



Missouri University of Science and Technology
Scholars' Mine

Computer Science Faculty Research & Creative Works

Computer Science

01 Jan 1996

A Systematic Tradeoff Methodology for Acquiring and Validating Imprecise Requirements

J. Yen

Xiaoqing Frank Liu

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

W. A. Tiao

Follow this and additional works at: https://scholarsmine.mst.edu/comsci_facwork

 Part of the [Computer Sciences Commons](#)

Recommended Citation

J. Yen et al., "A Systematic Tradeoff Methodology for Acquiring and Validating Imprecise Requirements," *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, 1996*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1996.

The definitive version is available at <https://doi.org/10.1109/FUZZY.1996.551766>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Computer Science Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

A Systematic Tradeoff Methodology for Acquiring and Validating Imprecise Requirements

John Yen W. Amos Tiao

Center for Fuzzy Logic, Robotics, and Intelligent Systems
Department of Computer Science
Texas A&M University
College Station, Texas 77843-3112

Xiaoqing Frank Liu

Department of Computer Science
University of Missouri
Rolla, Missouri 65401

Abstract

Requirement analysis is one of the most important phases in a software development process. Existing requirement methodologies are limited in specifying requirements that are usually vague and imprecise, and in supporting tradeoff analysis between the conflicting requirements. In this paper, the elasticity of imprecise requirements is captured using fuzzy logic to facilitate tradeoffs between conflicting requirements. Based on the marginal rate of substitution in decision science, we have developed a systematic approach to elicit the structures and the parameters of imprecise requirements, to validate the scheme for aggregating requirements, and to assess relative priorities of conflicting requirements.

1 Introduction

Requirement analysis begins when there is recognition that a problem exists and requires a software solution, and ends when we have a specification of the software to be built. The specification contains functional requirements that describe the external behavior of the software in terms of inputs, outputs, and their relationships, and nonfunctional requirements that impose constraints on the software, such as resource, cost, and politics. Brooks pointed out that no other parts of the work so cripple the resulting system if done wrong and no other parts are more difficult to rectify later than requirement analysis [3].

There are at least two challenges with requirement engineering [9, 6]. First, requirements are usually *vague* and *imprecise* in nature. Therefore, there is a need to bridge the gap between imprecise requirements and formal specification methods. Actually, as Balzer *et. al.* pointed out, informality is an inevitable and ultimately desirable feature of the specification

process [1]. Second, requirements often conflict with each other, which many conflicts are implicit and difficult to identify [2, 7]. Assessing the tradeoffs among conflicting requirements is a very challenging issue.

The imprecise nature of requirements leads to a mismatch to the existing formal specification methods. Most existing formal specification methodologies require the requirements to be stated precisely [7] or convert imprecise requirements into precise ones [1, 4]. If requirements are specified to be crisp, that is, requirements are either satisfied or not satisfied at all, their capabilities of capturing the semantics of imprecise requirements are limited.

Most early software specification methods consider a requirement specification containing conflicting requirements to be inconsistent. In the software development process, conflicts are inevitable and can be beneficial if they can be managed well. Several works have focused on conflict detection and resolution in requirement engineering [2, 8]. They are, however, limited in detecting implicit conflicts and do not address the issues of tradeoff analysis. If requirements are crisp, one of the conflicting requirements has to be dropped or modified to resolve the conflict. However, each conflicting requirement can be satisfied to some degree if the elasticity of a requirement is captured. Hence, it is possible to explore an effective tradeoff among conflicting requirements.

In this paper, imprecise requirements are represented by fuzzy logic so that the elasticity of constraints imposed by imprecise requirements can be captured. Once the conflicts between requirements are identified, they need to be resolved effectively. Tradeoffs should be made between conflicting requirements. Based on the marginal rate of substitution developed in decision science, we have developed a systematic approach to elicit the structures and the parameters of imprecise requirements, to validate the scheme for

aggregating requirements, and to assess relative priorities of conflicting requirements.

