The Acquisition, Analysis and Evaluation of Imprecise Requirements for Knowledge-Based Systems

John Yen, Xiaoqing Liu and Swee Hor Teh
Computer Science Department, Texas A&M University
College Station, Texas 77843, U.S.A.
email:yen@cs.tamu.edu

Abstract

In this paper, a theoretical foundation has been laid and a practical method has been developed for specifying, analyzing and evaluating the complex relationships between imprecise requirements in knowledge-based systems. Imprecise requirements are represented by the canonical form in test-score semantics. The relationships between requirements are classified to be conflicting and cooperative based on the qualitative and quantitative analysis of relationships between requirements. This kind of analysis makes it possible to formulate a feasible overall requirement from conflicting individual requirements. It also facilitates to find better trade-off strategies for conflicting requirements by using fuzzy multi-criteria optimization technique. A requirement engineering process has also been developed to incorporate imprecise requirements into the requirement analysis for knowledge-based systems.

Introduction

Several approaches exploiting requirement specification techniques have been used to develop reliable knowledge-based systems [Yen & Lee 1993a, Yen & Lee 1993b, Plant 1988, Batarekh *et al.* 1991, Tsai *et al.* 1988]. However, lack of precision leads to the difficulty for determining if a realization meets its requirements [Roman 1985]. A challenge with requirement engineering is thus 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 [Balzer, Goldman & Wile 1978]. However, most existing specification methodologies either require that the requirements be stated precisely, such as in formal specification methodologies (e.g., Z [Spivey 1987], Larch [Guttag, Horning & Wing 1985], etc.), or convert informal requirements into formal ones (e.g., SAFE project [Balzer, Goldman & Wile 1978] and Requirement Apprentice [Reubenstein & Waters 1991]. Therefore they do not capture the impreciseness of the requirements.

Another challenge with requirement engineering for knowledge-based systems is that requirements often conflict with each other [Robinson 1990]. 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 [Roman 1985]. Moreover, it is very difficult to analyze and specify a trade-off between conflicting requirements if these requirements are specified to be crisp [Robinson 1990].

In this paper, a theoretical foundation has been laid and a practical method has been developed for specifying, analyzing and evaluating the complex relationships between imprecise requirements in knowledge-based systems. Imprecise requirements are formulated based on the canonical form in test-score semantics [Zadeh 1986]. The relationships between requirements are classified to be conflicting and cooperative based on the qualitative and quantitative analysis of relationships between requirements. 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 [Zimmermann 1991, Dubois & Prade 1984] to formulate a feasible overall requirement and to find a better design. A requirement engineering process, which incorporates the analysis and specification of imprecise requirements in knowledge-based systems, has also been developed.

Imprecise Requirements

A target system T can be specified as a set of state transitions [Yen & Lee 1993a]. 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.

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. An imprecise requirement is represented as a soft requirement defined below.

Definition 1 *(Soft Requirement [Yen & Lee 1993b]): A soft 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 $\forall < s_1, s_2 > \in S(DST_R)$,*

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

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 $\overset{f}{\Longrightarrow}$ 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 [Zadeh 1986]. It has been established by the following theorem [Yen & Lee 1993b].

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*

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

Theoretical Analysis of Relationships between Requirements

There exist very complex relationships between requirements. Some of them are conflicting with each other and others may cooperate. Two soft requirements are said to conflict with each other if an increase in the degree to which one requirement is satisfied often decreases the degree to which another requirement is satisfied.

Definition 2 *(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

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

Then the degree R_1 and R_2 are conflicting wrt the before state b_k, denoted as $conf(b_k)$, is $\frac{\|\mathcal{F}(b_k)\|}{\|A_{R_1,R_2}(b_k)\| \times (\|A_{R_1,R_2}(b_k)\| - 1)}$.

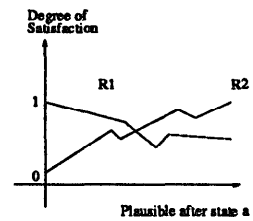

Figure 1: Conflicting wrt a Before State

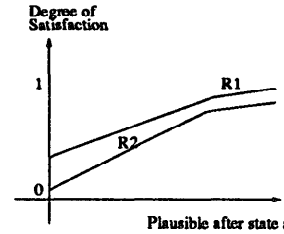

Figure 2: Completely Cooperative wrt a Before State

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.

