system. The pessimistic cooperative degree of R_1 and R_2 is defined as

$$pess - coop(R_1, R_2) = \min_{b_k \in B_{R_1} \cap B_{R_2}} coop(b_k).$$

The optimistic cooperative degree of R_1 and R_2 is defined as

$$opt - coop(R_1, R_2) = \max_{b_k \in B_{R_1} \cap B_{R_2}} coop(b_k).$$

The average cooperative degree of R_1 and R_2 is defined as

$$avg - coop(R_1, R_2) = \frac{\sum_{b_k \in B_{R_1} \cap B_{R_2}} coop(b_k)}{\|B_{R_1} \cap B_{R_2}\|}.$$

There is a dual relationship between conflicting degree and conflicting degree.

Theorem 5 *Let R_1 and R_2 be two requirements, then*

1. $pess - conf(R_1, R_2) = 1 - opt - coop(R_1, R_2)$
2. $opt - conf(R_1, R_2) = 1 - pess - coop(R_1, R_2)$
3. $avg - conf(R_1, R_2) = 1 - avg - conf(R_1, R_2)$

This theorem can be proved easily according to the definitions given before.

The relationship between disjointly conflicting requirements and cooperative requirements is established by the following theorem.

Theorem 6 *If two requirements are disjointly conflicting, they can not be cooperative.*

This theorem can be proved easily by the direct application of the definitions of conflicting and cooperative requirements given above.

V. Using Fuzzy Multi-criteria Optimization Techniques for Combining Conflicting Requirements

A. Compromise Operators

Averaging [3] and compensatory [13] operators are often used to combine multi-criteria in fuzzy multicriteria optimization. However many averaging operators are not compensatory and vice versa. In requirement engineering, a trade-off between conflicting requirements usually is a compromise which is compensatory. Thus the compromise operator is developed to combine conflicting requirements for trade-off analysis.

Definition 13 *(Averaging) [3]*
An operator M is said to be an averaging operator if and only if it satisfies:

1. $min((fhold(p_i, s), \ldots, fhold(p_n, s)) \leq M((fhold(p_i, s), \ldots, fhold(p_n, s))$; *and*

   $M((fhold(p_i, s), \ldots, fhold(p_n, s)) \leq max((fhold(p_i, s), \ldots, fhold(p_n, s))$;

   M is different from either min or max.
2. $M(1, 1, \ldots, 1) = 1$; $M(0, 0, \ldots, 0) = 0$

3. $\forall x_1, \ldots, x_n, y_1, \ldots, y_n$, *if* $(\forall j, x_j \geq y_j)$, *then* $M(x_1, \ldots, x_n) \geq M(y_1, \ldots, y_n)$.

The resulting trade-offs of an average operator lie between the most optimistic lower bound and the most pessimistic upper bound. The examples of averaging operator includes arithmetic mean, geometric mean, the two operators M_1 and M_2 such that:

$$M_1(x,y) = \frac{min(x,y)}{1-|x-y|} \quad M_2(x,y) = \frac{max(x,y)}{1+|x+y|}.$$

Definition 14 *(Compensation) [13]*
Suppose that $x_1, x_2, \ldots, x_n$ *be real numbers in* $[0,1]$ *and* f *be a function that maps to a real number in* $[0,1]$. *We say* f *is compensatory iff given*

$$f(x_1, \ldots, x_n) = k,$$

$\forall\, 0 \leq i \leq n, \forall \Delta x_i$ *such that* $0 \leq x_i + \Delta x_i \leq 1$, *there exists* $j \neq i$ $(1 \leq j \leq n)$ *and* Δx_j *such that*

$$0 \leq x_j + \Delta x_j \leq 1, and$$

$$f(\ldots, x_i + \Delta x_i, \ldots, x_j + \Delta x_j, \ldots) = k.$$

Thus the "min", a t-norm operator, and the "max", a t-conorm operator, are not compensatory. The averaging operator M_1 described above is also not compensatory. For a compensatory operator, a decrease in one operand can be compensated by an increase in another operand.