2 Representation of Imprecise Requirements Using Fuzzy Logic

Requirements represent the elastic criteria against which the acceptability of a realization of a target system is judged. The foremost effort in requirement analysis is to represent the system requirements. The universe being constrained by a requirement is called its domain. Typical domains for imprecise requirements include (1) the domain containing all possible system development processes under consideration, (2) the domain containing all possible system realizations under consideration, and (3) the domain containing all possible input-output state transitions.

The constraint imposed by an imprecise requirement R is represented as a satisfaction function, denoted as Sat_R , that maps an element of R 's domain D to a number in $[0, 1]$ that represents the degree to which the requirement is satisfied:

$$Sat_R : D \rightarrow [0, 1]. \quad (1)$$

In essence, the satisfaction function characterizes a fuzzy subset of D that satisfies the imprecise requirement.

The *canonical form* in Zedah's test score semantics is used as a basis for expressing imprecise requirements [11]. The representation of imprecise requirements on a system development process in canonical form is established by the following definition [6, 10].

Definition 1 Let R be an imprecise requirement on system development process in canonical form $R : A_i(p)$ is B , where p is a system development process, A_i is a property of the process, such as cost, B is a fuzzy set. Then

$$Sat_R(p) = \mu_B(A_i(p)).$$

An imprecise requirement about overall system behavior can be expressed using a summarization operation, such as AVERAGE, MIN, and MAX as follows.

Definition 2 Assume that R is an imprecise requirement about overall system functional behavior and is expressed in canonical form $R : \Psi_{R_i}(r)$ is B , where Ψ is a summarization operator, such as AVERAGE, R_i is a specific functional requirement, and r is a realization. The satisfaction degree of R is defined as

$$Sat_R(r) = \Psi_{\langle s_1, s_2 \rangle \in \Gamma_2(r)}(Sat_{R_i}(\langle s_1, s_2 \rangle)),$$

where $\langle s_1, s_2 \rangle$ is a state transition and $\Gamma_2(r)$ is the set of state transitions performed by a realization r .

3 Tradeoff Analysis between Requirements

Once we have represented the imprecise requirements, we would like to know the relationship between requirements. We have identified four types of significant relationships between requirements: conflicting, cooperative, mutually exclusive, and irrelevant [6]. Among these four types of relationships, we are particularly interested in the conflicting relationship. Two imprecise requirements, R_1 and R_2 , are said to be *conflicting* with each other if an increase in the degree of satisfaction of R_1 (R_2) often causes a decrease in the degree of satisfaction of R_2 (R_1). If an increase in the satisfaction degree of one requirement always decreases the satisfaction degree of the other, they are said to be *completely conflicting*.

Once conflicting relationship between imprecise requirements has identified, an effective tradeoff between them should be made. That is, we should explore a plausible approach that increases the degree of satisfaction of one requirement by sacrificing the degree of satisfaction of another requirement, while the overall degree of satisfaction should be maximized.

Keeney and Raiffa have developed the concepts of *value function*, *indifference curve* and *Marginal Rate of Substitution* (MRS) for the tradeoff analysis between multiple criteria in decision science [5]. A value function refers to the overall satisfaction. An indifference curve represents all the alternative combinations of X and Y for which a customer is equally well off. That is, the alternatives on an indifference curve all provide the same level of satisfaction. The MRS indicates the maximal amount of a decision attribute that a customer is willing to sacrifice for a unit increase in another decision attribute. For example, if, at the point (x, y) , the customer is willing to give up $\lambda\Delta$ units of X for Δ units of Y , then the MRS of X for Y at point (x, y) is λ . This is formally established by the following definition.

Definition 3 Suppose X and Y are two decision attributes, x_1 and y_1 are values for X and Y , respectively. If the indifference curve through (x_1, y_1) is given by $v(x, y) = c$, then the marginal rate of substitution (MRS) λ at (x_1, y_1) is

$$\lambda = -\frac{dx}{dy} \Big|_{x_1, y_1} = \frac{v'_y(x_1, y_1)}{v'_x(x_1, y_1)},$$

where v'_x and v'_y are the partial derivatives of v with respect to the first and second arguments, respectively.

Generally speaking, MRSs of two decision attributes at two different decision points (e.g., λ at

(x_1, y_1) and λ at (x_2, y_2)) can be different. If MRSs of two decision attributes are the same for all decision points, we call it a constant MRS.