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

Definition 3 *(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

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

The pessimistic conflicting degree of R_1 and R_2 is

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

The average conflicting degree of R_1 and R_2 is

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

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 4 *(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

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

Then the degree R_1 and R_2 are cooperative wrt before state b_k, denoted as coop(b_k), is $\frac{\|\mathcal{G}(b_k)\|}{\|A_{R_1,R_2}(b_k)\|(\|A_{R_1,R_2}(b_k)\|-1)}$.

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. 2. 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 5 *(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

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

The optimistic cooperative degree of R_1 and R_2 is defined as

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

The average cooperative degree of R_1 and R_2 is defined as

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

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

Theorem 2 *Let R_1 and R_2 be two requirements, then*

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

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

Approximate Analysis of the Relationships between Requirements

In the previous section, the theoretical foundation for analyzing the relationships between requirements has been laid. However in the real applications, it is usually difficult to specify all before states and after states for a target system. Thus we need to explain how to apply the theoretical results obtained in the previous section for the real applications.

First of all, domain experts can provide the qualitative specification of the relationships between requirements based on their expertise in the area. For example, considering the expert system ISPBEX [Flam *et al.* 1991] which is designed to determine the proper treatment recommendations for the suppression of Southern Pine Beetle(SPB), *Dendroctonus Frontalis Zimmerman* infestations. Management decisions are made based on the knowledge acquired from experts specializing in forest management, wildlife management, and SPB biology and control. The purpose of ISPBEX is to help Forest Service personnel make decisions about the Southern Pine Beetle spots by providing treatment recommendations. The requirements for the treatment recommendation generated by the system includes

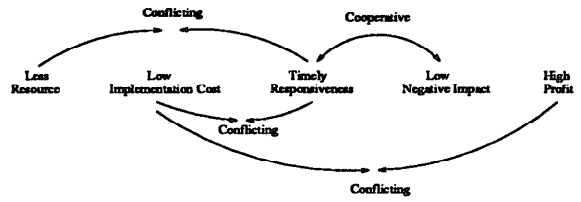

Figure 3: An Qualitative Specification of Relationships Between Requirements

- R_1 : The amount of resource required should be small.
- R_2 : The cost of implementation of the treatment should be low.
- R_3 : The time taken to begin the treatment should be short.
- R_4 : The impact of the treatment on endangered species (RCW) should be reduced to a low level.
- R_5 : The profit should be high.

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

- R_1 : Amount_of_Resource (Treatment Recommendation) should be SMALL.
- R_2 : Cost_of_implementation (Treatment Recommendation) should be LOW.
- R_3 : Responsive_Time (Treatment Recommendation) should be SHORT.
- R_4 : Negative_impact(endangered_species (Treatment Recommendation)) should be LOW.
- R_5 : Profit(Treatment Recommendation) should be HIGH.

SMALL, LOW, SHORT, and HIGH are fuzzy sets and serve as elastic constraints on a treatment recommendation.

The qualitative specification of relationships between these requirements is shown in Fig. 3.

The qualitative specification of the relationships between requirements can be then validated and revised by the approximate analysis of quantitative relationships between requirements based on an analytic tool called the Analytic Matrix (AM) and the theoretical results presented before. Let T denote the target

Case	Solution	Requirement				
		R_1	R_2	. . .	R_n	
EC_1	$S_{1,1}$	$d_{1,1,1}$	$d_{1,1,2}$	. . .	$d_{1,1,n}$	
	$S_{1,2}$	$d_{1,2,1}$	$d_{1,2,2}$	. . .	$d_{1,2,n}$	
	.	.	.		.	
	.	.	.		.	
	.	.	.		.	
	$S_{1,m}$	$d_{1,m,1}$	$d_{1,m,2}$	. . .	$d_{1,m,n}$	
.			.			
.			.			
.			.			
EC_p	$S_{p,1}$	$d_{p,1,1}$	$d_{p,1,2}$	. . .	$d_{p,1,n}$	
	$S_{p,2}$	$d_{p,2,1}$	$d_{p,2,2}$	. . .	$d_{p,2,n}$	
	.	.	.		.	
	.	.	.		.	
	.	.	.		.	
	$S_{p,m}$	$d_{p,m,1}$	$d_{p,m,2}$	. . .	$d_{p,m,n}$	