Definition 15 *(Compromise)*
An operator is said to be a compromise operator if and only if it is both averaging and compensate operator.

In the following context, C denotes a compromise operator. It realizes the trade-offs by allowing compensation between requirements. The resulting compromise is between the minimal and maximal degree of membership of the aggregated fuzzy sets. The arithmetic mean is an example of the C operator.

B. Combine Requirements using the Compromise Operator

The intended meaning of a soft condition is usually complex in nature. To represent the meaning of a complex term used in a soft condition, we often need to define the term using propositions regarding related variables whose values can be easily obtained. For example, the definition of a fuzzy proposition p may be an aggregation of other fuzzy propositions $\{p_1 \ldots p_k\}$, each of which assumes the canonical form of $p_i \rightarrow X_i$ *is* A_i, where X_i is a variable in the domain of U_i, $U = U_1 \times \ldots \times U_k$, and $A_1, \ldots, A_k$ are k fuzzy sets in $U_1, \ldots, U_k$, respectively.

To use a compromise operator to aggregate the conflicting requirements, we define the following rule to compute the function *fhold* for aggregation.

Definition 16 *(aggregation rule)*
$fhold(\{p_1, p_2, \ldots, p_n\}, s)$
$= C(fhold(p_1, s), fhold(p_2, s), \ldots, fhold(p_n, s))$
where $p_i, i = 1, 2, \ldots, n$, are propositions, s is a state, and C is a compromise operator [3].

Thus the overall possibility distribution for representing the meaning of the soft condition p may be obtained by using the compromise operator C from Definition 16 and Theorem 1 as follows:

$$\begin{aligned} fhold(p,s) &= fhold(\{p_1, \ldots, p_k\}, s) \\ &= C(fhold(p_1, s), \ldots, fhold(p_k, s)) \\ &= C(\mu_{A_1}(u_1), \ldots, \mu_{A_k}(u_k)) \end{aligned}$$

where $u_i \in U_i$.

It is very difficult to select an appropriate operator to aggregate multiple requirements because of the variety of operator and requirements. For example, a t-conorm operator is inappropriate to disjoin conflicting requirements. A t-norm operator needs to be used instead in this case since only one of disjoin conflicting requirements can be satisfied. On the other hand, a t-conorm may be more helpful for cooperative requirements since cooperative requirements can usually be satisfied at the same time.

A lot of averaging operators which are compensatory have been developed such as harmonic mean, geometric mean, arithmetic mean, quadratic mean, symmetric summations, and parametrized averaging operators [9, 3]. The variety of compromise operators might make it hard to decide which one to use in a specific application. Eight criteria have been summarized by Zimmermann [13] for selecting the appropriate general aggregation operator. They include axiomatic strength, empirical fit, adaptability, numerical efficiency, compensation, range of compensation, aggregating behavior, and required scale level of membership functions. In terms of requirement engineering, the following additional criteria need to be taken into account: 1. The intended relationship: The operator should conform to the intended relationship between requirements and the semantic interpretation [4]; 2. Feasibility: The operator should make the combined requirement feasible and increase its feasibility. For example, since the combined requirement of disjoin conflicting requirements using compromise operator is infeasible, a compromise operator can not used to aggregate them; 3. Satisfiability: The operator should help to find a better design and realization which achieves a high satisfiability; 4. Criticality: The operator should be able to handle criticality in the requirement specification. Criticality may be expressed in either number or linguistic variable; 5. Incommensurable units: Each requirement may have a different unit of measurement.

VI. An Example

To illustrate the idea of soft requirement, let us consider the R1/SOAR expert system for computer configuration [7]. The purpose of the system is to generate a feasible configuration that is close to an optimal one. To achieve it, several rigid and soft requirements can be defined. One of its rigid requirements is that the configuration is feasible. Its soft requirements for the configuration include (1) The ordering of modules is optimal, (2) The wasted cabinet space is minimal, and (3) The extra components, which need to be added to the customer order, are least expensive.