3.1 Priority Assessment Based on Constant MRS

A set of requirements often need to be ordered based on their degree of importance to resolve the potential conflicts among them. The purpose of requirement priority analysis is to establish an ordering of requirements based on their importance and discover how much a requirement is more important than another requirement. It is, however, often difficult for customers to directly provide priorities for all requirements due to complex relationships between them. Thus it is desirable to develop a technique to assist them in identifying the priority of each requirement.

Compromise operators are often used to achieve a tradeoff among conflicting requirements. Since the weighted arithmetic average operator is one of the most widely used compromise operators to aggregate requirements with complex tradeoff relationships, we focus our discussion on assessing the priority of requirements combined using this operator. The overall satisfaction degree of the combined requirements can be computed as follows:

$$\sum_{i=1}^n w_i \times Sat_{R_i}(A_i(p)),$$

where w_i is a normalized weight of requirement R_i , p is a process, A_i is an attribute of p on which requirement R_i imposes a constraint, and $Sat_{R_i}(A_i(p))$ is the satisfaction degree of the case on the requirement. For convenience, $A_i(p)$ is abbreviated as x_i in the following discussion. The following theorem show how to compute relative priority based on a constant MRS.

Theorem 1 *We assume that an imprecise requirement R_k ($k = i, j$) has a linear satisfaction function which maps an attribute value to a satisfaction degree: $Sat_{R_k}(x_k) = a_k \times x_k + b_k$, $k = i, j$. In addition, we assume that the marginal rate of substitution of attribute x_j for x_i is a constant $\lambda_{i,j}$. Then,*

$$\frac{w_i}{w_j} = \frac{a_j}{a_i} \lambda_{i,j}.$$

Proof: Since $\lambda_{i,j}$ is the MRS of attribute x_j for x_i , the level of overall satisfaction degree should be the same before and after the substitution. Hence, we have

$$\sum_{k=1}^n (w_k \times Sat_{R_k}(x_k)) =$$

$$\sum_{k=1, k \neq i, j}^n (w_k \times Sat_{R_k}(x_k)) + w_i \times Sat_{R_i}(x_i + 1) + w_j \times Sat_{R_j}(x_j - \lambda_{i,j}).$$

That is,

$$w_i \times Sat_{R_i}(x_i) + w_j \times Sat_{R_j}(x_j) = w_i \times Sat_{R_i}(x_i + 1) + w_j \times Sat_{R_j}(x_j - \lambda_{i,j}).$$

Because of the linear membership function assumption, we have

$$w_i \times (a_i \times x_i + b_i) + w_j \times (a_j \times x_j + b_j) = w_i \times (a_i \times (x_i + 1) + b_i) + w_j \times (a_j \times (x_j - \lambda_{i,j}) + b_j).$$

After simplification, we obtain $w_i \times a_i = w_j \times a_j \times \lambda_{i,j}$.

That is,

$$\frac{w_i}{w_j} = \frac{a_j}{a_i} \lambda_{i,j}.$$

Therefore, we have proven the theorem.

3.2 Assessing Satisfaction Function Using MRS

One of the most critical tasks in a formal approach to specify imprecise requirements is to determine the structures and the parameters of satisfaction functions that reflect the customer's intention and preferences. Hence it is desirable to develop a systematic approach to assist customers and requirement engineers to construct the satisfaction functions.

Suppose that customers think that the substitution rate between two requirements is constant. From the definition of marginal rate of substitution, it is easy to derive that the value function is in the following form:

$$v(x, y) = x + \lambda y + c,$$

where λ is the constant MRS and c is a constant. Assuming that customers also feel that it is appropriate to use a linear satisfaction function to specify each imprecise requirement, we can derive a systematic way to assign the parameters of the linear function using the following theorem.