Figure 4: An Illustration of Analytic Matrix

Case	Solution	Requirement				
		R1	R2	R3	R4	R5
Case 1	C/L	0.4	0.4	0.6	0.7	0.2
	C/R	0.2	0.3	0.6	0.6	0.9
	P/B	0.1	0.1	0.8	0.1	0.1
	Mon	0.8	0.8	0.3	0.5	0.1
	C/HS	0.3	0.2	0.7	0.8	0.1
Case 2	C/L	0.1	0.3	0.5	0.6	0.1
	C/R	0.1	0.2	0.7	0.7	0.4
	P/B	0.1	0.1	0.9	0.9	0.1
	Mon	0.6	0.5	0.2	0.3	0.1
	C/HS	0.1	0.4	0.4	0.4	0.1
Case 3	C/L	0.6	0.7	0.6	0.6	0.3
	C/R	0.4	0.4	0.6	0.3	0.6
	P/B	0.2	0.1	0.7	0.1	0.1
	Mon	0.8	0.8	0.2	0.5	0.1
	C/HS	0.5	0.2	0.3	0.4	0.1
Case 4	C/L	0.8	0.4	0.7	0.7	0.5
	C/R	0.8	0.3	0.8	0.6	0.7
	P/B	0.9	0.1	0.9	0.1	0.1
	Mon	0.9	0.8	0.6	0.9	0.1
	C/HS	0.8	0.6	0.8	0.8	0.1

Figure 5: The Analytic Matrix for ISPBEX

knowledge-based system. It has n individual requirements, denoted as R_k, $1 \leq k \leq n$. A set of p typical cases ,denoted as EC_i, $1 \leq i \leq p$ are formulated by knowledge engineers and domain experts. For example, considering a medical diagnosis expert systems, a case may consists of a list of symptoms a patient may have. The domain experts are then asked to provide a set of m possible solutions for each case EC_i, denoted as $S_{i,j}$, $1 \leq j \leq m$. For each solution $S_{i,j}$, a degree $d_{i,j,k}$, to which the solution $S_{i,j}$ for the case EC_i satisfies the requirement R_k can be assigned according to Theorem 1. These degrees are organized into a matrix, called the Analytic Matrix (AM), $AM = (d_{i,j,k})$, where $1 \leq i \leq p, 1 \leq j \leq m$, and $1 \leq k \leq n$, as shown in Fig. 4.

To illustrate the concept of the analytic matrix, let us consider ISPBEX again. In ISPBEX, the input parameters are treatment month, type of forest (general forest, wilderness), colony-condition (active, not active), breeding season (spring, summer, fall, winter), priority(A,B,C,D where A has the highest priority and D the lowest), type of trees (loblobby or short leaf pine), characteristics of trees (age, size), growth distance (in feet) and direction of growth (in degrees). The output is a treatment recommendation and explanation why a particular recommendation is provided. Thus a case corresponds to a set of fixed input parameters. For example, the Case 1 in the analytic matrix, shown in Fig. 5, represents that the treatment season is march, the type of forest is wilderness, SPB spot is active, endangered species are breeding, priority is high, SPB is going to impact the endangered species colony within 60 days, and so on. The analytic matrix for ISPBEX includes four typical cases. For each case, five major recommendations are possible. They include Cut and Leave(C/L), Cut and Remove(C/R), Pile and Burn(P/B), Monitor(Mon) and Cut and Hand Spray(C/HS). For example, the recommendation P/B requires the largest amount of resource for Case 1 and satisfies the requirement R_1 at the lowest degree (0.1).

Now let us to explain how to use the analytic matrix for the quantitative analysis of the relationships between requirements. Actually, a before state corresponds to a legal input of the system and an after state corresponds to an output which is expected to be generated for a given input by the system in our paradigm. Thus, a before state corresponds to a case and an after state corresponds to a possible solution given for the case in the analytic matrix.