The canonical forms of the postconditions of these soft requirements can be represented as follows:

1. Modules_Ordering (configuration) is OPTIMAL;
2. Space (Wasted_Cabinet (configuration)) is MINIMUM;
3. Cost (Extra_component_needed (configuration)) is LEAST_EXPENSIVE.

OPTIMAL, MINIMUM, and LEAST_EXPENSIVE are fuzzy sets and serve as elastic constraints on a configuration.

These requirements sometimes are conflicting with each other. R1/SOAR often can not guarantee the optimal ordering of modules and minimum wasted cabinet space for a configuration at the same time. Therefore a trade-off between them is desirable. Thus the compromise operator is used to aggregate them. The degree of truth for a configuration c generated by the system to be close to optimal (denoted by p) can thus be derived based on Theorem 1 and Definition 16 as follows:

$$\begin{aligned}
&fhold(p,c)\\
&= fhold(\{p_1,p_2,p_3\},c)\\
&= C\ (fhold(fhold(p_1,c), fhold(p_2,c), fhold(p_3,c))\\
&= C\ (\mu_{OPTIMAL}(ModulesOrdering(c)),\\
&\quad \mu_{MINIMUM}(Space(WastedCabinet(c))),\\
&\quad \mu_{LEAST\_EXPENSIVE}(Cost(ExtraCompNeeded(c))))
\end{aligned}$$

VII. Conclusion

In this paper, a systematic approach for specifying imprecise conflicting requirements using fuzzy logic has been developed. It can formulate a feasible overall requirement from conflicting requirements by using fuzzy multi-criteria optimization technique. Moreover it can help to find a better design and realization by a trade-off analysis of conflicting requirements.

VIII. Acknowledgements

We wish to thank Professor Lofti A. Zadeh for suggesting that we use fuzzy multi-criteria optimization to aggregate soft requirements. We also would thank Dr. Jonathan Lee for his early work in this research.

References

[1] Gruia-Catalin Roman, "A Taxonomy of current issues in requirements engineering", *IEEE Computer*, 14-21, April, 1985.

[2] R. Balzer, N. Goldman, and D. Wile, "Informality in program specifications", *IEEE Transaction on Software Enginering*, 4(2):94–103, 1978.

[3] D. Dubois and H. Prade, "Criteria aggregation and ranking of alternatives in the framework of fuzzy set theory", in *Studies in the Management Sciences 20*, 209-240, 1984.

[4] R. Felix and S. Reddig, "Multiple Attribute Decision Making based on Fuzzy Relationships between Objectives and its Application in Metal Forming", *Proc. of the IEEE International Conference on Fuzzy Systems*, 378-383, 1993.

[5] J.V. Guttag, J.J. Horning, and J.M. Wing, "The larch family of specification languages", *IEEE Software*, 2(5):24–36, 1985.

[6] H.B. Reubenstein and R.C. Waters, "The requirements apprentice: Automated assistance for requirements acquisition", *IEEE Transaction on Software Engineering*, 17(3), 1991.

[7] P.S. Rosenbloom, J.E. John, J. McDermott, A. Newell, and E. Orciuch, "R1-soar: An experiment in knowledge-intensive programming in a problem-solving architecture", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-7(5):561–569, September 1985.

[8] J.M. Spivey, The Z-Notation: a Reference Manual, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[9] R. Yager, "Connectives and Quantifiers in Fuzzy Sets", *Fuzzy Sets and Systems*, 40:39-75, 1991.

[10] J. Yen and J. Lee, "A task-based methodology for specifying expert systems", *IEEE Expert*, Feb. 1993.

[11] J. Yen and J. Lee, "Fuzzy Logic as a Basis for Specifying Imprecise Requirements", *Proc. of the IEEE International Conference on Fuzzy Systems*, 745-749, 1993.

[12] L.A. Zadeh, "Test-score semantics as a basis for a computational approach to the representation of meaning", *Literacy Linguistic Computing*, 1:24–35, 1986.

[13] H.-J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer Academic, MA, 1991.