Theorem 2 *We assume overall satisfaction function is aggregated using the weighted summation, $\lambda_{i,j}$ is a constant MRS for requirements R_i and R_j , and requirement R_i and R_j have a linear satisfaction function: $Sat_{R_k}(x_k) = a_k \times x_k + b_k$, $k = i, j$. In addition,*

we assume that the relative priority between the two requirements is fixed, i.e. $\frac{w_i}{w_j} = \alpha$, where α is a constant, and w_i and w_j are normalized weights of requirements R_i and R_j , respectively. Then, we have

$$\begin{aligned} a_i &= \frac{\alpha + 1}{\alpha}, \\ a_j &= (\alpha + 1)\lambda_{i,j}, \text{ and} \\ \alpha b_i + b_j &= c(\alpha + 1). \end{aligned}$$

Proof: Since $\lambda_{i,j}$ is a constant, we get

$$v(x_i, x_j) = x_i + \lambda_{i,j}x_j + c. \quad (2)$$

Using weighted summation, we get

$$v(x_i, x_j) = w_i \times \text{Sat}_{R_i}(x_i) + w_j \times \text{Sat}_{R_j}(x_j).$$

Because of fixed relative priority and the linear satisfaction function assumption, we have

$$v(x_i, x_j) = \frac{\alpha}{\alpha + 1}(a_i x_i + b_i) + \frac{1}{\alpha + 1}(a_j x_j + b_j).$$

After simplification, we obtain

$$v(x_i, x_j) = \frac{\alpha a_i}{\alpha + 1}x_i + \frac{a_j}{\alpha + 1}x_j + \frac{\alpha b_i + b_j}{\alpha + 1}. \quad (3)$$

The right hand sides of Eq. (2) and Eq. (3) should be equal for all x_i and x_j , thus we have

$$\begin{aligned} 1 &= \frac{\alpha}{\alpha + 1}a_i, \\ \lambda_{i,j} &= \frac{1}{\alpha + 1}a_j, \text{ and} \\ c &= \frac{\alpha b_i + b_j}{\alpha + 1}. \end{aligned}$$

After simplification, we obtain

$$\begin{aligned} a_i &= \frac{\alpha + 1}{\alpha}, \\ a_j &= (\alpha + 1)\lambda_{i,j}, \text{ and} \\ \alpha b_i + b_j &= c(\alpha + 1). \end{aligned}$$

Thus, we have proven the theorem.

There are two possible uses of the above theorem. First, requirement engineers may use it to construct satisfaction functions. Second, it can be used to validate satisfaction functions previously formulated.

3.3 Assessing the Structures of Satisfaction Functions Based on Dynamic MRS

In section 3.2, we have shown that parameters of satisfaction function can be constructed in a systematic way if the satisfaction function is linear. In general, we may need to determine the structure of the

satisfaction function before we can actually identify its parameters. Hence, it is very desirable that we can develop a technique for determining the possible structures of the satisfaction function based on a systematic assessment.

Suppose that customers feel that the substitution rates depend on one requirement but not on the other, how can this qualitative statement help in the assessment of the possible structures of the satisfaction function? To answer this question, we first determine the form of a value function. If the MRS depends on y but not on x , it is easy to derive the following form for the value function

$$v(x, y) = x + v_Y(y) + c,$$

where v_Y is a value function over requirement Y and c is a constant. That is, the amount a customer is willing to pay in x units for additional y units depends on the level of y but not on the level of x . Under such a circumstance, we can determine the possible structures of the satisfaction function of attribute X using the following theorem. For notational convenience, we use x and y to represent x_i and x_j , respectively.

Theorem 3 We assume the overall satisfaction function is aggregated using the weighted summation, and relative priority between requirements R_i and R_j is fixed, i.e. $\frac{w_i}{w_j}$ is a constant, where w_i and w_j are normalized weight of imprecise requirements R_i and R_j , respectively. Let attributes constrained by R_i and R_j be denoted by x and y , respectively. In addition, we assume MRS satisfies

$$\lambda(x_1, y_1) = \lambda(x_2, y_1), \forall x_1, x_2, y_1.$$

That is, λ depends on the level of y but not on the level of x . Then we have

$$\text{Sat}_{R_i}(x) = a_i x + b_i.$$

where a_i and b_i are constants.