Considering two requirement R_{k_1} and R_{k_2}, $1 \leq k_1, k_2 \leq n$. Let

$$\mathcal{F}'(EC_i) = \{(S_{i,j_1}, S_{i,j_2}) | ((d_{i,j_1,k_1} - d_{i,j_2,k_1}) \times (d_{i,j_1,k_2} - d_{i,j_2,k_2})) < 0, 1 \leq j_1, j_2 \leq m, j_1 \neq j_2\}.$$

Then the estimation $conf'(EC_i)$ of the conflicting degree $conf(EC_i)$ between R_{k_1} and R_{k_2} for the case EC_i is $\frac{\|\mathcal{F}'(EC_i)\|}{m(m-1)}$. For example, the estimation of the conflicting degree between requirement R_1 and R_3 for the Case 1 is $conf'(EC_1) = \frac{\|\mathcal{F}'(EC_1)\|}{m(m-1)} = 0.8$

Similarly the cooperating degree of two requirements for a given case can also be estimated based on the analytic matrix. From the estimated conflicting degree and cooperating degree of two requirements for each case, the optimistic, pessimistic, and average conflicting degree, and the optimistic, pessimistic, and average

cooperating degree for all cases given in the analytic matrix can be estimated based on their definitions. The qualitative specification of the relationships between requirements can then be validated and revised by the quantitative analysis. For example, it has been found that R_1 and R_2 are cooperative in ISPBEX by the quantitative analysis, which has missed in the qualitative specification given by domain experts.

Combining Conflicting Requirements

Compromise Operators

Averaging [Dubois & Prade 1984] and compensatory [Zimmermann 1991] 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 6 *(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. Thus the "min", a t-norm operator, and the "max", a t-conorm operator, are not compromise operator since they are not compensatory. For a compromise operator, a decrease in one operand can be compensated by an increase in another operand. The arithmetic mean is an example of the C operator.

Combine Conflicting 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 $\{p_1 \ldots p_k\}$.

Since the cooperating requirements can be satisfied at the same time, it is appropriate that they are combined with t-norms operators [Zimmermann 1991, Yen & Lee 1993b]. To use compromise operators to aggregate the conflicting requirements, we define the following rule to compute the function *fhold* for aggregation.

Definition 7 *(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 $p_i, 1 \leq i \leq n$, are propositions, s is a state, and C is a compromise operator.

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 7 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 $u_i \in U_i$.

The variety of compromise operators might make it hard to decide which one to use in a specific application. Several criteria have been summarized by Zimmermann [Zimmermann 1991] for selecting the appropriate general aggregation operator. 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; 2. Feasibility: The operator should make the combined requirement feasible and increase its feasibility. 3. Conflicting type and conflicting degree: A different trade-off strategy may be used for combining requirements with different conflicting type and degree. 4. Criticality: The operator should be able to handle criticality in the requirement specification.

Requirement Engineering Process

The requirements need to be revised many times during the development of knowledge-based systems. Actually the refinement of requirements with complex relationships between them is an iterative process, as shown in Fig. 6.

Knowledge engineers acquire individual requirements by a number of interaction with domain experts. The domain experts are then asked to provide the qualitative descriptions of the relationships between requirements. These qualitative relationships are not only the basis of but also validated and revised by the further quantitative analysis of relationships between requirements. The analytic matrix is an effective tool of the quantitative analysis.

Based on the analytic matrix, the quantitative analysis can be conducted by the computation of the conflicting degree and cooperating degree, and the classification of the relationships based on these degrees. The results of quantitative analysis can be used to decide the trade-off strategies and select appropriate aggregation operators to form a feasible overall system requirement. These trade-off strategies can be evaluated as follows:

1. The selected aggregation operators are used to combine the individual requirements and thus the degree, to which the overall requirement is satisfied by a possible solution given for a case in the analytic matrix, can be calculated.
2. The solutions for each case are ordered in the descending of these degrees. If these orders are quite different from the orders given by the domain experts