Proof: Since the MRS depends on the level of y but not on the level of x , we have

$$v(x, y) = x + v_Y(y) + c, \quad (4)$$

where $v_Y(y)$ is a value function over attribute Y and c is a constant. Using the weighted summation to aggregate requirements, we get

$$v(x, y) = w_i \times \text{Sat}_{R_i}(x) + w_j \times \text{Sat}_{R_j}(y). \quad (5)$$

Because of the fixed relative priority, it implies w_i and w_j are constants. Since the right hand sides of Eq. (4)

and Eq. (5) should be equal for all x and y , $Sat_{R_i}(x)$ should be a linear structure. That is,

$$Sat_{R_i}(x_i) = a_i x_i + b_i.$$

where a_i and b_i are constants. Therefore, we have proven the theorem.

Furthermore, a customer sometimes may be willing to pay less and less for an additional unit in Y as the value of y increases. In other words, he might feel that

$$v_Y(y+1) - v_Y(y) < v_Y(y) - v_Y(y-1), \forall y; \quad (6)$$

it worths less to go from y to $y+1$ than from $y-1$ to y , regardless of the value of y . It can be shown that v_Y is strictly concave if λ does not depend on x and it decreases as y increases. One example of concave functions is \sqrt{y} . This is formally stated in the theorem below.

Theorem 4 We assume the overall satisfaction function is aggregated using the weighted summation, and relative priority between requirements R_i and R_j is fixed, i.e. $\frac{w_i}{w_j}$ is a constant, where w_i and w_j are normalized weights of requirements R_i and R_j , respectively. In addition, we assume marginal rate of substitution satisfies the following conditions;

$$(1) \lambda(x_1, y_1) = \lambda(x_2, y_1), \forall x_1, x_2, y_1;$$

$$(2) \lambda(x_1, y_1) < \lambda(x_1, y_2), \forall y_1 > y_2.$$

Then we have

$$Sat_{R_i}(x) = a_i x + b_i,$$

$$Sat_{R_j}(y) \text{ is a concave function,}$$

where a_i and b_i are constants.

Proof: The proof is similar to that of Theorem 3.

3.4 Assessing the Validity of Operator

In the previous sections, we have assumed that the overall satisfaction function is aggregated using weighted summation. There are, however, many operators that can be used to aggregate a set of imprecise conflicting requirements. Thus, it is desirable to develop techniques for validating the correctness of a chosen aggregation operator.

In general, the marginal rate of substitution depends on the level of x and on the level of y . A value function is *additive* if it has the form $v(x, y) = v_X(x) + v_Y(y)$, where $v_X(x)$ and $v_Y(y)$ are single variable value functions depending only on attribute X and Y , respectively. For an overall satisfaction function combined by the weighted summation, we have

the structure $\sum_{i=1}^n w_i \times Sat_{R_i}(x_i)$. Assuming that each weight w_i is fixed, it is easy to show that this function is *additive*. Then we can formally describe the *corresponding tradeoff condition* in decision science as follows.

Theorem 5 Consider four points $A : (x_1, y_1)$, $B : (x_1, y_2)$, $C : (x_2, y_1)$, and $D : (x_2, y_2)$ in an evaluation space. The corresponding tradeoff condition is said to be satisfied if given

$$\begin{aligned} \lambda(x_1, y_1) &= \lambda_A = \frac{b}{a}, \\ \lambda(x_1, y_2) &= \lambda_B = \frac{c}{a}, \text{ and} \\ \lambda(x_2, y_1) &= \lambda_C = \frac{b}{d}, \end{aligned}$$

we have

$$\lambda(x_2, y_2) = \lambda_D = \frac{\lambda_B \cdot \lambda_C}{\lambda_A} = \frac{c}{d}. \quad (7)$$

Proof: For simplicity, we assume the value function is additive. Thus, we can represent the value function as $v(x, y) = v_X(x) + v_Y(y)$. From the definition of MRS, we obtain

$$\lambda_A = \lambda(x_1, y_1) = \frac{v'_y(x_1, y_1)}{v'_x(x_1, y_1)} = \frac{v'_Y(y_1)}{v'_X(x_1)} = \frac{b}{a};$$

$$\lambda_B = \lambda(x_1, y_2) = \frac{v'_y(x_1, y_2)}{v'_x(x_1, y_2)} = \frac{v'_Y(y_2)}{v'_X(x_1)} = \frac{c}{a};$$

$$\lambda_C = \lambda(x_2, y_1) = \frac{v'_y(x_2, y_1)}{v'_x(x_2, y_1)} = \frac{v'_Y(y_1)}{v'_X(x_2)} = \frac{b}{d}.$$

Therefore,

$$\begin{aligned} \lambda_D &= \lambda(x_2, y_2) = \frac{v'_y(x_2, y_2)}{v'_x(x_2, y_2)} \\ &= \frac{v'_Y(y_2)}{v'_X(x_2)} = \frac{v'_Y(y_2)}{v'_X(x_1)} \frac{v'_X(x_1)}{v'_Y(y_1)} \frac{v'_Y(y_1)}{v'_X(x_2)} \\ &= \lambda_B \cdot \frac{1}{\lambda_A} \cdot \lambda_C \\ &= \frac{\lambda_B \cdot \lambda_C}{\lambda_A} = \frac{c}{d}. \end{aligned} \quad (8)$$

Thus we have proven the theorem.

Based on the works in decision science, the corresponding tradeoff condition provides us with necessary and sufficient conditions for an important result, which is described in the following theorem [5].

Theorem 6 A value function is additive if and only if the corresponding tradeoff condition is satisfied.

From Theorem 5 and Theorem 6, we can develop a procedure to assess the validity of the weighted summation.

4 Conclusion

In this paper, a systematic approach has developed for specifying imprecise requirements and resolving conflicts between them. The elasticity of imprecise requirements is captured based on the fuzzy logic. Based on the marginal rate of substitution in decision science, we have developed a systematic approach for assessing the relative priority of imprecise requirements. We have also developed techniques to determine the structures and the parameters of membership functions that characterizes the elasticity of requirements. Finally, we have derived a procedure to validate whether requirements should be aggregated through an additive form. These techniques not only can facilitate the acquisition and the validation of requirements in software engineering, they can also be applied to the acquisition and the validation of decision criteria in multi-criteria fuzzy decision making.

Acknowledgements

This research is supported by National Science Foundation Young Investigator Award IRI-9257293 and by Texas Higher Education Coordination Board through Advanced Research Program Award 999903-253.

References

- [1] R. Balzer, N. Goldman, and D. Wile. Informality in program specifications. *IEEE Transactions on Software Engineering*, 4(2):94–103, 1978.
- [2] A. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, and B. Nuseibeh. Inconsistency handling in multiperspective specifications. *IEEE Transactions on Software Engineering*, 20(8):569–578, Aug. 1994.
- [3] J. Frederick P. Brooks. No silver bullet: essence and accidents of software engineering. *Computer*, 15(1):10–18, April 1987.
- [4] W. L. Johnson, M. S. Feather, and D. R. Harris. The kbsa requirements/specification facet: Aries. In *Proc. of the 6th Annual Knowledge-Based Software Engineering Conference*, pages 156–162, 1991.
- [5] R. L. Keeney and H. Raiffa. *Decisions with multiple objectives: preferences and value tradeoffs*. Wiley, New York, 1976.
- [6] X. F. Liu and J. Yen. An analytic framework for specifying and analyzing imprecise requirements. In *Proc. of the 18th International Conference on Software Engineering*, March 1996.
- [7] J. Mylopoulos, L. Chung, and B. Nixon. Representing and using nonfunctional requirements: a process-oriented approach. *IEEE Transactions on Software Engineering*, 18(6):483–497, June 1992.
- [8] W. N. Robinson. Negotiation behavior during requirement specification. In *Proc. of the 12th International Conference on Software Engineering*, pages 268–276, 1990.
- [9] J. Yen and J. Lee. Fuzzy logic as a basis for specifying imprecise requirements. In *Proc. of the IEEE International Conference on Fuzzy Systems*, pages 745–749, 1993.
- [10] J. Yen, X. Liu, and S. H. Teh. A fuzzy logic-based methodology for the acquisition and analysis of imprecise requirements. *Concurrent Engineering: Research and Applications*, (2):265–277, 1994.
- [11] L. Zadeh. Test-score semantics as a basis for a computational approach to the representation of meaning. *Literacy Linguistic Computing*, 5(1):24–35, 1986.