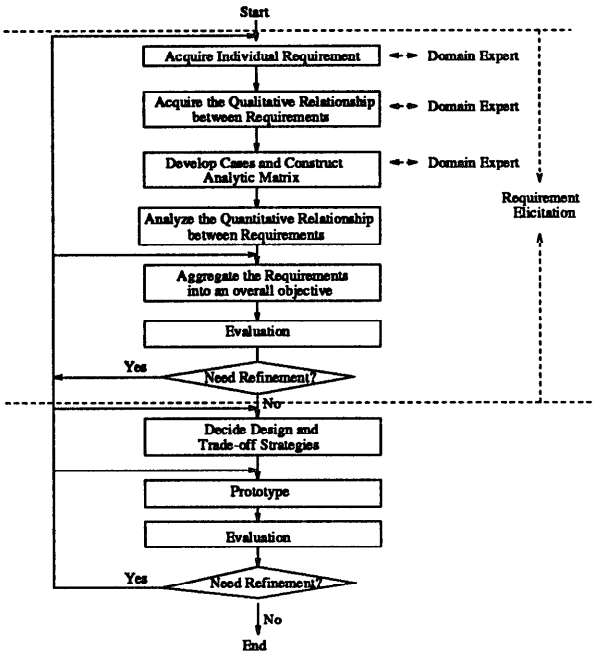

Figure 6: A Requirement Engineering Process

when the analytic matrix is obtained, the trade-off strategies needs to be revised.

If the evaluation result is positive, the design strategies need to be decided to fulfill the requirements effectively. A prototype can be then built to verify and validate the requirements. The prototype is tested by running a number of tests and their results are compared with the requirement specification and the opinions of domain experts. If the result of the evaluation is negative, the individual requirement may need to be altered since it may be too strong, trade-off strategies may need to be changed, or new design strategies and implementation techniques may need to be used.

Conclusion

In this paper, a systematic approach for specifying, analyzing and evaluating imprecise requirements using fuzzy logic has been developed. It provides both a sound theoretical foundation and a realistic method for the analysis of the complex relationships between requirements. A feasible overall requirement can be thus formulated from conflicting requirements by using fuzzy multi-criteria optimization technique. Moreover it can help to achieve the optimal system objective by a trade-off analysis of conflicting requirements.

Acknowledgements

We wish to thank Professor Lotfi A. Zadeh for suggesting that we use fuzzy multi-criteria decision techniques to aggregate soft requirements. We also would thank Dr. Jonathan Lee for his early work in this research. This research is supported by NSF Young Investigator Award IRI-9257293.

References

Gruia-Catalin Roman. 1985. A Taxonomy of current issues in requirements engineering. *IEEE Computer* 4:14-21.

R. Balzer; N. Goldman and D. Wile. 1978. Informality in program specifications. *IEEE Transaction on Software Enginering* 4(2):94-103.

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

R. O. Flamm, *et. al.* 1991. The Integrated Southern Pine Beetle Expert System: ISPBEX, *Journal of Expert Systems with Applications*, 2:97-105.

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

A. Batarekh; A.D. Preece; A. Bennett and P. Grogono. 1991. Specifying An Expert System. *Journal of Expert Systems with Applications* 2(3):285-303.

R. Plant. 1988. Specifying Expert System in Z. *PhD thesis*, University of Liverpool.

H.B. Reubenstein and R.C. Waters. 1991. The requirements apprentice: automated assistance for requirements acquisition. *IEEE Transaction on Software Engineering* 17(3):226-240.

W. N. Robinson. 1990. Negotiation behavior during requirement specification. In *Proc. of the 12th International Conf. on Software Engineering.* 268-276.

W.T. Tsai; P.E. Johnson; J.R. Slagle; I.A. Zualkernan, *et. al.* 1988. Requirements Specification for Expert Systems? A Case Study, *Technical Report TR 88-44*, University of Minnesota, Minneapolis, MN.

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

R. Yager. 1991. Connectives and Quantifiers in Fuzzy Sets. *Fuzzy Sets and Systems* 40:39-75.

J. Yen and J. Lee. 1993a. A task-based methodology for specifying expert systems, *IEEE Expert* 8(1):8-15.

J. Yen and J. Lee. 1993b. Fuzzy Logic as a Basis for Specifying Imprecise Requirements. In *Proc. of the IEEE International Conf. on Fuzzy Systems*, 745-749.

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

H.-J. Zimmermann. 1991. *Fuzzy Set Theory and Its Applications.* Mass.: Kluwer Academic.